The Application of Classical Information Retrieval Techniques to Spoken Documents

David Anthony James

Downing College
Cambridge
United Kingdom

February 1995

This thesis is submitted for consideration
for the Degree of Doctor of Philosophy
at the University of Cambridge

Summary

The research presented in this thesis addresses the topic of *ad hoc* retrieval of information from collections of spoken items such as radio news bulletins.

Modern digital computers are becoming increasingly adept at processing non–textual data, such as speech. Consequently, new methods are required to allow users to pin–point specific items of interest in large data collections. Such a method might exploit the Hidden Markov Model (HMM), which has proved successful as the basis for many experimental speech recognition systems, and the well–understood techniques of document retrieval that have arisen from many years' research into textual information retrieval (IR).

However, so far there has been little exploration of the potential combination of these methods in order to index "spoken word" data. In the IR community, several papers have put forward an approach to the problem but this approach has not been properly tested. Work done in the speech recognition area has tended to concentrate on developing systems for *topic classification*. These systems are extensively pre–trained for the task of partitioning a set of spoken messages into a set of disjoint and exhaustive classes, each one representing some topic. Their utility is, in practice, limited by the fixed class set and slow operation, and they do not represent an approach to the problem of retrieving items that correspond to *arbitrary* topics.

This thesis describes experiments combining the techniques of classical information retrieval with HMM–based speech recognition methods in order to retrieve items from a collection of spoken messages corresponding to items of radio news. In a baseline system, a new technique for wordspotting allows items matching an arbitrary expression of the information requirement to be retrieved quickly and reasonably accurately. The system is subsequently improved through the addition of appropriate language models and the use of state–of–the–art acoustic modelling. Finally, performance is compared with that obtained by two alternative approaches, including one recently proposed in the IR literature, and found to be considerably superior.

Key Words: speech recognition, information retrieval, topic classification, keyword spotting, wordspotting.

Declaration

This thesis describes research carried out in the Speech, Vision and Robotics Group of the University of Cambridge Engineering Department between October 1991 and February 1995. It is the result of my own work and includes nothing which is the outcome of work done in collaboration. The length of this thesis, excluding references and figure captions, is forty–four thousand words.

Acknowledgements

Firstly, I wish to express my gratitude to Prof. Steve Young, my supervisor over the last three (and a bit) years, for his support and guidance, and his provision of the excellent HTK software. I am also grateful to Dr Andy Hopper of Olivetti Research Ltd., for his generosity in providing a CASE award, and vision in suggesting spoken message retrieval before anyone else seems to have thought of it. In addition, I wish to thank Dr Karen Sparck Jones for her endless stream of ideas, and the readiness to share her experience in the field of IR.

I would like to thank my colleagues, many in number, in the Speech Vision and Robotics Group for their friendship and technical assistance. Special mention must go to Drs Gareth Jones, Jonathan Foote and Kate Knill for many technical discussions and their helpful comments on the myriad draft copies of this thesis.

I am also grateful to the Engineering and Physical Sciences Research Council for providing my main income during my research. I would also like to thank Peter Schäuble for his interest in my work and hospitality on several occasions.

Finally, I dedicate this thesis to my parents, with thanks for their unfailing encouragement, and to Fiona, who has had to put up with a lot.

Contents

1 Introduction 6
1.1 Speech Recognition . . . 7
1.2 Information Retrieval . . . 8
1.3 Structure of this Thesis . . . 8
1.4 Original Contribution . . . 9

2 Hidden Markov Modelling 10
2.1 Introduction . . . 11
2.2 HMM Anatomy . . . 11
2.3 Output Probability Distributions . . . 13
2.4 Model Training . . . 14
2.4.1 Initial Model Estimation . . . 14
2.4.2 Model Parameter Re–estimation . . . 16
2.4.3 The Forward–Backward Algorithm . . . 16
2.4.4 Transition Probabilities . . . 18
2.4.5 Extension to Gaussian mixture training . . . 18
2.4.6 Extension to embedded re–estimation . . . 19
2.5 Speech Recognition using HMMs . . . 20
2.6 Limitations of Hidden Markov Modelling . . . 23
2.7 Conclusions . . . 24

3 Information Retrieval 25
3.1 An introduction to IR . . . 26
3.2 Indexing . . . 27
3.2.1 Manual Indexing . . . 27
3.2.2 Automatic Indexing . . . 28
3.3 Queries and Relevance Assessments . . . 30
3.4 Query–Document Matching . . . 32

3.4.1 Inverse Document Frequency Weighting 33
3.4.2 Other Term Weighting Strategies 34
3.4.3 Relevance Feedback 35
3.5 Probabilistic Retrieval 36
3.6 System Evaluation 38
3.6.1 Average Precision 40
3.7 Conclusion 41

4 Previous Work in Spoken Message Retrieval 43
4.1 Wordspotting 45
4.2 Topic Classification Systems 51
4.3 Spoken Message Retrieval Systems 57
4.3.1 Sub–word methods 57
4.3.2 Whole–Word Methods 59
4.3.3 Spoken Indexing into Textual Document Collections 61
4.4 Conclusion 63

5 Baseline Retrieval System 64
5.1 Data Collection and Parametrisation 68
5.2 Test Data Collection 72
5.3 Lattice Generation 73
5.4 Queries and Relevance Assessments 76
5.5 Initial Retrieval Experiments 78
5.5.1 Textual Reference Run 79
5.5.2 Viterbi wordspotting 81
5.5.3 Lattice wordspotting 85
5.6 Analysis of the results 87
5.6.1 The effect of thresholding and term weighting 87
5.6.2 Varying the wordspotter threshold 89
5.7 Implementational considerations in lattice wordspotting 94
5.8 Conclusions 95

6 Improvements to Language and Acoustic Modelling 97
6.1 Stemming 98
6.2 No–Grammar Language Modelling 100
6.2.1 Fixed–Vocabulary Retrieval 100
6.2.2 Retrieval on Full Queries 106

6.2.3 Improvements to lattice term detection 108
6.3 Bigram Language Modelling 112
6.4 Improved Acoustic Modelling 116
6.4.1 Decision Tree–based Model Clustering 118
6.4.2 Training of new Acoustic Models 119
6.4.3 Retrieval experiments 121
6.5 Conclusion 123

7 Experiments with the Retrieval Model 125
7.1 The Application of Relevance Feedback 125
7.1.1 Feedback experiments on textual message representations . . . 126
7.1.2 Feedback experiments on spoken messages 129
7.1.3 Summary of Results on Relevance Feedback 131
7.2 Implementation of VCV–Feature Retrieval 131
7.2.1 Feature Selection and Acoustic Modelling 132
7.2.2 Improvements to Feature Modelling 137
7.2.3 Conclusions 138
7.3 Data–driven indexing 139
7.3.1 Data–driven Indexing Features 140
7.3.2 Query–Message Matching and Retrieval Results 141
7.4 Conclusion 142

8 Conclusion 143
8.1 Review of Experiments Performed and Conclusion 143
8.2 Further Work 145

Chapter 1

Introduction

Computers are now far more able to process large amounts of data than they were in the recent past. This increase in capability is due to recent increases in processing speed and the relatively low cost of memory and high-capacity storage devices. With the advent of wide-bandwidth networks, more data is being transmitted around the world than ever before. Recent figures show that the number of machines connected to the principal global academic and recreational computer network, Internet, is growing at an exponential rate [1].

The increase in capacity has unsurprisingly brought diversification. The average desktop workstation is equipped to handle audio and video information as well as text, and modern networking protocols now allow all these different data to be transmitted around the world at speed. The recent World Wide Web initiative has consolidated the range of accessible information and made it easy for anyone to browse data in any of these formats [2].

For the average user, however, it is an inescapable fact that the majority of the content of the information avalanche will not be of the slightest interest. This fact has been recognised in the development of so-called "autonomous agents", computer programs whose task it is to scour available information sources for textual data of potential interest or utility and then present these to the user [3]. An agent becomes increasingly better at its job by observing whether the user reads or rejects the individual items of data it presents, and feeds this information back to improve its model of the user's interests. As the electronic haystack becomes larger and its content more diverse, the search for the needle is proportionally more difficult.

There is no simple way for a user to order an automatic search through an item of non-textual data, such as a video or audio recording, or a collection of such, for a section or item of specific interest. The problem has a familiar analogy; the only way to search through a number of domestic video cassettes for a particular recording is for a human user to perform a manual search through each one until the item is found, or assumed not to be present. In this case, a piece of text assigned to the cassette manually by the user, such as a label, might be of great assistance in directing the search. Analogously, textual tags attached in some way to each non-textual data file would allow a collection of these files to be searched for some specific content, so long as each text tag was sufficiently representative of the content of the file.

However, it is very unlikely that each item in a large non–textual data collection could have its content assessed in this way by a human user, especially if the collection was relatively large, or *dynamic*, with new items added frequently and old ones deleted. To facilitate searching on a data collection of this type, it would therefore be helpful to devise some method of automatically deriving, from non-textual data, some textual tokens, representative of content, that can be stored and searched.

1.1 Speech Recognition

A computer-based technique already exists for deriving representative textual tokens from a certain type of audio recording — automatic speech recognition. Although much progress has recently been made, the performance of speech recognisers is still fragile with respect to factors such as speaker identity, speaking rate, and choice of words. This fragility would affect the hypothetical automatic indexing application by causing the generation of incomplete and errorful sets of textual tokens from the set of audio recordings. However, if the non–interference of these factors can be guaranteed, by ensuring that the recogniser is trained and tested on speech from the same speaker or set of speakers, and that the acoustic characteristics of the speech, (as influenced by the choice of room, microphone and level of ambient noise) are consistent, then the output of the speech recogniser will almost certainly have some utility.

It would seem sensible to gauge the performance of a speech recogniser using two measures of performance; by generating a "detection accuracy" figure by comparing the recogniser output against a list of "true tags" obtained from a manual transcription of the speech, and also by performing searches based on both sets of tags. This is because a missed tag would certainly degrade the detection accuracy of the recogniser, but not necessarily harm the ability of the search system to find items of interest.

1.2 Information Retrieval

Modern methods for the automatic searching of collections of textual items are grouped together under the title of *Information Retrieval (IR)* [4, 5, 6]. In a simple modern IR system, the set of tags for an item in a textual data collection (such an item will hereafter be called a "document") might simply be the list of *content words* (generally nouns, verbs, adjectives and adverbs) appearing in that document, or alternatively, appearing in a more manageable *surrogate* for that document. For example, where the document collection is composed of scientific papers, the abstract conventionally serves as a surrogate for the corresponding paper.

A search is carried out by matching the *query*, an expression derived from the user's *information need*, and again most simply a list of words, against the set of tags for each document. The matching process generates a score for the query against each document's tag–set, where this score might simply be the number of words in common. Ranking the list of documents in decreasing order of score presents the contents of the document collection in descending estimated likelihood of *relevance* to the user's information need. While this description is rather simplified, it reflects the core IR concepts of deriving a representation of a document, matching this against an expression of information need, and returning a set of documents estimated to be of interest to the user.

There are several existing scenarios in which a *spoken message retrieval* system, if sufficiently good, could be put to work. It would be an ideal technique to apply to collections of speech recordings for which it would not be convenient or desirable to obtain a full text transcription of each message, or part thereof, from a human listener. At one end of the scale, it would assist in the management of large sound archives, by providing a better characterisation of the content of a recording than its title; at the other, it would allow users of a voice mail system, or other personal voice–based messaging systems, to retrieve stored items of particular interest.

1.3 Structure of this Thesis

The aim of the work presented in this thesis is to combine the techniques of IR and speech recognition, thereby producing a spoken message retrieval system that can locate items of interest in response to an arbitrary, *ad hoc* expression of information need.

Chapter 2 presents an overview of current methods in speech modelling and recognition based on the Hidden Markov Model (HMM).

Chapter 3 describes the the theory of statistical Information Retrieval (IR). Since the readership of this thesis is likely to be rather cross–disciplinary, these chapters cover the background in more detail than might otherwise be necessary.

Chapter 4 presents a survey of recent papers covering topics related to Spoken Message Retrieval.

Chapter 5 describes a baseline spoken message retrieval system, based on relatively simple speech modelling techniques and no language model, and presents results from an experiment in which it is used to retrieve articles from a collection of BBC Radio News reports.

Chapter 6 presents a series of improvements to the acoustic and language modelling of the baseline system.

Chapter 7 investigates the application of the technique of *relevance feedback* from conventional text IR. It also details an implementation of a significantly different speech retrieval system recently proposed by Glavitsch and Schäuble [7], and compares its performance with that of the earlier system on the spoken message collection.

Finally, **Chapter 8** reviews the work and lists those experimental areas in which future research should be carried out.

1.4 Original Contribution

The principal original contribution of this work is one of the first implementations of a classical IR system based on the output of a conventional speech recogniser. The effect on retrieval effectiveness of the use of differing acoustic models and language models is investigated, and a novel wordspotting technique is used to detect occurrences of arbitrary words as quickly as possible throughout a spoken message collection. In addition, this work presents the first implementation and practical evaluation of Schäuble and Glavitsch's sub–word feature–based retrieval paradigm.

Chapter 2

Hidden Markov Modelling

Hidden Markov Modelling is a widely used method for modelling statistical processes [8]. It is particularly suited to automatic speech recognition, since it is able to encapsulate wide variation in the acoustic realisation of speech sounds, whether this variation is exhibited by a number of different speakers, or by the same speaker, owing to factors such as mood or health. All the speech recognition necessary for the experiments presented in this thesis are based on the Hidden Markov Model.

Hidden Markov Modelling techniques for speech recognition depend on the initial processing of "raw" speech, which has the form of an acoustic pressure waveform, at regular time intervals to extract a sequence of so–called *observation vectors*. This is necessary in order to make speech recognition computationally tractable, and ease the task of decoding a continuous waveform into a discrete set of symbols (corresponding to the individual word or set of words uttered). It is assumed that speech observation sequences corresponding to an acoustic event, for example a word, can be modelled by traversing an underlying sequence of connected states, each associated with an speech vector output distribution. These distributions, and the relative likelihoods of moving between states, are estimated from a number of example vector sequences of the particular word (or other unit) to be modelled. The estimation process is usually termed *training* and the example vector sequences *training data*. Hidden Markov Modelling is a popular technique in speech recognition, because few assumptions need to be built into the models, and all model parameters can be efficiently estimated directly from the training data.

This chapter describes the theories of Hidden Markov Model training and recognition which underpin the experiments described later in the thesis.

2.1 Introduction

It is assumed in the first instance that the recognition task is very simple and consists solely of building and testing a set of speech models, based on the Hidden Markov Model, for a small number of words. In addition, the training and test examples of these words are assumed to be spoken in isolation. The applications of a practical system like this would be extremely limited, but it is a useful starting point at which to introduce the theory, which can then be extended to more sophisticated cases.

It is assumed that a set of speech models M_i for $1 \leq i \leq I$, has been trained for a population of words w_i, and that a sequence of speech vectors $\mathbf{O} = \{\mathbf{o_1}, \mathbf{o_2}, \ldots \mathbf{o_T}\}$ is extracted from a spoken utterance of a word whose identity is not known. Simple speech recognition can be performed by calculating the probabilities of each model M_i having generated observation sequence $\mathbf{O}$, $\Pr(M_i|\mathbf{O})$. By Bayes' Theorem,

$$\Pr(M_i|\mathbf{O}) = \frac{\Pr(\mathbf{O}|M_i)\Pr(M_i)}{\Pr(\mathbf{O})}.$$

It is also assumed that all observation sequences are equally likely. So long as estimates of the probabilities $\Pr(M_i)$ are available, $\Pr(M_i|\mathbf{O})$ can be obtained, since $\Pr(\mathbf{O}|M_i)$ can be calculated directly from the observation sequence $\mathbf{O}$ and the models M_i.

2.2 HMM Anatomy

Figure 2.1 is a diagram of a typical 5–state Hidden Markov Model. An HMM[1] consists of a set of *states*, indexed from 1 to N, a set of probability distributions $\{b_j()\}$ describing the speech vector output of the j'th state, and a discrete probability distribution $\{a_{ij}\}$ describing the set of possible *transitions* between the states. In the figure, the vector–emitting states are represented by the shaded circles, speech vector emission by the arrowed broken lines, and state transition by the arrowed unbroken lines. States 1 and N do not output speech vectors; they are included in the model definition to facilitate the concatenation of sub–word HMMs, where this is necessary in the modelling of large vocabularies.

It is initially convenient to think of a trained word–level HMM, M_i, as a *synthesizer* of output vector sequences corresponding to new examples of the word w_i. The process starts with the occupation of state 1 of the model. At time $t = 1$, a transition

[1] Speech models based on the Hidden Markov Model will be referred to as *HMMs* throughout this thesis.

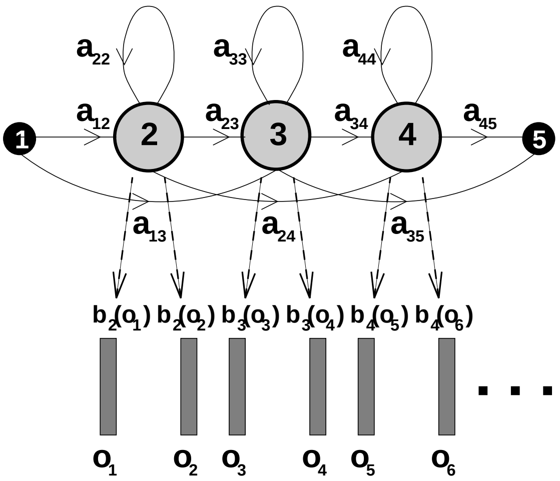


Figure 2.1: The anatomy of a Hidden Markov Model

is taken from state 1 to some state j of the model, depending on the set of transition probabilities $\{a_{11}, a_{12}, \ldots a_{1N}\}$. An observation vector $\mathbf{o_1}$ is now emitted from state j, in accordance with the probability distribution $b_j()$ associated with state j. The process of state–to–state transition and output vector emission is carried out for each time $t \geq 1$, until a transition is taken into state N of the model. The complete vector sequence corresponds to a newly–generated occurrence of the word w_i.

Now, the HMM is used in speech recognition not as a synthesizer but as a *predictor*. Given an observation sequence $\mathbf{O} = \{\mathbf{o_1}, \mathbf{o_2}, \ldots \mathbf{o_T}\}$, representing an occurrence of an unknown word, an HMM M_i for word w_i, and a state sequence $X = \{x(0), x(1), \ldots x(T+1)\}$, where $x(0)$ is the model start state and $x(T+1)$ the final state, then the probability that the HMM generates the whole observation sequence $\mathbf{O} = \{\mathbf{o_1}, \mathbf{o_2}, \ldots \mathbf{o_T}\}$ by traversing state sequence X can be obtained, simply by taking the product of the associated probabilities.

As an example, consider a 5–state HMM with 3 vector–emitting states. If an observation sequence $\{\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3}\}$ is generated by each of the emitting states generating a single observation vector in turn, and this simple state sequence is labelled X, then

$$\Pr(\mathbf{O}, X|M_i) = a_{12}b_2(\mathbf{o_1})a_{23}b_3(\mathbf{o_2})a_{34}b_4(\mathbf{o_3})a_{45}.$$

However, in recognition, every *allowable* state sequence may be responsible for the generation of a particular observation sequence. Therefore, the probability $\Pr(\mathbf{O}|M_i)$ that an HMM M_i generates an observation sequence $\mathbf{O}$ along *any* state sequence is obtained by summing probabilities over all such state sequences.

Explicitly,

$$\Pr(\mathbf{O}|M_i) = \sum_X a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(\mathbf{o_t}) a_{x(t)x(t+1)}$$

where $X = \{x(0), x(1), \ldots x(T+1)\}$ is a general state sequence through the model, again with $x(0)$ and $x(T+1)$ constrained to be the model entry and exit states respectively.

Each HMM M_i can therefore be viewed as a potential generator of the unknown word observation sequence **O**. Word recognition can now be performed by calculating $\Pr(\mathbf{O}|M_i)$ for each HMM M_i, multiplying by the prior probability of the occurrence of each word w_i, and choosing the word w_i for whose index i the product of probabilities is maximised.

In speech recognition tasks where the HMMs are used to represent *phones* or other *sub-word* units of speech production, the use of three emitting states is widely deemed to be sufficient; in whole word model training, however, there is no consensus on the optimum number of states. Instead, this is generally determined empirically, or from other factors, such as the amount of available training speech. The set of possible state transitions for a model defines its *topology*. In the simple *left–right* model topology, transitions are allowed only from state n to itself and to the single successor state $n+1$. Lee performed experiments to determine the effect on recogniser performance of tailoring model topologies, but found that for continuous speech recognition, the choice of model was not critical [9]. In general, enriching the model topology allows for greater flexibility of speech modelling, but at the expense of having to train a greater number of transition probabilities and output distribution parameters.

2.3 Output Probability Distributions

A popular method of modelling the distribution of the output of each emitting state of an HMM is the use of multivariate Gaussian distributions [8]. A model in which the output of a state is modelled by a single multivariate Gaussian distribution is known as a *single Gaussian* model. Single Gaussian models perform poorly in recognition, since they do not encapsulate the acoustic variability of a speech event sufficiently well. Therefore, it is common to model state outputs as the weighted sum of a number of Gaussian distributions. Such models are known as *Gaussian mixture* models.

In a single Gaussian model, the output of state j is described by the equation

$$b_j(\mathbf{o_t}) = \mathcal{N}(\mathbf{o_t}; \mu_j, \Sigma_j)$$

where μ_j and Σ_j are the mean vector and covariance matrix of the multivariate Gaussian distribution $\mathcal{N}$. $\mathcal{N}$ is defined by the equation

$$\mathcal{N}(\mathbf{o_t}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(\mathbf{o_t}-\mu_\mathbf{j})'\mathbf{\Sigma}_\mathbf{j}^{-1}(\mathbf{o_t}-\mu_\mathbf{j})}$$

The corresponding equation for Gaussian mixture modelling is

$$b_j(\mathbf{o_t}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o_t}; \mu_{jm}, \Sigma_{jm})$$

where M is the number of Gaussian mixtures used to model the output of state j, μ_{jm} is the mean vector of mixture m in state j, Σ_{jm} the corresponding covariance matrix, and the mixture weights c_{jm} sum to 1.

2.4 Model Training

Training is the process in which the *parameters* of a model, namely the state transition probabilities and means and covariances of the state output distributions, are estimated from a set of example observation sequences corresponding to the word or words to be modelled. This section will describe techniques applicable to the training of Gaussian mixture whole–word HMMs.

The ability of a set of HMMs to recognise some unknown speech depends on how well they have been trained. In order to produce well–trained HMMs it is necessary to have sufficient data from which the model parameters can be estimated satisfactorily. If not enough training speech is available, models will not be sufficiently representative of the speech they are meant to model and the recognition performance of these models will be very poor. In general, the more individual parameters there are to be trained in an HMM, the larger the required amount of training data.

The training process typically consists of obtaining an initial estimate of the model parameters, and iteratively improving the model set by generating better estimates of these parameters, with respect to the training data, until some convergence criterion is reached. This iterative process is called *re–estimation*.

2.4.1 Initial Model Estimation

It is crucial to obtain good initial estimates of HMM parameters because model parameter re–estimation procedures typically guarantee convergence only to a locally optimal solution. The initialisation method used in this thesis is illustrated in Figure

2.2. It is based on simply dividing each utterance in the training data into roughly equal segments, each one corresponding to an emitting state of the HMM, and assigning to each state an initial mean vector and covariance matrix obtained by pooling the speech vectors within the corresponding segment. State transition probabilities are preset and unaltered by the initialisation process.

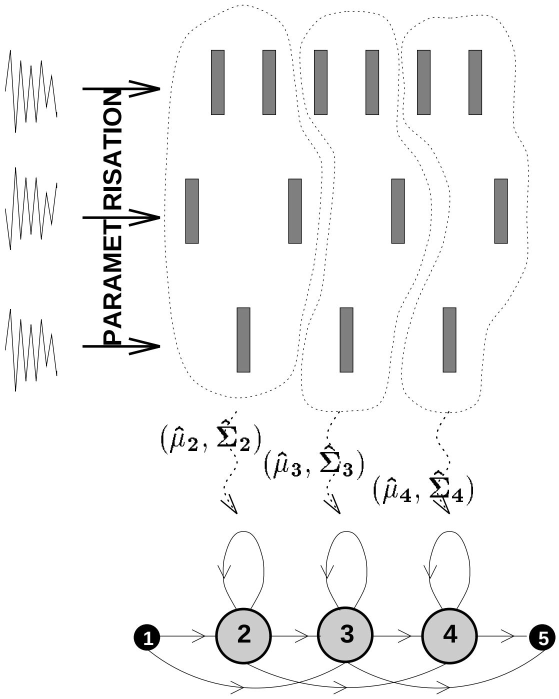

PARAMETRISATION
$(\hat{\mu}_2, \hat{\Sigma}_2)$
$(\hat{\mu}_3, \hat{\Sigma}_3)$
$(\hat{\mu}_4, \hat{\Sigma}_4)$
1
2
3
4
5

Figure 2.2: Initial uniform segmentation of training data

The uniform segmentation of the training utterances gives rise to a state sequence in which each state is occupied roughly the same number of times. The next stage of initial model estimation is to generate an improved "fit" of the speech vectors to the model states by calculating the most likely sequence of states through the model that corresponds to each training data sequence. This method is called *Viterbi alignment*. Each training utterance is re–segmented in accordance with this alignment, and new means and covariances based on the re–segmentation are assigned to the model states. The process ends when the change in successive values of the probability $\Pr(\mathbf{O}|M_i)$ falls below a threshold.

2.4.2 Model Parameter Re–estimation

Section 2.1 introduced the concept that if a sequence of speech vectors $\{\mathbf{o_1}, \ldots \mathbf{o_T}\}$ is observed, then *every* possible sequence of states through an HMM M contributes to the *total probability* that the observation sequence was generated by model M. In theory therefore, for every speech vector $\mathbf{o_t}$ and every model state j, there is some state sequence in which the vector $\mathbf{o_t}$ is emitted from state j with some probability, however small. Therefore, in model re–estimation, it is a core assumption that every observation vector contributes towards the the re–estimation of the parameters of every state in the model. The re–estimation method used by the experiments in this thesis is the *Baum–Welch re–estimation process.*

Assuming that a set $\{\mathbf{O}^r : 1 \leq r \leq R\}$ of training observations is available, the main step of the Baum–Welch process is the calculation of the total likelihood of occupancy of each state j, at each time t, for each observation sequence $\mathbf{O}^r$. This likelihood, written $L_j^r(t)$, can be obtained efficiently using the *Forward–Backward Algorithm*, which is described below in Subsection 2.4.3. Now, the earlier model initialisation method assigned observation vectors directly to states. Here, however, once the state occupancy likelihoods have been obtained, an observation vector is assigned to a state in proportion to the likelihood of that state being occupied when the vector was observed. New estimates of the mean vector μ_j and covariance matrix Σ_j can therefore be obtained from the equations

$$\hat{\mu}_\mathrm{j} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_j^r(t) \mathbf{o_t^r}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_j^r(t)}$$

and

$$\hat{\Sigma}_\mathrm{j} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_j^r(t) (\mathbf{o_t^r} - \mu_j)(\mathbf{o_t^r} - \mu_j)'}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} L_j^r(t)}.$$

2.4.3 The Forward–Backward Algorithm

The Forward–Backward Algorithm allows the model state occupancy likelihoods $L_j(t)$ to be calculated efficiently, and is instrumental in allowing the Baum–Welch algorithm, described above, to generate locally optimal HMM parameters.

First, the *forward probability* $\alpha_j(t)$ is defined as the joint probability that speech vectors $\{\mathbf{o_1}, \ldots \mathbf{o_t}\}$ are observed and that the underlying state sequence $\{x(1), \ldots x(t)\}$ traversed during this observation ends at state j of model M. Formally,

$$\alpha_j(t) = \Pr(\mathbf{o_1}, \ldots, \mathbf{o_t}, x_t = j|M).$$

The initial conditions are set as follows;

$$\alpha_1(1) = 1$$

and

$$\alpha_j(1) = a_{1j} b_j(\mathbf{o_1}),$$

where a_{ij} is the probability of transition from state i to state j. The forward probabilities for the vector–emitting states (*i.e.* states 2 to $N-1$) of the model can now be obtained by the recursion

$$\alpha_j(t) = [\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij}] b_j(\mathbf{o_t}),$$

which represents the total probability of transition into state j, and emission by that state of speech vector $\mathbf{o_t}$, summed over all possible predecessor states. The probability of transition into the final state of the model, when all the speech has been observed, is obtained by the equation

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}.$$

$\alpha_N(T)$ is, in fact, the total probability of the unknown observation sequence for the model M.

Analogously, the *backward probability* $\beta_j(t)$ is defined to be the probability that beginning at model state j at time t, speech vectors $\{\mathbf{o_{t+1}}, \ldots \mathbf{o_T}\}$ will be observed. With the initial condition

$$\beta_i(T) = a_{iN},$$

this probability can also be computed recursively, by the formula

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(\mathbf{o_{t+1}}) \beta_j(t+1),$$

and the final condition is given by

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\mathbf{o_1}) \beta_j(1).$$

From the definitions of the forward and backward probabilities, it is possible to obtain the total probability of occupation of any state j at any time t, since

$$\Pr(\mathbf{O}, x(t) = j | M) = \alpha_j(t) \beta_j(t).$$

Finally, the state occupancy likelihood $L_j(t)$ is obtained as follows.

$$\begin{aligned} L_j(t) &= \Pr(x(t) = j|\mathbf{O}, M) \\ &= \frac{\Pr(\mathbf{O}, x(t) = j|M)}{\Pr(\mathbf{O}|M)} \\ &= \frac{\alpha_j(t)\beta_j(t)}{\Pr(\mathbf{O}|M)} \end{aligned}$$

Adopting the convention that $\{L_j^r(t)\}$ is the set of state occupancy likelihoods obtained for observation sequence $\mathbf{O_r}$, the parameters of the state output distributions can now be re-estimated.

2.4.4 Transition Probabilities

The transition probabilities can also be re–estimated using the forward and backward probabilities. The re–estimated probability of transition between two vector–emitting model states i and j is simply the ratio between the expected number of times that a transition is made from i to j, and the expected number of times state i is occupied. Formally,

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P^r} \sum_{t=1}^{T_r-1} \alpha_i^r(t) a_{ij} b_j(\mathbf{o^r_{t+1}}) \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P^r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)}$$

for $1 < i < N$ and $1 < j < N$, where N is the number of states in the model and $P^r = \Pr(\mathbf{O^r}|M)$ the total probability of the r'th observation. The probability of transition from the start state is re–estimated by

$$\hat{a}_{1j} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P^r} \alpha_j^r(1) \beta_j^r(1)$$

for $1 < j < N$, and the probability of transition into the final state by

$$\hat{a}_{iN} = \frac{\sum_{r=1}^{R} \frac{1}{P^r} \alpha_i^r(T_r) \beta_i^r(T_r)}{\sum_{r=1}^{R} \frac{1}{P^r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)}$$

where $1 < i < N$.

2.4.5 Extension to Gaussian mixture training

The extension of the Baum–Welch algorithm to Gaussian mixture models, which offer considerably improved recognition accuracy, is relatively simple. It depends on thinking of each state j as being composed of several *substates*. Each substate i has an output distribution given by the i'th Gaussian component of state j, with the transition probabilities into the substates obtained by multiplying the original transition probabilities by the Gaussian mixture weights. This scheme is illustrated in Figure 2.3[2]. The substate occupancy likelihood function for the m'th mixture of state j for observation sequence $\mathbf{O_r}$ is then obtained by substituting a revised form of the forward probability calculation into the usual state occupancy likelihood formula:

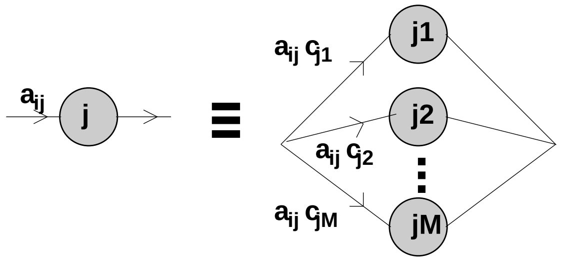


Figure 2.3: A Gaussian mixture state as a network of substates.

$$L^r_{jm}(t) = \frac{1}{P^r}[\sum_{i=2}^{N-1} \alpha^r_i(t-1)a_{ij}]c_{jm}b_{jm}(\mathbf{o^r_t})\beta^r_j(t)$$

for $t > 1$, with the initial condition

$$L^r_{jm}(1) = \frac{1}{P^r}a_{1j}c_{jm}b_{jm}(\mathbf{o^r_1})\beta^r_j(1).$$

Re–estimation then follows as for the single Gaussian case, with the means and covariances of each substate being obtained individually, and new mixture weights calculated by the formula

$$\hat{c}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{jm}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L^r_{j}(t)}.$$

2.4.6 Extension to embedded re–estimation

The previous section described techniques for the training of Gaussian mixture whole–word HMMs. Whole–word models are suitable for a number of speech recognition applications, such as fixed–vocabulary *wordspotters*, which detect occurrences of a small number of words in running speech. They are not suitable, however, for large vocabulary systems, such as continuous speech recognisers, since these systems have

[2] This figure is taken from [10].

vocabularies containing tens of thousands of words; the building of whole–word models for the entire vocabulary would be computationally highly expensive and would rely on impractically huge amounts of acoustic training data.

For this reason, continuous speech recognisers employ word models created by concatenating HMMs of sub–word units. A number of differing sub–word units have been proposed in continuous speech recognition experiments in recent years; the experiments described in this thesis are all based on the *phone* [9]. Sub–word modelling is desirable principally for two reasons; firstly, an HMM for any word can be built from the appropriate sub–word models, and secondly, since the data for sub–word model training are pooled from all the utterances in the training set, sub–word HMMs can all be well trained, so long as the training set is sufficiently big.

It is possible to train a set of sub–word models completely by excising the set of observations for each sub–word unit from the training corpus and using the Baum–Welch algorithm as already described. However, this method sets fixed boundaries on the start and end of each phone occurrence. This intrinsically limits the amount of data that can contribute to the re–estimation of each sub–word model, and does not make best use of the available training data. Therefore, once initial model estimates have been generated, the usual strategy for training sub–word HMMs is to concatenate the appropriate models together, creating a single HMM spanning the entire utterance, and then performing Baum–Welch re–estimation over the entire observation sequence.

For each sub–word unit, the numerator and denominator of the re–estimation equations are accumulated as each observation sequence is encountered; when all the data have been observed, the results of the accumulation are used to obtain the new model parameter estimates. The primary modification to the Baum–Welch equations outlined in subsection 2.4.2 is to allow α and β values to be passed through the concatenated sequence of HMMs for the training speech. This is done by propagating them through the initial and final states of each sub–word HMM. The transition probability re-estimation equations are revised to reflect the fact that the entry state of each model can now be occupied at any time. The modified equations can be found in [10].

2.5 Speech Recognition using HMMs

As has been mentioned, it is possible to recognise an isolated example of a word, $\mathbf{O} = \{\mathbf{o_1}, \mathbf{o_2}, \ldots \mathbf{o_T}\}$ by obtaining $\Pr(\mathbf{O}|M_i)$, the total probability of the unknown speech for each model M_i, multiplying this by the prior probability of the occurrence of each word, and choosing the index i which minimizes the product. However, this method does not generalise to *connected* speech recognition, in which it is necessary to obtain the single best *sequence* of models matching an unknown utterance, due to the explosion in the number of state sequences.

Instead, it is usual to perform recognition by obtaining the single *most likely state sequence* through a model or set of models for an utterance. The method is similar to that for obtaining the total probability for an utterance except that the summation is replaced by a maximisation. This is known as the *maximum likelihood* or *Viterbi* method [11].

Assume that for some single HMM M, $\phi_j(t)$ is the maximum (as opposed to total) likelihood of observing speech vectors $\{\mathbf{o_1}, \mathbf{o_2}, \ldots \mathbf{o_t}\}$ and traversing some underlying state sequence ending in state j. Then

$$\phi_j(t) = \max_i(\phi_i(t-1)a_{ij})b_j(\mathbf{o_t})$$

with the initial conditions

$$\phi_1(1) = 1$$

and

$$\phi_j(1) = a_{1j}b_j(\mathbf{o_1}),$$

and with the maximum likelihood through the whole model obtained by

$$\phi_N(T) = \max_i \phi_i(T)a_{iN}.$$

To avoid the problem of computational underflow in the calculation of these probabilities, it is usual to perform all calculations in the log domain. Transforming the main maximum likelihood equation in this way produces

$$\psi_j(t) = \max_i(\psi_i(t-1) + \log(a_{ij})) + \log(b_j(\mathbf{o_t}).$$

The maximum likelihood method, unlike the total likelihood method, has a simple extension to connected speech recognition. One method is the Token Passing Paradigm of Young *et al* [12]. In the Token Passing Paradigm, word models are built from the corresponding sub-word units and a finite state network constrains the allowable word sequences that can be generated by the recogniser. At every time t, each model state in the network contains a single *token* which represents the maximum likelihood path through the network, ending in that state, for the speech vectors observed up to that time. The set of tokens is updated in accordance with the above maximum likelihood equation every time a new speech vector is observed. The arrival of a token into the exit state of a word model M_i corresponds to the hypothesis by the recogniser of word w_i ending at time t. The token is *propagated* to the entry states of all the models of words allowed by the finite state network to follow word w_i. Figure 2.4 illustrates the action of the finite state network as a *grammar*, constraining output word sequences to belong to the alphabet of the grammar. An alternative method of

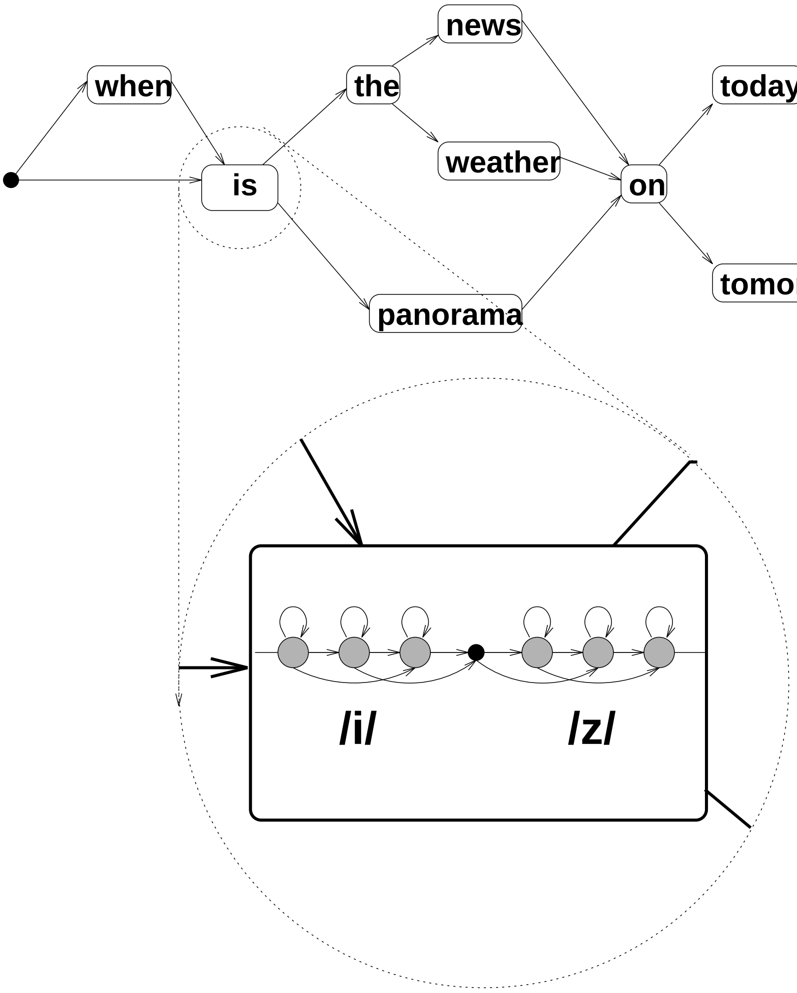


Figure 2.4: An example recognition network based on sub–word HMMs

constraining the output word sequence is to weight word–to–word transitions by a score that favours likely word pairs and penalises unlikely ones. These scores are derived from word pair (*bigram*) probabilities, which may be estimated from the word sequences observed in the acoustic training data, or from some other suitable data source.

All such word hypotheses are recorded, so that when all the unknown speech has been observed, the single token in the exit state of the network can be examined and its path traced back through the network to produce the best sequence of word models.

2.6 Limitations of Hidden Markov Modelling

The Hidden Markov Model provides an elegant framework for the development of automatic speech recognisers, and is still generally superior to other recognition frameworks such as neural networks [13]. Several recognition systems based on the Hidden Markov Model are now commercially available as "shrink–wrapped" products, with the emphasis placed on the use of speech recognition as an alternative to keyboard input into personal computers. Despite the general success of the Hidden Markov Model, however, it has a number of inherent flaws.

A particular problem is in the modelling of state occupancy durations. The many utterances of a single word necessary for whole word training may vary a great deal in duration, especially when the word is spoken by more than one speaker. It would seem sensible for the modelling process to ensure that the most likely time spent in each model state during recognition was related to the number of observation vectors assigned to the state in training. However, in the underlying Markov process, the most likely amount of time spent in each state is always the duration of one speech vector. This is because the implicit state distribution, as specified by the transition probabilities, is geometric, and characterised by exponential decay. The actual effect in recognition of the state transition probabilities is quite small, since their contribution of the transition probabilities to the log path score $\psi_j(t)$ is considerably outweighed by the state output likelihoods.

A number of improved durational models have been proposed. One is to clone each emitting state into three or four substates with smaller self–transition probabilities. The output distributions of each state are "tied" together so that the increased number of states does not require a correspondingly increased amount of training speech [14]. Alternatively, state durations can be explicitly modelled using a discrete distribution such as a Poisson or Gamma distribution, and the recognition method modified to incorporate transition probabilities estimated from these distributions. The improvement in recognition accuracy is generally counterbalanced by the increased computational complexity of training a large number of durational parameters [15].

Also, it is known that speech, when viewed as a statistical process, is highly correlated in time, since it is produced by continuous movements of the articulators (the tongue, lips, teeth *etc*). Therefore, subsequent observation vectors are likely to be statistically dependent. However, HMMs are first–order models, so the probability of speech vector emission, at any time for any state, depends only on the speech vector and the identity of the state; it is independent of the state occupation history or any of the previously observed speech vectors. This mismatch between the observed phenomena and the statistical model can in this case be reduced to some extent by the addition of rate–of–change information to the parametrized speech vectors, but this is a rather crude solution to the problem and does not correct the underlying fault in the analysis.

2.7 Conclusions

This chapter has described the underlying theory of Hidden Markov Modelling at both the whole–word and sub–word level. It has also described the application of the technique to continuous speech recognition. Hidden Markov Modelling will form the theoretic backbone of the recognition system used in subsequent experiments in spoken message retrieval.

Chapter 3

Information Retrieval

"Information Retrieval" (usually abbreviated to *IR*) can be used as a general phrase to describe the extraction of information held in modern computerised data collections. In this thesis, however, it is taken to refer to the set of experimental techniques that have been developed to address a specific problem — the design and testing of automatic systems for locating items of specific interest within a collection of textual documents. Modern retrieval collections are generally so large that it is not convenient or possible for a single human user to examine the collection exhaustively and select required items by hand. The issue of collection size motivates the development of effective retrieval techniques, but is also a significant problem in retrieval system *testing*, as will be discussed later. The documents that make up a collection are usually related by a broad subject area. For example, the Medlars test [16] was based on a collection of 700,000 documents from the field of medicine, and other large collections have been built up in the areas of chemistry and law. Recently, a collection of a million documents has been amassed for the yearly US Advanced Projects Research Agency (ARPA) TREC evaluation experiments [17, 18, 19].

IR considerably predates the modern computers whose processing and storage power nowadays enable sophisticated retrieval systems to be implemented. Cleverdon points out its roots in the manual card–index comparison methods of the Fifties, and comments that the first use of a computer in document retrieval was actually not much of an advance, since it was largely based on the old manual methods, and offered no speed advantage [20]. However, the development over the last thirty years of modern computing power has both enabled and motivated the experiments whose conclusions underpin most current thinking about IR. Many different approaches to retrieval have been attempted over this period; some linguistically motivated, some based on probabilistic or statistical models. However, certain features have been common to them all. The following section presents a brief overview of these core concepts.

3.1 An introduction to IR

For documents to be automatically retrieved from a collection, each document must be available to the IR system in some electronic form. This form need not always be related to the exact document texts. This may be for the simple reason that the whole document collection is too large to be stored in its entirety, although nowadays it is far more likely that the *full text* of documents in a collection is available for searching. In the recent past, it has been common to use some short piece of text, known as a *surrogate* to stand in for the document. In the case where the document is a scientific paper, the surrogate is usually the abstract of the paper, since it encapsulates the document content in a relatively short form. There has in fact been some recent work concerning the problem of how to obtain an automatic summary of a document when no abstract is explicitly provided [21].

Whether in full text or surrogate form, a document is processed, on its addition to a collection, to obtain a so–called *document representation* or *document description*. As the first name suggests, this is the form of the document which represents it in automatic retrieval. It must reflect the key information contained in the document in a form which can be matched against a representation of information requirement. At the simplest level, an automatically extracted document representation may be a list of words or word fragments, known as *terms*, extracted from the full or surrogate text; a more complicated approach might involve the extraction of phrasal units, or the use of linguistic and knowledge–based methods to build a higher–level content representation [22]. It shall be assumed for the purposes of this brief overview that document representations are simple lists of words extracted from the appropriate documents. The process of obtaining a document representation, however this is done, is known as *indexing*.

The goal of any retrieval system is to satisfy the ineffable *information need* of some user of the system. To achieve this, it is necessary to obtain a verbalized expression of the information need. This expression is known as the *request*. The request is usually derived from user keyboard input, and must be processed to produce a *query*. The query is the expression of the user's information need that will be used in retrieval to obtain documents from the collection. Analogously to the processing of documents, the query can be thought of as the *request representation*. The query terms may have been combined, by the user, using the logical operators *AND*, *OR* and *NOT*, to form a complex *Boolean query*. If not, they may be automatically associated with some numbers to reflect their relative importance. This association is called *term weighting*.

The last stage of the main retrieval process is the comparison of the query against each member of the set of document representations. This should return a subset of the document collection containing those documents whose representations *match* the query, possibly to some varying degree. Where query terms are conjoined using Boolean operators, the set of matching document representations consists of those which simply contain the query terms in the specified Boolean combination. Where query terms are weighted, the matching procedure returns, for each document representation, some function of the weights of the terms shared between the query and the document.

A retrieval system is judged on its ability to present to the user those documents which are *relevant* to the query, whilst suppressing those that are not. The concept of document *relevance* is at best vague, and the subject of some debate, but has been loosely defined by Robertson [23] as the extent to which a document satisfies the user's underlying information requirement. Clearly, some proportion of the retrieved documents must be assessed, however informally, when the search has been carried out, so that the user can discard the uninteresting information and make best use of what remains. If the assessment of documents is formalised, then the assessments can be exploited to modify the initial query. This might either involve assigning weights to existing query terms, to reflect how good they appear to be in discriminating relevant from irrelevant documents, or adding new terms to the query. Matching can then be performed again, against the remainder of the document collection, using the newly modified query. For reasons which should be apparent, this useful technique is known as *relevance feedback*.

The remainder of this chapter is concerned with a deeper examination of the central principles of Information Retrieval.

3.2 Indexing

3.2.1 Manual Indexing

Indexing is the process of extracting from a document an expression of its content for use in retrieval. The first widely recognised experiments in IR utilised document representations obtained using manual indexing, that is, the assignment of index terms to a document by a skilled human indexer. The technique has long since fallen out of favour due to the expense involved in indexing larger and larger document collections, and the lack of consistency in the assignment of indexing terms to document by different indexers [20]. Automatic indexing, the use of computer programs to analyse a document and obtain a representation using primarily statistical methods, has now largely replaced manual methods in experimental retrieval systems. However, there are many operational (*i.e.* commercially available) systems in which retrieval is performed on manually–derived document representations.

3.2.2 Automatic Indexing

By its nature, manual indexing was to some extent grounded in semantic and linguistic extraction of document index terms, as well as more straightforward syntactic analysis. Salton's attempts to investigate the use of intuitively plausible linguistic methods in the *automatic* indexing of document surrogates, produced such unsatisfactory results that the bulk of research into automatic indexing since has been concerned solely with statistical techniques [16]. More recently, Sparck Jones and Tait [24] applied powerful language analysis techniques to extract phrasal terms from queries and automatically generate syntactic variants of these, in order to determine whether the inclusion of these variants in the matching process improved retrieval performance. Now that the on–line storage of full document texts is feasible, some writers are optimistic about the potential of natural language processing (NLP) techniques in a future generation of retrieval systems [22, 25].

The fundamental unit of automatic statistical indexing is the word. Each word in a document can be thought of as discriminating, to some (potentially quantifiable) extent and possibly negatively, between that document and the remainder of the document collection. During the 1950s, Luhn [26] suggested that from a retrieval point of view, the most discriminating words in a document were those that occurred with relatively medium frequency. High frequency words (such as pronouns, conjunctions *etc*) could not possibly characterise the content of a document; also, occurrences of low frequency words, despite their potential discriminative power, were so unlikely that they would seldom be contained in a query presented to a retrieval system. This observation laid the foundations for frequency–based techniques for term extraction.

In fact, instead of extracting terms based on their frequency of occurrence, it is far more common to use all the terms in a document, attaching a frequency–related weighting to each one, with the exception of the high–frequency function words. They are eliminated by means of a "stop list" consisting of a large number of function words, plus any words which might not be particularly discriminating, given the subject area of the document collection (for example, "program" where the document collection is composed of computer–related articles). Stop lists have been used to reduce the amount of storage space and memory required to implement retrieval systems.

It is a problem of English, and all other inflected languages, that the same underlying concept can be represented by any number of similar but not identical words (for example, the singular and plural forms of most English nouns differ). This poses a problem for matching functions which look for exact occurrences of query terms in document representations. Several techniques have been proposed to alleviate this problem, and they are based on mapping words into a set of word family *descriptors*, each of which is broadly indicative of some meaning. Methods for identifying variant word forms are called *conflation*.

To illustrate, a successful technique of this kind should be able to map each of the words "house", "houses" and "housing" into the same word family descriptor. A common morphological technique is called *stemming*, or *suffix–stripping*, and is based on the rule–based removal of the derivational and inflection suffixes of a word. This reduces a word to a word–stem which is not necessarily a word in itself. If this technique were applied to the set of three related words above, it should be able to reduce each one down to "hous". Of course, the agglomeration of suffixes in English does not rigidly obey a set of rules, so it is not generally possible to implement a stemming system that operates with 100% accuracy. Porter's algorithm is an accurate and widely–used stemming method [27]. Not all conflation methods, however, are based solely on right–hand truncation. A more general method for word clustering involves the comparison of two words based on the number of *trigrams*, that is, 3–character substrings, the words have in common. It is therefore able to identify words regardless of whether they differ in prefixing or suffixing [28].

Stop word removal and suffix stripping, are performed, as Lewis [22] points out, to achieve two conflicting goals – those of *discrimination* and *normalisation*. Documents must be discriminated from each other so that individual items can be retrieved from the collection; on the other hand, they must be normalised so that successful query–document matches can be made where the query terms match the documents on concept but not on exact term morphology. The choice of stemming algorithm is one of the factors affecting the point at which the two goals trade off. Non–optimal indexing will always occur; Kupiec [29] illustrates this with reference to the book and film title "Gone With the Wind", which is reduced by standard indexing techniques to "go wind". Over–normalisation and under–discrimination are just two of the problems that plague automatic indexing; others include how to deal with misspellings, inappropriate hyphenation, and the use of alpha–numeric terms such as "1990s". However, in the indexing "balancing act", more is generally gained in normalisation than is lost in discrimination.

3.3 Queries and Relevance Assessments

Commercial retrieval systems are, on the whole, based on Boolean query formulations; the combination of search terms by the Boolean logical operators *AND*, *OR* and *NOT*. Willett [5] claims that retrieval system users are typically not sufficiently good at performing successful Boolean searches, and require a trained searcher to assist in query formulation. Boolean querying has recently been supplanted, at least experimentally, by so–called *best–match* searching, in which a numerical measure of query–document similarity is calculated for each document for a given query. A best–match measure could simply be the number of terms in common between a query and a document. Best–match searching offers many advantages over Boolean searching. For example, the simple nature of query formulation obviates the need for user assistance or training. In addition, documents can be presented in decreasing order of relevance, so the user looking only for one or two specific documents is more likely to find them quickly, assuming they are at the top of the ranked list. Finally, the well–understood method of relevance feedback allows the exploitation of assessments to produce an improved query. For these reasons, all the experiments in this thesis are based on best–match searching.

Best–match searching offers considerable ease of query formulation. Query terms can be extracted from a natural language sentence request, for example "Find me all documents concerning Premiership football". This is a form in which new users may find it easy to phrase information needs. Alternatively, requests may be supplied in a denser form, for example, "football premiership clubs match goals score", which in this case contains a greater number of potentially useful search terms. The use of a stop list allows the syntactic "fluff" of the natural language sentence to be removed, so the matching process is essentially invariant of the original format of the request.

A query may be *expanded*, either before retrieval has initially been carried out, or afterwards, so that adjustments can be made in the light of its initial performance (this case will be discussed in some detail later). A statistical or linguistic *thesaurus* is one method of query expansion before retrieval. Looking up the query terms in a thesaurus allows the addition to the query of new terms, with similar meanings to the original terms, to increase the number of term matches between the query and the relevant documents. However, thesaurus look–up seems to be another of the intuitively plausible techniques which do not improve the effectiveness of a retrieval system. In the 1993 TREC–2 evaluations, results showed that query expansion independent of relevance information did not appear to offer any advantage [19].

The design and testing of differing retrieval strategies in experimental systems depend on the invariance with time of the query formulations and the users' underlying information needs which gave rise to these queries. This assumption allows the effect of varying experimental factors such as stemming, the size of the stop–word list, the use of term weighting *etc* to be investigated in a controlled manner. Therefore, it is important to obtain queries and corresponding relevance assessments *before* any experiments are carried out. The lists of documents retrieved by differing systems can then be compared with the list of documents *known* in advance to be relevant. The queries and lists of relevant documents are thus treated as "snap–shots" of users' information needs.

The "best–case" scenario is that queries are sourced from users' real information needs, and that all documents in the collection are assessed for relevance to all the queries. However, document collections are typically very large indeed, and factors such as expense and user fatigue unfortunately prevent the exhaustive advance assessment of the collection to obtain lists of relevant documents. There are a number of ways around this problem. One such method is the performing of a number of searches on the collection using differing queries generated from the same request. The output of these searches are *pooled* and the documents in the resulting pool assessed for relevance, with unassessed documents assumed to be non–relevant. The pooling method is used in the ARPA TREC evaluations [17, 18]. The larger the number of queries for which retrieval output is pooled, the greater the coverage of the true document set relevant to the request.

Pooling is an example of how *user–domination* of requests and document assessments is tempered with *experimenter–dominated* factors, thereby trading off some experimental "authenticity" against tractability. Here, the experimenter–dominated factors are related to the size of the document pool eventually assessed for relevance. In Tague's analogy [30], the space of possible experiments is a 1–dimensional continuum, with user–dominated queries at one extreme, experimenter–dominated at the other, and most real experiments lying somewhere in between; however, since there are many factors in the querying process that can be thought of as independent (for example, queries and relevance assessments can unfortunately sometimes not be obtained from the same user), the experimental space is maybe best thought of as multi–dimensional.

An example of a real retrieval experiment is the classic Cranfield 2 work carried out during the mid–1960s [31]. Beginning with a base set of some 221 scientific papers on aeronautics, a query was constructed for each base document by asking the authors to pose the questions which their papers had set out to answer. Each paper cited by each base document was assessed by the author for relevance to the query derived from that document. These relevant sets were made complete with respect to the set of 1400 documents that resulted from this procedure, by soliciting relevance assessments from skilled judges. This is an example of a retrieval task involving *exhaustive* relevance assessments obtained from a number of different judges, and semi–artificial queries. In contrast, experiments to analyse the performance of the Medlars system were based on 300 genuine queries; with a collection, however, of some 700,000 documents, there was no possibility for exhaustive assessment. In experiments on this scale, methods like pooling must be used. These two systems can be represented as different points in the multi–dimensional experimental space.

Finally, as Tague points out, it must borne in mind that the group of users involved in any retrieval experiment is essentially *self–selecting*. They are typically either those people particularly knowledgable of the main subject area addressed by the document collection, or those prepared to spend some time supplying queries and assessing documents.

3.4 Query–Document Matching

Best–match searching is a quantitative approach to retrieval. It differs in this respect from Boolean matching, which simply finds the subset of the document collection exactly matching the query expression, and presents them, unranked, to the user. Best–match searching depends on some method of comparing the query to the members of the document collection and returning a *query–document score* reflecting this comparison. The more similar a query to a document, the higher the score should be. As mentioned earlier, a simple scoring method is to count the number of terms a query and a document have in common. This can be formalised by expressing both the query and the document representation as vectors of length I, where I is the number of terms in some indexing vocabulary $t_1, t_2, \ldots t_I$ of terms. We write

$$\mathbf{Q} = (b_Q^1, b_Q^2, \ldots b_Q^I)$$

and

$$\mathbf{D} = (b_D^1, b_D^2, \ldots b_D^I),$$

where b_Q^i is either 0 or 1, depending on the absence or presence of term i in the query $\mathbf{Q}$, and the document vector elements are defined similarly. The binary query–document similarity measure is then expressed as

$$sim(\mathbf{D}, \mathbf{Q}) = \sum_{i=1}^{I} b_D^i b_Q^i.$$

This similarity measure is known as the *co–ordination level* or *quorum* matching function.

The expression of the relatively simple concept of counting common terms in such a complicated manner is given in order to illustrate that the elements of the query and document vectors need not be restricted to the values 0 and 1. It is intuitively fairly plausible that query or document terms could be assigned values to represent their relative "importance" (however this is defined) and that this would be likely to improve user satisfaction by ranking documents containing some number of important terms higher than those containing an identical number of less important terms. This association is called *term weighting*, and it is generally accomplished using automatic frequency–based or probabilistic schemes.

3.4.1 Inverse Document Frequency Weighting

Assume for the moment that in some hypothetical retrieval environment, document representations are lists of unstemmed words obtained after high–frequency stop–words have been removed, that query–document similarity is based on a co–ordination level match, and that the request "house of commons" is submitted. Those documents containing both *house* and *commons* would be ranked at the top of the list of retrieved documents with a similarity score of 2, but those documents containing only one term would be ranked together with score 1, regardless of which term was matched on. In reality, one of these terms is far more useful than the other in this context; the word *house* on its own is too frequent in English to match reliably on the required topic by itself, but *commons*, as a noun plural form, is relatively rare, specifically meaning either the House of Commons of the UK Parliament (as in "The Commons"), or "pieces of land in public ownership and use". In this case, it would be desirable to assign to each query term a weight varying inversely with its frequency in some amount of text. Taking this text to be the whole of the collection from which documents are to be retrieved, gives rise to the *inverse document frequency* or *collection frequency* weighting, first proposed by Sparck Jones [32]. In this scheme, a term t_i is assigned a weight

$$\log(\frac{N}{n_i}),$$

where N is the number of documents in the collection and n_i is the number of those documents containing term t_i. The inverse document frequency (generally abbreviated to *IDF*) is well-understood; Yu and Salton [33] showed that the use of IDF weighting would always improve retrieval effectiveness over the use of simple co-ordination level matching.

3.4.2 Other Term Weighting Strategies

Salton has long argued for the use of a weighting scheme in which a base weight for a word in a document collection (the IDF weight being a such a possible weight) is multiplied by a factor related to the number of times it occurs in each document [34]. This factor might simply be the *within-document* term frequency itself, or a linear transformation of it into some fixed interval, to compress the range of frequency values. Salton and Buckley [35] proposed the *normalisation* of term frequencies by the linear mapping

$$n(tf) = 0.5 + 0.5\frac{tf}{\max tf},$$

where $\max tf$ is the maximum term frequency of any term in a document. Salton also suggested the *length normalisation* of query-document scores, in order to make them invariant with respect to differing query and document lengths. This involves scaling each of the query and document vectors to unit length before calculating the similarity score. The length-normalised query-document similarity function is equal to the cosine of the angle between the two vectors in multi-dimensional space. The technique is of benefit where the contents of the document collection have widely varying lengths and the tendency of the unnormalised scoring method to favour longer documents is detrimental to retrieval effectiveness. Using the notation introduced earlier in this section,

$$sim(\mathbf{D}, \mathbf{Q}) = \frac{\sum_{i=1}^{I} b_D^i b_Q^i}{\sqrt{\sum_{i=1}^{I}(b_D^i)^2}\sqrt{\sum_{i=1}^{I}(b_Q^i)^2}}.$$

More recently, Robertson developed a theoretically sound basis for the incorporation of within-document term frequencies and document length normalisation into the query-document matching score [36, 37]. His method was shown to perform well in the TREC-2 evaluations.

3.4.3 Relevance Feedback

It would seem a natural development to exploit any information about the set of relevant documents that might come to light *during* retrieval, in order to improve the retrieval system's ability to locate unseen relevant documents. This is implemented by first performing a run of the retrieval system using the initial formulation of the query. When assessments of the first retrieved documents are obtained from the user, the query is reformulated to make it more representative of the documents marked as relevant, and less representative of the remainder. Assuming that the assessed documents are characteristic of the relevant and irrelevant documents in the collection as a whole, the system should now be more able to rank relevant documents highly on its next pass. This process is known as *relevance feedback* and a number of different strategies for its implementation have been tested.

Rocchio [38] proposed a method for relevance feedback in which mean vectors were obtained for each of the sets of retrieved relevant and retrieved non-relevant documents, and a new query obtained by taking a weighted sum of these vectors with the initial query vector. Formally, with initial query Q_0, and after the initial retrieval of n_1 relevant and n_2 nonrelevant documents,

$$Q_1 = Q_0 + \frac{1}{n_1} \sum_{relevant} \frac{D_i}{|D_i|} - \frac{1}{n_2} \sum_{nonrelevant} \frac{D_i}{|D_i|}.$$

This method therefore had the effect of *adding* new terms to the query and *reweighting* existing query terms. Salton and Buckley [39] tested a number of variants of this linear vector modification method and found that in general this technique was a computationally inexpensive way to perform relevance feedback. They noted that performance varied a great deal across differing document collections, and that there was more potential for improving retrieval where the initial performance was poor, than where output on the first pass was satisfactory. Another, very popular method of query expansion and reweighting is the probabilistic method [40], which will be described thoroughly in the next section.

In general, methods for relevance feedback must be scored by their ability to rank *unassessed* relevant documents more highly than was possible before the feedback step. Therefore, the initially-retrieved documents, which are seen and assessed by the user, must be removed from the document collection before retrieval using the reformulated query. This is the so-called "residual collection" method.

Queries gradually "honed" by the application of relevance feedback during retrieval can also be used in the automatic classification of new documents into the areas of interest they represent. This use of queries is called the *selective dissemination of information (SDI)* or *routing* and has a particular application in journalism, where documents, in this case news stories, are coming into news agencies constantly. Routing allows an incoming news wire can be diverted to a particular person without the need for a user to carry out an explicit search [41].

3.5 Probabilistic Retrieval

Historically, best match retrieval systems were initially concerned with simple query–document matching formulae, such as Cleverdon's co–ordination matching function [31]. However, retrieval was put on a firm probabilistic foundation by the work of Robertson and Sparck Jones [42]. They formulated the classic *probability ranking principle*, which states that a ranking of documents based on probabilities estimated using any available data, is necessarily optimal. Assuming the independence of term occurrences, and using binary document representations, they showed that an optimal ranking could be obtained in response to a query by assigning to each query term t_i a single weight

$$w_i = \log \frac{\psi_{i1}(1-\psi_{i2})}{\psi_{i2}(1-\psi_{i1})}$$

where

$$\psi_{i1} = \Pr(\text{document contains } t_i|\text{document is relevant})$$

and

$$\psi_{i2} = \Pr(\text{document contains } t_i|\text{document is nonrelevant}).$$

These probabilities can be calculated by splitting the document collection D into four partitions, as illustrated for a hypothetical collection and query in Figure 3.1, and defining the following values.

$$N = |D|$$

$$R = |\{d \in D|d \text{ relevant to the query}\}|$$

$$r_i = |\{d \in D|d \text{ is relevant and contains term } t_i\}|$$

$$n_i = |\{d \in D|d \text{ contains term } t_i\}|.$$

If each probability is estimated by taking the ratio of the appropriate pair of values, the substitution of the probability estimates into the weight formula gives

$$w_i = \log \frac{r_i(N-n_i-R+r_i)}{(R-r_i)(n_i-r_i)}.$$

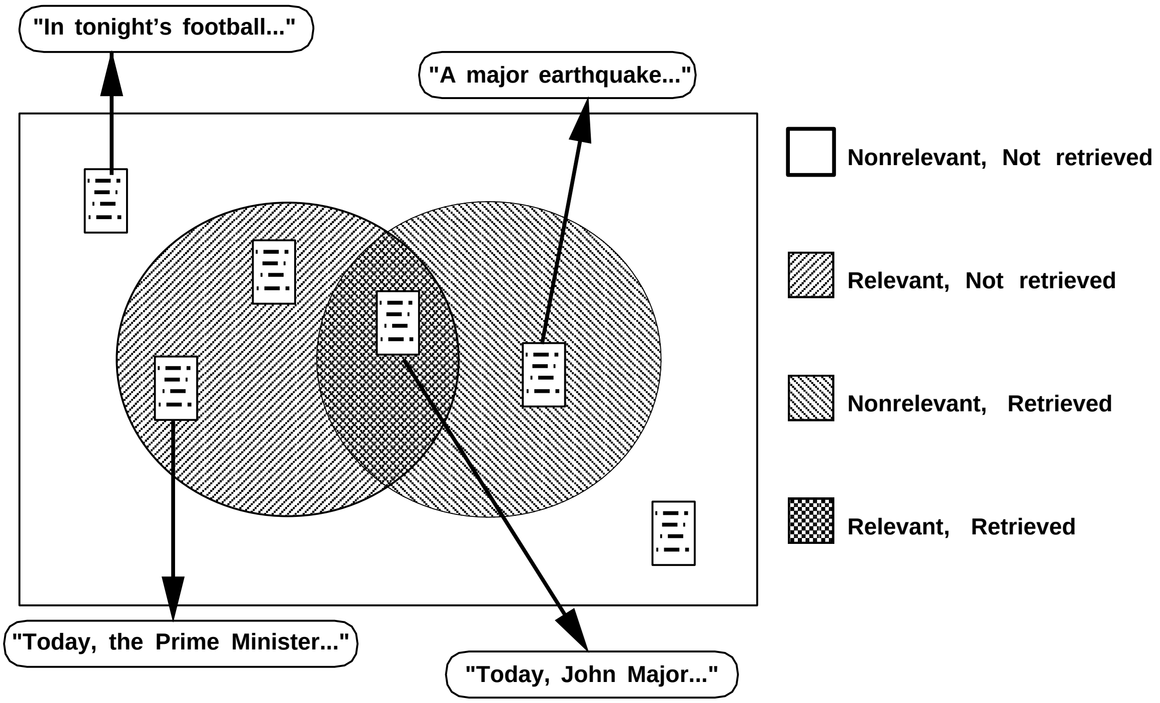


Figure 3.1: The partition of a document collection after retrieval on the best–match query "john major".

In this form, the *term relevance weight*, as it is called, can only be used *retrospectively* to deliver an optimal ranking on the document collection when a complete classification of the collection into relevant and non–relevant subsets is available. This is useful experimentally, as it allows the setting of an upper bound on how well retrieval can be performed in response to a given request [43]. In practice, it is common to use the term relevance weight as a method of probabilistic relevance feedback. The retrospective formulation of the weight can be made into a predictor simply by adding $\frac{1}{2}$ to each of the factors on the numerator and the denominator of the expression; this circumvents the problem that one of the partitions of the retrieved set might be empty. A threshold is set on the number of documents that are assessed for the estimation of term relevance weights; 10 and 20 are usual values of this threshold.

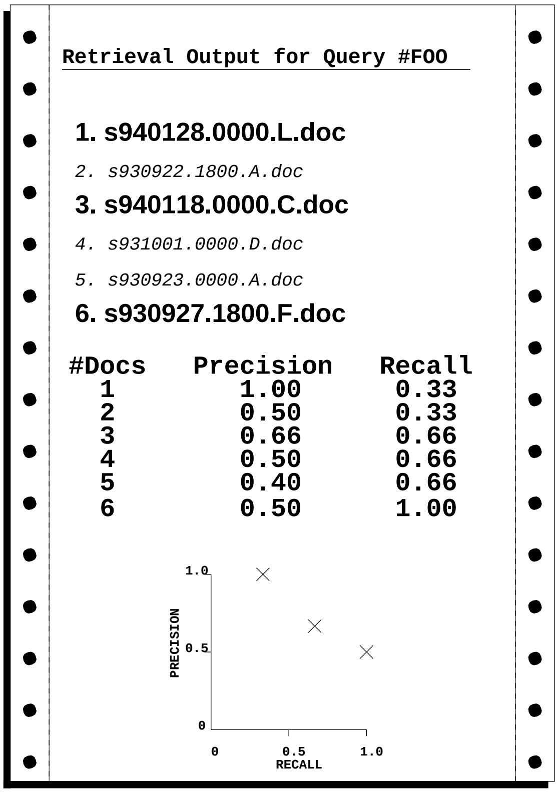
It is interesting to note that the term relevance weight and the inverse document frequency weight are mathematically related. Croft and Harper [44] showed that if it was assumed that no information was known about the relevant documents in the collection, then an attempt to calculate term relevance weights was reduced to a weighted sum of the query–document similarity score , using a "probabilistic" IDF weight, $log\frac{N-n}{n}$, and a simple co–ordination level matching score. Therefore, probabilistic methods can be used in retrieval in the first instance, and not solely after some initially–retrieved subset of the collection has been assessed for relevance.

3.6 System Evaluation

It is vital to be able to evaluate the performance of a number of retrieval systems to ascertain which one is the best. System evaluation depends on several "real–world" factors, such as expected financial cost, speed of indexing and retrieval, and so on. However, the major factor is *retrieval effectiveness*. This should reflect the ability to which a system is able to perform its job, that is, the retrieval of relevant documents and the suppression of non–relevant ones. The two most popular measures for retrieval effectiveness are *recall* and *precision*, which are defined as follows: given a set of retrieved documents, recall is the fraction of the relevant documents in the collection that have been retrieved, while precision is the fraction of retrieved documents that are relevant. A Boolean retrieval system retrieves only one set of documents, so the performance can be expressed as a single precision–recall pair, whereas a best–match retrieval strategy presents a ranked list of retrieved documents to the user. If this ranked list is thought of as a series of nested sets, obtained by moving a threshold down the list of retrieved documents, then a set of precision–recall pairs can be obtained and plotted. Figure 3.2 illustrates what the output of a simple experimental retrieval system might look like, after documents have been retrieved and assessed. For extreme simplicity, the names of relevant documents are shown in bold; it is also assumed that the entire set of relevant documents is known, and that this set contains only 3 documents. The output shows the ranked list of documents, and a table and graph which show how precision has varied against recall.

In all retrieval systems, precision and recall generally vary inversely with each other. With limited document representations[1], it does not seem possible to search the document collection for relevant documents without retrieving increasingly larger proportions of non–relevant documents. This inescapable fact will affect the way in which the user deals with the ranked retrieval output; a high–precision search will typically involve the user assessing the relevance of the top few retrieved documents and being satisfied with the one or two most relevant items in the collections. Alternatively, a high–recall search may involve the user assessing a larger number of the initially retrieved documents, then reformulating the initial query using relevance feedback, and searching "deeper" in the collection for relevant items. Precision and

[1] In best match searching, it is clearly of no consequence in retrieval whether the document representations contains many different terms, or just the query terms.



Retrieval Output for Query #FOO

1. s940128.0000.L.doc

2. s930922.1800.A.doc

3. s940118.0000.C.doc

4. s931001.0000.D.doc

5. s930923.0000.A.doc

6. s930927.1800.F.doc

#Docs	Precision	Recall
1	1.00	0.33
2	0.50	0.33
3	0.66	0.66
4	0.50	0.66
5	0.40	0.66
6	0.50	1.00

PRECISION
1.0
0.5
0
0
0.5
1.0
RECALL

Figure 3.2: The output of a hypothetical simple retrieval system.

recall are popular and useful measures, because they give a direct indication of the retrieval system parameters that are likely to be of interest to the user.

Precision and recall also have a close relation to automatic indexing; the tradeoff in which indexing discrimination and normalisation exist is closely related to the precision–recall tradeoff. As an example, the automatic indexing technique of word stemming is designed to improve indexing normalisation and thereby improve recall, but correspondingly it can have an adverse effect on precision. For example, document representations based on unstemmed indexing terms may allow precise query–document matching on the words which make up the phrase "teething troubles", which has a precise meaning, and may be used metaphorically. If a stemming algorithm is employed, the words may be reduced to, say, "teeth troubl"; occurrences of the precise phrase will still be matched correctly, but in this case the relaxation of the exact form of the words may only improve recall very little, if at all, whilst lowering precision by causing spurious matches on several phrases, such as "trouble with my teeth", which does not convey the relatively subtle intent of the original phrase.

The advantages of precision and recall as measures of retrieval effectiveness are that they are highly intuitive and easy to calculate. There are several disadvantages to their use, however. One of the most important problems is that for any set of retrieved documents, retrieval effectiveness must be expressed as a precision–recall *pair*. There is no universally accepted single measure for retrieval effectiveness, and it is unavoidable that a large amount of information will be lost when a set of precision–recall pairs is condensed into a single performance measure. Nevertheless, such measures can provide a useful characterisation of retrieval effectiveness, so long as their deficiencies are borne in mind [45]. An example single–unit measure is simply the precision obtained after the assessment of the top n documents in the ranked retrieved list. n may be set to values such as 10, 20 or 30 documents, to reflect the number of documents likely to be reviewed by different users interested in different precision–recall tradeoffs.

Salton has proposed a query–dependent measure, "R–precision" [46], which is the precision obtained when the number of documents retrieved equals the number of documents relevant to the query. For a hypothetical "perfect" retrieval system, this is where recall and precision would both equal 100%. Also, van Rijsbergen [4] has formulated a single–unit measure, E, to permit the direct comparison of Boolean and best–match retrieval strategies.

3.6.1 Average Precision

It would be foolish, from an experimental point of view, to attach too much importance to the retrieval effectiveness for a single query. This is because there is typically a large amount of intrinsic variation in request quality; some may be very good at retrieving the set of relevant documents[2], some very poor. As a result, it is common practice to perform retrieval for a number of queries and then to pool the results obtained on each query to obtain some average indicator of performance over the set of queries.

Van Rijsbergen [4] identified two different methods for cross–query averaging. In the *predictive* approach, precision values are pooled and averaged for fixed recall levels irrespective of the real precision-recall pairs produced by each query. Conversely, in the *descriptive* method, cross–query correspondence is based on some variable underlying parameter common to both queries, such as the number of documents retrieved. In the standard TREC scoring software, both methods are employed; the predictive approach to generate precision levels for fixed recall values between 0 and

[2] When requests are expressed as queries, of course.

1, the descriptive approach to produce "precision after the retrieval of n documents" scores.

To pool precision–recall curves over the query set to obtain a predictive average performance curve, a technique called *macroaveraging* is used. Macroaveraging is a method of interpolation between discrete points on a curve, not linearly, but using a step function, so as not to inflate results misleadingly, as illustrated in Figure 3.3[3]. For any recall value, the corresponding macroaveraged precision value is the average of the precisions for that point, obtained from each of the step functions.

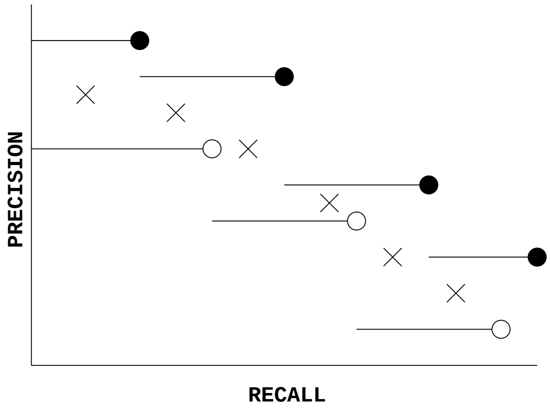


Figure 3.3: The use of macro–evaluation. The sets of circles correspond to precision–recall pairs observed for two separate queries. The adoption of the step functions means that an averaged precision for each recall value can be obtained simply by averaging the value of the step functions at that recall value.

Also in the TREC software, a single figure *average precision* is generated. Firstly, precision values are calculated, for each query, after each relevant document is retrieved. It is assumed that the precision for for any non–retrieved relevant document is zero. The precision values are then averaged together to produce a single averaged precision for the query, which corresponds conceptually to the area under the precision–recall curve for the query. Finally, the values are averaged for all queries to generate a single figure.

3.7 Conclusion

This chapter has described in some detail the core concepts of IR; indexing, querying, matching and assessment. Each concept will form a vital part of the the experimental

[3] This figure is taken from [4].

development of the *ad hoc* spoken message retrieval system which will be described in the later chapters of this thesis.

Chapter 4

Previous Work in Spoken Message Retrieval

There has been little contact so far between researchers in the areas of speech recognition and information retrieval. The work published so far in the area of spoken message retrieval seems to have had a firm foundation in one, but not both, of the parent disciplines. For example, a number of papers from the speech recognition literature have addressed the subject of *topic classification* or *topic spotting* [47, 48, 49]. Although topic classification, as will be seen, is roughly equivalent to routing in conventional text IR, there has been little research by the speech community on spoken message retrieval in response to *ad hoc* querying. Conversely, when attempts have been made by information scientists to build spoken message retrieval systems, these have involved a firm grasp of IR, with the emphasis on the employment of the conventional experimental techniques of obtaining queries and relevance assessments, and matching queries to automatically–derived "spoken document" representations [7, 50]. However, this work has so far remained in simulation only, and has not produced any results obtained from retrieval experiments on a *real* spoken message collection.

The two disciplines of speech recognition and IR have, unsurprisingly, recently progressed at different rates. Since fast, reliable speaker–independent speech recognition is still outside the reach of even the most powerful of today's personal computers and workstations, few people have daily contact with deployed speech recognition systems. However, the user base of restricted systems, such as wordspotters for automated telephone–based services [51, 52], and isolated–word recognisers as an alternative to keyboard input for personal computers [53] is now increasing. Whereas research into speech recognition has been slow to bear fruit, text–based document retrieval systems, as mentioned earlier, have been commercially available for many years.

Sharply differing philosophies characterise current research in speech recognition and information retrieval. In the field of automatic speech recognition, by far the most important research goal is a robust automatic dictation system that can recognise fluent speech uttered by any native speaker of the language for which the system is trained. Obtaining a measure of recogniser accuracy for experimental purposes therefore depends solely on a comparison between the string of words output by the recogniser and the correct string of words. The process of scoring is entirely objective. In contrast, the performance of a document retrieval system is dependent on several human factors — the requests supplied by each user, the relevance assessments made by that user, or by another human judge, the form of the initial query, and the degree and manner in which the query is subsequently refined.

The recent experiments in topic classification provide a concrete example of the contrasts described above. Topic classification, as it sounds, is the exhaustive partition of a set of spoken messages into a number of classes indicative of a broad message topic. System training involves the training of the acoustic models (which may or may not be HMMs) required for the speech recognition component of the classifier, and the use of a pre–classified message corpus to train weights relating the occurrence of words in each pre–classified message to the topic class of that message. The same corpus may be used for the training of both the acoustic models and the weights. Topic classification is performed on a set of unseen messages by recognising the content of each message, and combining the weights of the recognised words to produce a score representing the likelihood that each message belongs to each class.

The similarities between such a spoken message classifier and a document retrieval system will be discussed in some depth later; however, the primary differences are of particular interest here. Two major differences are the pre–training of weights relating words with topic classes, and the generation by the topic classifier of an exhaustive, "correct" partition of the message collection. This contrasts with an Information Retrieval approach to the problem, in which a "topic" is represented by a query, and the document collection partitioned only into two sets — the set of documents estimated to be relevant to the query (*i.e.* belonging to a single topic class of interest) and the remainder. Weights relating word occurrences in a document to the relevance or non-relevance of that document can only be estimated after the assessment by the user of the initially retrieved documents.

In general, the speech community's approach to information retrieval has concentrated on the preselection of queries and advance estimation of weights which correspond to the term relevance weights of IR. Topic classification can thus be seen to resemble the IR procedure of routing, in which a set of queries, honed by the use of relevance feedback, is used to classify incoming documents into each of a number of classes, each corresponding to and described by one of the existing queries. The *ad hoc* querying that is more usual in textual IR, is consequently not possible. This severely limits the potential utility of current experimental systems.

This chapter explores the existing literature, from the research communities of speech recognition, and to a lesser extent, information retrieval, covering topics related to the retrieval of spoken messages. First of all, the speech recognition technique of wordspotting is described, since it is the basis of several of the experimental systems described in the literature and a number of the original experiments presented in this thesis. Next, systems for topic classification of message corpora will be examined. Finally papers detailing the possible implementation of a system to retrieve spoken messages using conventional IR techniques will be described.

4.1 Wordspotting

Wordspotting is the particular application of automatic speech recognition in which the specific vocabulary of interest is relatively small, and it is the job of the recogniser to pick out occurrences only of these words (usually known as *keywords*) from the unknown speech. This is not to say that the total vocabulary of a wordspotter is necessarily as small as the keyword set; on the contrary, it is important to have some method for the acoustic modelling of non–keywords, in order to avoid a large number of incorrect keyword hypotheses (generally termed *false alarms*). The non–keyword model component of a wordspotter can range from a single acoustic "garbage" model intended to match all non–keyword speech, to a large number of models for a vocabulary of non–keywords, with a language model constraining the the recognised sequence of keywords and non–keywords.

Wordspotting has been a popular method to use in experimental telephone–based applications of speech recognition. Large–vocabulary speech recognition techniques are typically not used in these applications, since the acoustic quality of the speech is too poor to guarantee accurate word recognition, and the output of the recogniser need only be a single specific word or phrase, required to progress an automated transaction, for example in a telephone banking [54]. This has been reflected in the use of telephone–quality speech corpora, such as RoadRally and Switchboard, in wordspotting experiments [55, 56, 57]. The output of a wordspotter is typically a list of hypothesized keyword "hits", labelled with keyword start and end timings, and a score reflecting the relative certainty of the putative occurrence.

Assuming that keyword occurrences in a test collection are known exactly, the performance of the wordspotter can be measured. A typical measure of wordspotter performance can be obtained by first ranking the list of hypothesized keyword hits by score, selecting a threshold on the keyword scores, and calculating, for the set of putative keyword detections whose scores are above the threshold, the percentage of keyword occurrences correctly spotted and the rate at which incorrect hypotheses ("false alarms") were generated. Bearing in mind the obvious analogy with the ranking by decreasing score of documents in best–match document retrieval, an alternative approach to wordspotter scoring could be to define *keyword recall* to be the percentage of keyword occurrences correctly detected above a certain threshold, and *keyword precision* to be the percentage of keyword hypotheses that are correct.

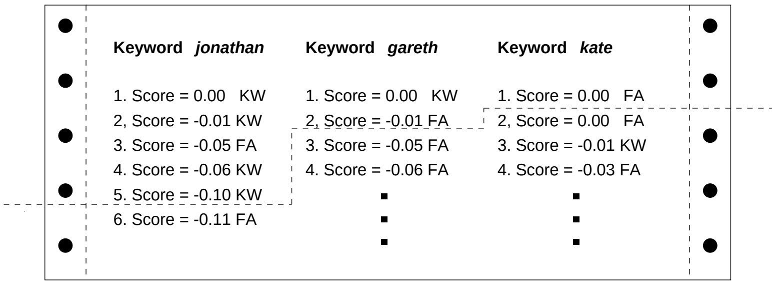
Each of these methods returns a pair of values characterising wordspotter performance. The standard measure of wordspotter performance, however, is a single figure measure called the *figure of merit (FOM)* [58]. For each keyword, a set of score thresholds is chosen, each corresponding to the point just before the generation of the n'th false alarm, as shown in Figure 4.1. The figure of merit is then calculated by an interpolative averaging of the detection rates, and a time normalisation, in accordance with the formula

$$FOM = (p_1 + p_2 + p_3 \ldots + p_N + ap_{N+1})/10T.$$

Here, p_i is the percentage of keyword hits, for all keywords, detected before the ith false alarm of each keyword, T is the size in hours of the wordspotter test collection, N is the first integer greater than $10T - \frac{1}{2}$, and $a = 10T - N$ is the interpolative factor.

The figure of merit averages keyword detection rates over the range of 0 to 10 false alarms per keyword per hour. It is experimentally useful in characterising the performance of differing wordspotters, but is limited by its dependence on retrospective keyword score thresholds. In practice, it is not possible to determine the exact threshold corresponding to some given number of false alarms for a keyword, since it is obviously not known whether a putative keyword detection is correct or a false alarm.

One of the best introductions to HMM–based wordspotting is Rose and Paul's seminal paper [59]. In it, they described one of the first wordspotters to be based on the Viterbi recognition paradigm. Viterbi recognition, since it is based on the calculation of the most likely sequence of models for the unknown speech, guarantees



Keyword *jonathan*	Keyword *gareth*	Keyword *kate*
1. Score = 0.00 KW	1. Score = 0.00 KW	1. Score = 0.00 FA
2, Score = -0.01 KW	2, Score = -0.01 FA	2, Score = 0.00 FA
3. Score = -0.05 FA	3. Score = -0.05 FA	3. Score = -0.01 KW
4. Score = -0.06 KW	4. Score = -0.06 FA	4. Score = -0.03 FA
5. Score = -0.10 KW		
6. Score = -0.11 FA		

Figure 4.1: The output of a hypothetical wordspotter. The horizontal dotted line represents the posterior keyword–dependent thresholds required for the Figure of Merit calculation.

that keyword hypotheses cannot overlap, and thereby limits the ability of the recogniser to generate false alarms. Figure 4.2 illustrates the recognition network for a wordspotter similar to Rose and Paul's baseline system.

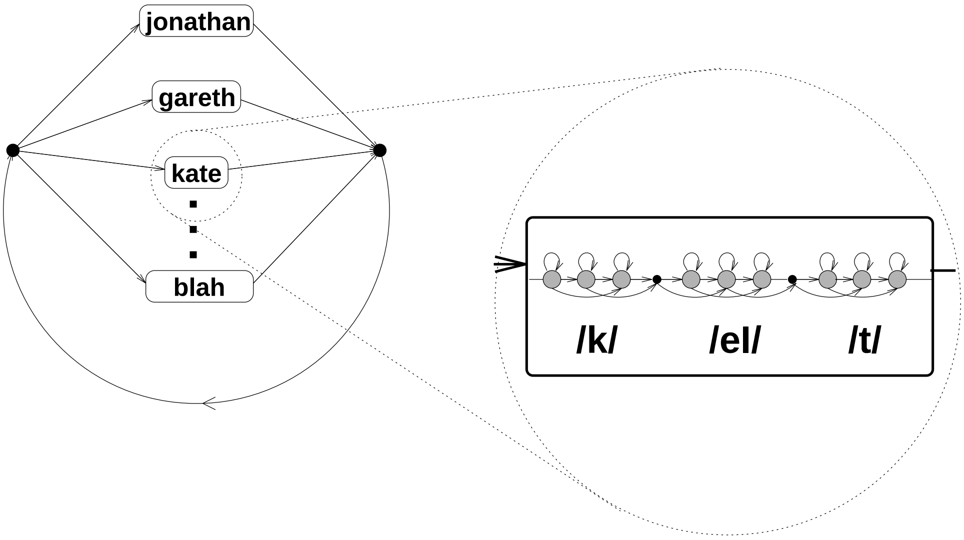


Figure 4.2: The recognition network traversed by a basic Viterbi wordspotter.

Rose and Paul identified and investigated several crucial areas of wordspotter design. They realised that while the use of keyword models trained at the whole word level guaranteed good wordspotter performance, it meant that the flexibility of keyword choice was severely compromised. Consequently, they experimented with creating keyword models by concatenating more flexible, sub–word HMMs. They also experimented with differing non–keyword models, ranging from a single, word–

independent acoustic "garbage" model designed to match all non–keyword speech, to a small vocabulary of whole words. They also proposed the keyword likelihood ratio scoring method. The main component of this method is the *log likelihood score*, the sum of state output and state transition log probabilities that is output, for each keyword hypothesis, by the Viterbi wordspotter. This score is normalised by subtracting from it the log score generated for the same section of the unknown utterance by a *keyword–independent* recogniser. The motivation for this technique is so that a correct keyword detection with a relatively poor score for acoustic reasons (such as, for example, cross–talk or background noise) is not penalised, since the same acoustic factors affect the normalising score.

The central issues of keyword training, choice and training of non–keyword models, and keyword hypothesis scoring, have since been explored extensively in the literature. A number of authors have recently come to the conclusion that the best way to perform wordspotting accurately is to implement a large vocabulary continuous speech recogniser, involving sophisticated approaches to sub–word modelling, and a well–estimated probabilistic language model to constrain the allowable word sequences. Weintraub [56] describes the use of the SRI *Decipher* large-vocabulary continuous speech recogniser to perform wordspotting on a vocabulary of 78 keywords on the ATIS air–travel enquiry task, and demonstrates a sharp increase in wordspotter figure of merit, from 48.8% to 75.9%, when the non–keyword model is improved from a word–independent set of 60 monophone models to a set of 1100 word models, many of which are trained at the whole word level or concatenated from sub–word *triphone* models, in which phone modelling accuracy is improved by the incorporation of acoustic context dependency. Rose [60], however, points out that Weintraub's conclusions were drawn on a task that was relatively constrained in terms of vocabulary size and grammar, and demonstrates that the use of large vocabulary systems or word–pair grammars on the "Credit Card" subset of the "Switchboard" corpus does not deliver greatly improved wordspotter performance when compared with the the use of a few hundred vocabulary words, without a grammar constraining the allowable word sequences.

Not all approaches to non–keyword modelling are based on the phone or the word as the intrinsic acoustic unit. Lleida *et al* [61], exploit the strong syllabic structure of Spanish in a wordspotting task in which the keywords are spoken digits. They modelled non–keyword utterances with a set of sixteen syllabic filler models created by mapping the set of Spanish phones into four broad acoustic–phonetic classes. They did not, however, present any objective results comparing the syllabic filler models with phone or word–based fillers on the same task.

The method used to score keyword hypotheses is closely related to the search algorithm used by the wordspotter. The log likelihood, and related measures like the log likelihood ratio score, is a simple by–product of the Viterbi recognition method, and makes this an efficient method for wordspotting. The principal alternative to the use of the Viterbi algorithm in wordspotting is the forward–backward algorithm, which is also the basis of the Baum–Welch training algorithm described in chapter 2. Wilcox and Bush describe a single–keyword wordspotter based on the forward–backward algorithm and claim that it performs more accurately and more quickly than a comparable Viterbi wordspotter [62, 63]. Using the notation introduced in chapter 2, the probability of occupation of the end state e of the keyword model, given the speech data observed so far, is computed for each time t using the formula

$$\mathrm{Pr}(x_t = e|\mathbf{o_1} \ldots \mathbf{o_t}) = \frac{\alpha_e(t)}{\sum_j \alpha_j(t)}.$$

Peaks in this probability are detected and a backward search initiated at each peak. The backward probability is normalised by the keyword duration and the non–keyword score for the same section of speech to produce a likelihood ratio score for each keyword hypothesized in the forward pass. The computational advantage over the Viterbi method derives from the fact that backtrace through the model sequence is only required for a hypothesized keyword, instead of for the entirety of the unknown speech.

A similar technique is used by Jeanrenaud *et al* [64], except that the keyword score output by the wordspotter is the *posterior* probability of occupation of the end state e_w of word w at time t, that is, the probability of state occupation given the entire sequence of unknown speech. Formally,

$$\mathrm{Pr}(x_t = e_w) = \frac{\alpha_{e_w}(t)\beta_{e_w}(t)}{\sum_{\text{all } s} \alpha_s(t)\beta_s(t)}.$$

In practice, owing to the uncertainty of word boundaries in continuous speech, this probability is integrated over a time window whose size depends on the approximate length of the putative word detection.

The experiments described in Rose's more recent publications have sought to decrease the dependence of keyword and non–keyword model training on the availability of many spoken examples of these specific words. Rose and Hofstetter attempted to reduce the task–dependency of model training in their telephone–speech wordspotting experiments by training populations of triphone, biphone and monophone models from the phonetically balanced TIMIT speech recognition corpus. They then interpolated poorly–trained triphones with biphones and monophones to improve model robustness, and finally interpolated the TIMIT triphones with the corresponding models trained from the telephone–speech corpus [65]. Rose has also used the more sophisticated method of *decision–tree model clustering* to generate populations of well–trained sub–word HMM models for wordspotting [60].

The state of the art in flexible wordspotters based on the Hidden Markov Model can arguably be said to be a speaker–independent speech recogniser in which acoustic models for keywords, and a vocabulary of non–keywords, are concatenated from well–trained, acoustic context–sensitive phone models and word sequences are constrained by a task grammar, if appropriate. However, a factor of particular importance to the recognition task involved in spoken message retrieval, is the *speed* at which speech is decoded by the recogniser into a string of keyword and non–keyword hypotheses.

The issue of wordspotting speed is seldom mentioned in the literature. It is hinted at in the work of Wilcox, since the specific application of interest for the forward–backward wordspotter is the fast indexing of spoken messages such as voice mail. Rose also mentions that his wordspotter runs constantly, with partial traceback through the Viterbi path, so that putative word hits can be identified before the end of the unknown speech [59]. The delay between the utterance of the keyword and its hypothesis by the wordspotter is given as being of the order of a few seconds, which indicates that the wordspotter, once initialised by the loading of the models and network, runs in real time.

Real time operation can be considered "fast" in a telephone–based wordspotting application, since a delay of three seconds before some action is performed in response to a caller's request would probably be acceptable to the caller. However, it should be clear that in order to index and retrieve the contents of a spoken message collection in response to an *ad hoc* query submitted by a user, a conventional wordspotter will be unacceptably slow in detecting occurrences of the specified query terms. Any approach to *ad hoc* message retrieval using word–like query terms must consequently depend on one or both of the following; a large–vocabulary continuous speech recogniser to detect occurrences of a large proportion of the potential query terms, and a very fast wordspotter (where "very fast" here means "many times faster than real time") to spot relatively infrequent but important terms.

4.2 Topic Classification Systems

As was mentioned in the introduction to this chapter, the speech recognition community has realised the potential utility and computational tractability of spoken message retrieval, but only so far to a limited extent. In topic classification, the experimental philosophy of speech recognition, namely the training of acoustic model parameters and then the testing of the models on unseen acoustic data, has also been applied to the retrieval aspect of the problem. This has resulted in the adoption of a "top–down" approach, as it were; typical experimental practice is to define a set of allowable message "topics" and collect messages for the experimental collection with specific regard for these topics. Moreover, the corpus of messages is partitioned into training, evaluation and test subsets; the training subset is used not only to prepare acoustic models, but to select, for each topic, some vocabulary of words (which can be thought of as the terms of a query intended to retrieve messages on the specified topic), and train weights relating each of these words to the topic.

In addition, word–topic weight estimation is not necessarily based on the output of the trained speech recogniser, but may use a manually–obtained text transcription of the training speech. For each message, topic classification is then performed by obtaining a score for each message, based on the output of a speech recogniser (which may or may not be a wordspotter) and weights relating each keyword with each topic. The objective is then to obtain an exhaustive partition of the message collection as accurately as possible. Performance is usually measured in terms of the percentage of messages correctly classified.

Figure 4.3 shows a schematic diagram of a very simple topic classifier. Unknown speech messages input into the system are recognised and the output recogniser transcriptions passed to the classifier, which classifies each message into one of two categories based on the occurrence of each of six words in the recognition output. The network depicts a set of word nodes which are related to each of the topic *nodes* by the set of pre–trained word–topic weights. The topic scores for a message are obtained by summing the appropriate word–topic weights, and the message is classified as belonging to the highest–scoring topic.

Whilst there are many philosophical and implementational differences between the topic classifiers reported in the literature and a putative *ad hoc* query–based retrieval system, it is interesting that the design of such classifiers exhibits several principles in common with conventional information retrieval. This section examines several recently proposed topic classifiers with particular regard to the theory of document retrieval that was outlined in chapter 2.

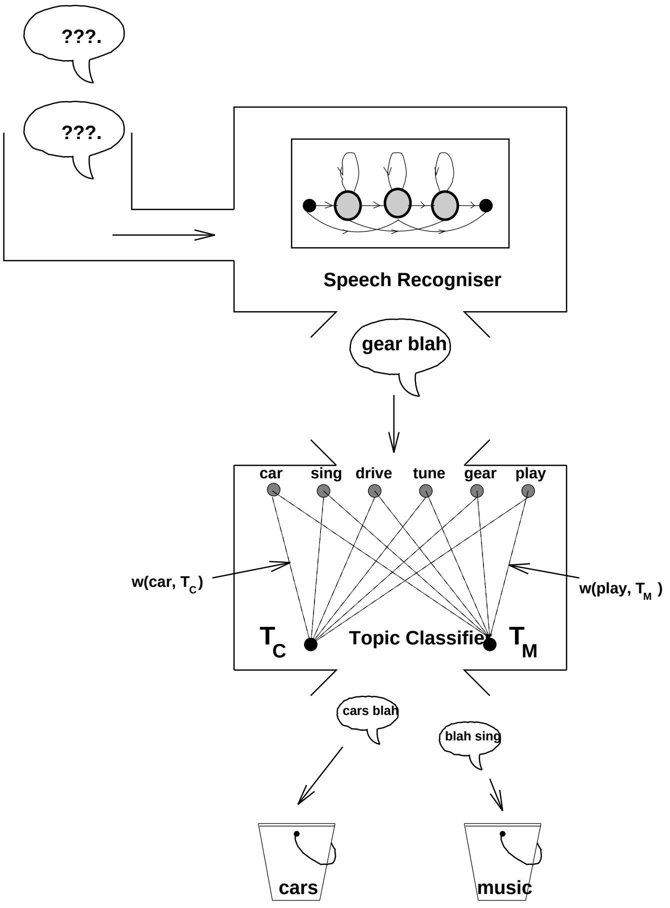


Figure 4.3: A 2–class topic classifier with a 6–word message classification vocabulary.

Rose, Chang and Lippmann presented a topic classifier which they claimed was the "first end–to–end speech message information retrieval system" [48]. Pre–defining the 6 message *scenarios* illustrated in Table 4.1, they collected a corpus of 510 spoken messages. For each topic class $\mathrm{C_i}$, the mutual information

$$I(\mathbf{C}_\mathrm{i}, w_k) = \log \frac{\Pr(\mathbf{C}_\mathrm{i}, w_k)}{\Pr(\mathbf{C}_\mathrm{i})\Pr(w_k)}$$

was calculated between $\mathrm{C_i}$ and each word w_k occurring in the corpus. The top 40 words with the highest mutual information with the topic class were then selected as the "query" to be used to retrieve documents on topic $\mathrm{C_i}$. The message classifier was a network whose input was a vector describing the presence or absence of each term in the input message M and in which the mutual information was used to weight the connection between each word w_k and each topic output.

Toy Description	Abstract Object Description
General Discussion	Map Reading
Photographic Interpretation	Cartoon Description

Table 4.1: Message classes in classification experiments of Rose *et al.*

Now, an estimate of $I(\mathbf{C}_\mathrm{i}, w_k)$ can be calculated by a four–way partition of the set of test messages, depending on (a) whether or not a message belongs to topic class $\mathbf{C}_\mathrm{i}$ and (b) whether or not it contains word w_k. If N is the number of messages in the test collection, R_i is the number belonging to topic class $\mathbf{C}_\mathrm{i}$, n_k is the number of messages containing word w_k and r_{ik} is the number of messages in class $\mathbf{C}_\mathrm{i}$ containing word w_k, then, estimating the probabilities by frequency counts,

$$I(\mathbf{C}_\mathrm{i}, w_k) = \log \frac{\left(\frac{r_{ik}}{R_i}\right)}{\left(\frac{n_k}{N}\right)}.$$

This is actually identical to a form of retrospective term relevance weight, initially proposed in the IR literature by both Barkla [66] and Miller [67], and reviewed by Robertson and Sparck Jones in their classic paper on the subject [42]. Moreover, Rose proposed, but did not test, a different approach employing a weighting scheme intended to maximise the log likelihood of the class of interesting messages relative to the class of uninteresting messages. Using the same partition of the message collection and again estimating the probabilities using frequency counts leads to a weight v_{k1} given by the formula

$$\begin{aligned} v_{k1} &= \log \frac{\Pr(w_k|\mathbf{C_1})}{\Pr(w_k|\mathbf{C_2})} \\ &= \log \frac{\Pr(w_k, C_1)\Pr(C_2)}{\Pr(w_k, C_2)\Pr(C_1)} \\ &= \log \frac{\left(\frac{r_{ik}}{R_i}\right)}{\left(\frac{n_k - r_{ik}}{N - R_i}\right)}, \end{aligned}$$

which corresponds, using the terminology of the Sparck Jones paper, to a proportion–based (as opposed to odds–based) version of the standard IR probabilistic term relevance weight.

The speech recognition component of the message classifier is a speaker independent wordspotter, with whole–word HMM keyword models trained from a specially obtained corpus of keyword utterances. The classification accuracy on text transcripts of the 120–message test set is 78.2% on a wordspotter vocabulary reduced from 240 words to 126 words using a genetic algorithm method. However , the accuracy drops dramatically, to 50.0%, when classification is performed using the wordspotter output. This result is important as it shows the effect of using incomplete and errorful recogniser output in classification, in comparison with making the assumption that wordspotter performance is "perfect".

Rose proposes an interesting method of overcoming the shortcomings of the wordspotter. Bearing in mind that the performance of the wordspotter varies widely for differing keywords (for example, false alarms of short keywords are typically far more frequent than those for longer ones), he proposes a method whereby the keyword log likelihood scores output by the wordspotter are transformed by a set of keyword–dependent sigmoid functions. The output of the transformation is a set of weights reflecting the potential utility of the keywords in message classification. The effect of the transformation should be to output low "utility scores" for low–scoring detections of unreliable keywords, whilst assigning high utility scores to more reliable keywords with similar scores.

The parameters of the sigmoid are adjusted by back–propagating a measure related to message classification error. It does not require specific knowledge of which keyword detections are correct and which are false alarms. The sigmoid output, being in the range 0 to 1, is then used as input to the message classifier instead of the binary keyword occurrence value. The method improves classification performance to 62.4%, but the lack of sufficient data to have separate evaluation and test sets means that the sigmoid parameters have to be estimated from the 120–message test set.

McDonough *et al* [47] have recently performed some topic classification experiments using the same underlying methodology, but with some sharply differing implementational details. They test the performance of two separate strategies for representing the content of speech messages. In a continuous speech recognition approach, the unknown speech is decoded into an unscored sequence of word hypotheses, and the score input into the classifier for each keyword is simply the number of times it is hypothesized by the speech recogniser for each message. However, they experiment with a more sophisticated method of obtaining a keyword occurrence count for each message, by using the same recogniser in "wordspotting" mode. It calculates the forward–backward posterior probability of occupation of the final state of each keyword for each time t, given the observation sequence, the set of models and the recognition network. Putative occurrences of each keyword are obtained every time this probability reaches a local maximum. The total number of occurrences of a keyword in a message is obtained by summing the probabilities of each of the putative keyword hits.

As in Rose's system, the message collection (in this case, the Switchboard corpus of spontaneous telephone speech) has been collected by recording conversations prompted by a number of scenarios, which are then used as the topics into which the messages will be classified. In this case, however, there are enough data to allow the collection to be partitioned into training, development and evaluation sets. The training set is used to prepare the sub–word HMMs used by the recogniser, the development set to select a vocabulary of keywords for each topic and train word–topic weights, and the evaluation set to test the topic classifier.

McDonough *et al* realise that keywords that may be good in discriminating between topics for textual transcriptions of the messages may not be at all good at doing the same for the output of a speech recogniser. For example, one of the Switchboard conversation scenarios is "buying a car". The list of Switchboard message topics used in their experiments, illustrated in Table 4.2, suggests that the occurrence in the message texts of the words "buy" and "car", or word variants thereof, would discriminate those messages generated by this scenario. However, since these words are short, a wordspotting front end would have a strong tendency to generate many false alarms of these keywords. One solution to this problem is Rose's back–propagated keyword score correction function; another, as adopted by McDonough, is to match the generation of topic vocabularies to the speech recogniser used to generate message transcriptions. This means that frequently hypothesized words are omitted from the vocabularies describing each topic, since in recognition output they do not appear to discriminate any topic from any other.

Air Pollution	Music
Crime	Gun Control
Buying a Car	Public Service
Pets	Public Education
Exercise & Fitness	Family Life

Table 4.2: Switchboard message classes in classification experiments of McDonough *et al*.

Two methods are used to select the topic–specific word vocabularies. Both are based on the chi–squared hypothesis test. For each topic, word frequencies in topic messages are accumulated and the χ^2 value calculated for each word for each topic a) against all the other topics simultaneously and b) across the union of all other topics. Moving a threshold on the significance values of each word allows vocabularies of varying sizes to be obtained. These two different approaches to topic modelling approximate closely to Rose's methods for 2–class and n–class topic classification. This method for selecting the best words for discriminating between topics serves to filter out function words[1] in the case of textual input to the word selector, and words which have a strong tendency to generate false alarms, in the case of speech recognition input.

Once the topic vocabularies have been selected, the keyword–topic weights are calculated. Here, the log conditional probability of word w_k given topic T_i, $\log \Pr(w_k|T_i)$, is used. This weight can be rewritten as $I(T_i, w_k) + \log \Pr(w_k)$, and therefore corresponds to the Rose form of term relevance weight plus a log factor which penalises generally infrequent terms. The likelihood of a message M belonging to topic class T_i is then given by the equation

$$L_i(M) = \sum_{k=0}^{K} \frac{n_k}{N} \log \Pr(w_k|T_i),$$

where K is the number of keywords in the vocabulary for topic T_i, n_k is the number of times word w_k is hypothesized for message M, whether this number is a straightforward count from the word recogniser or a sum of probabilities from the wordspotter, and $N = \sum_k n_k$ is the total number of words in message M. Note that it is used in the formula to normalise the message–topic likelihood by message length.

The acoustic component of the topic classifier is a medium–vocabulary continuous speech recogniser with a vocabulary of 4400 words, word models built from context–dependent phone HMMs, and a bigram language model. The bigram is trained by mapping all occurrences of non–recognition–vocabulary words occurring in the training set to a single "out–of–vocabulary" (OOV) label, and an acoustic OOV model is constructed simply by constructing a loop of context–independent phone models arranged in parallel.

Classification results obtained on the development–test set show that the "topic–against–all–other" method of obtaining topic vocabularies outperforms the "topic–against–each–other" method for cross–topic vocabularies of up to 1000 words. The more successful method corresponds more closely to relevance feedback in conventional retrieval, in which term weights are chosen to maximise discrimination of the retrieved set of relevant documents from the set of non–relevant retrieved documents. The results also demonstrate the importance of matching the generation of message transcriptions to the method for selecting the topic vocabularies and training the word–topic weights. Where these are mismatched, classification performance

[1] such as *and*, *or* and *not*, for example.

can drop significantly, in one case by as much as 20%. This mismatch problem would not occur in a spoken message retrieval system, as the only form of the message from which terms could be selected, and weights trained, would be the output of the speech recogniser. In this case, the problem of function words and high–false–alarming keywords could be dealt with by the use of the inverse document frequency weight, or probabilistic term relevance weight, which would automatically assign a low weight to those words occurring frequently across all documents.

4.3 Spoken Message Retrieval Systems

In the last few years, research in the area of "pure" spoken message retrieval has begun to appear. In contrast to the topic classifiers which have been reported on in the speech literature, they employ the methodology of conventional information retrieval. That is, document representations are constructed using terms from an indexing language, queries and relevance assessments are obtained from real users, and spoken messages are presented to the user by a query–document score ranking, instead of an exhaustive partition of the message set. In contrast with the top–down, corpus–driven methods of speech topic classifiers, spoken message retrieval can be viewed as a "bottom–up", query–driven task. In this section, a number of crucial papers in the relatively new field of spoken message retrieval will be discussed in some detail.

4.3.1 Sub–word methods

Glavitsch and Schaüble [7] were the first to consider the potential application of classical text Information Retrieval methods to speech messages. In their paper, they point out that speech document indexing cannot be query–dependent, at least not heavily so, using existing speech recognition methods. They argue that the use of existing recognition techniques, such as wordspotting, to detect occurrences of arbitrary query terms in a large speech message database would result in an unacceptable delay before the messages of interest are retrieved. They consequently reject the use of words or word–stems as indexing units. They also reject the use of conventional acoustic sub–word units, like the phone, in indexing, since they are too unreliably recognised, and not sufficiently discriminating by themselves.

Glavitsch and Schaüble propose a single syllable–like acoustic unit for both recognition and indexing. The units are obtained by slicing the phonetic decomposition of any word into units of the form CV, VCV or VC, where V and C denote maximal strings of vowels and consonants respectively. These sub–word units are referred to as "VCV–features". Table 4.3 illustrates the feature decomposition of a number of words. Acoustically, the constituent VCV–features of a word do not overlap, since each vowel is bisected and each half assigned to a different feature. This can be seen from Figure 4.4.

Word	Example Phonetic Decomposition	Feature Decomposition
president	p r e z I d @ n t	pre ezI Id@ @nt
advice	@ d v aI s	@dvaI aIs
demonstrated	d e m @ n s t r eI t I d	de em@ @nstreI eItI Id

Table 4.3: Examples of VCV–feature decomposition.

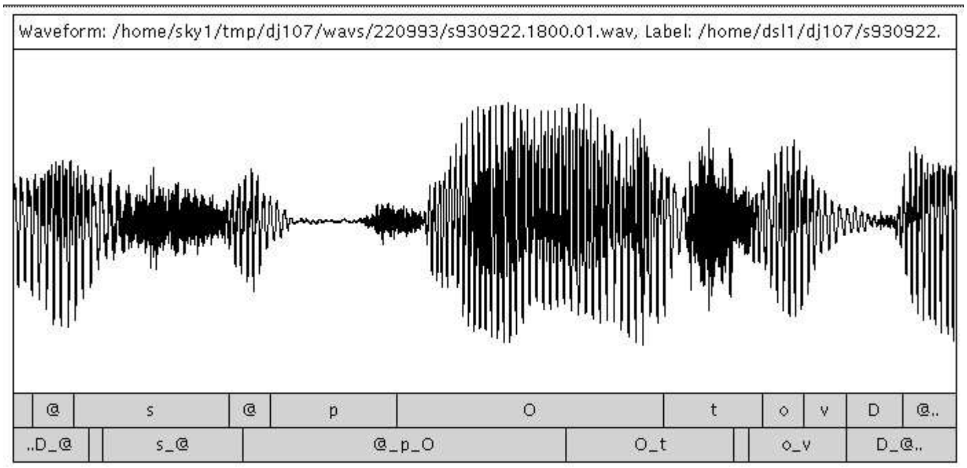

Waveform: /home/sky1/tmp/dj107/wavs/220993/s930922.1800.01.wav, Label: /home/dsl1/dj107/s930922.
@
s
@
p
O
t
o
v
D
@..
..D_@
s_@
@_p_O
O_t
o_v
D_@..

Figure 4.4: A speech waveform with both phone and VCV–feature labels. It can be seen how VCV–feature boundaries bisect vowel occurrences.

For retrieval, a vocabulary of VCV–features is estimated from text transcriptions of the acoustic training data, selecting those features which occur frequently enough that "whole–feature" HMMs can be trained for them, but not so frequently that their ability to discriminate between messages is poor. Glavitsch and Schaüble claim that once a set of VCV–features is estimated for a particular message domain, such as items of radio news, it is useful in indexing any set of messages from that domain.

The acoustic component of their proposed retrieval system is a wordspotter based on Wilcox's [62]. Since Wilcox's forward–backward wordspotter can only detect occurrences of a single keyword, it would have to be run over a speech message collection once for each single VCV–feature. Whilst all the message indexing is query–

independent, and the system should retrieve documents in response to a user's query with little delay, this method of message indexing would in practice be highly time–consuming, and clearly a more efficient method (such as Viterbi recognition) would be required.

Once the message representations are obtained, query–message matching can be performed by decomposing a query into its component VCV–features, extracting those which are members of the indexing vocabulary for the task and calculating the similarity between the query and each message, weighting each feature with inverse document frequency and term frequency weights, and performing message length normalisation. IDF weights are used because, in information retrieval, unlike topic classification, no measures associating specific words with a particular topic of interest can be obtained in advance of retrieval.

So far, however, no results produced by a real speech retrieval system, of the kind outlined above, have been published. Some further work has been carried out, on the same indexing and retrieval paradigm, working with *simulated* wordspotter output, to demonstrate the potential utility of a sub–word feature–based system [50]. These experiments are based on extracting the feature representation of a document and simulating the incomplete and errorful output of a wordspotter at particular detection and false alarm rates. This is simply done by removing a proportion of the correct features, and inserting random features, under an assumption about speaking rate. Experiments carried out on conventional text–based document collections demonstrate, Schäuble and Glavitsch claim, the feasibility of the technique, even with the simulated wordspotter performance as poor as 40% correct feature detection and 50 false alarms per feature per hour. Glavitsch and Schaüble conclude that conventional retrieval methods seemed quite good at distinguishing the separate distributions of correct features and false alarms.

4.3.2 Whole–Word Methods

The "Video Mail Retrieval by Voice" (VMR) project, mounted by the University of Cambridge and Olivetti Research Ltd., was established in response to the problem of managing and retrieving the contents of a archive of stored video messages sent between users of *Medusa*, a high–capacity desktop multimedia environment installed at the offices of Olivetti Research Ltd. in Cambridge, UK. [68]. Jones *et al* describe in their recent paper, the setting up of a database of video mail message soundtracks, and initial experiments carried out in message retrieval [69].

The VMR message corpus was obtained by initially selecting 10 message categories to reflect the kind of correspondence being carried by Medusa. 50 prompts were derived from these categories, and 15 speakers prompted to produce a total of 300 messages in response to them. Examination of the prompts and some of the message transcriptions shows that the VMR corpus does not exhibit the same high degree of topic "separation" as the Switchboard corpus.

Retrieval experiments have so far concentrated on a fixed vocabulary of 35 keywords chosen from the vocabulary of the message set, with message representations generated using a wordspotter built on a Viterbi HMM recogniser. Keywords are modelled by speaker–dependent whole–word HMMs, trained from isolated and within–sentence spoken keyword exemplars, and non–keywords by speaker dependent sub–word HMMs trained from non–keyword speech. Keyword hypotheses are scored by obtaining a durationally–normalised comparative log likelihood score, except that unlike Rose's approach, the keyword and filler model likelihoods are divided rather than subtracted.

Jones *et al* realise that the *ad hoc* nature of query formulation and relevance assessment in conventional information retrieval does not allow the learning in advance of characteristics related to the correct detection, or incorrect hypothesis, of individual keywords. Their current approach to the incorporation of keyword acoustic likelihoods into the message representation is to set a word–independent threshold on the likelihood ratio keyword scores. Decreasing this threshold has the effect of incorporating more correct keyword detections into the message representations, at the expense of adding more false alarms.

Initial retrieval experiments have been performed on an artificial set of 50 queries and relevance assessments. The queries were derived from the set of message prompts by extracting keyword occurrences from each prompt and suffix–stripping. The set of relevant documents for each query was defined as the set of 6 messages spoken in response to the original prompt. While in this initial experiment, the set of queries and assessments correspond to an exhaustive partition of the message set, as in topic classification experiments, the estimation of this partition is *not* the aim of these experiments, since the queries are submitted to the retrieval system sequentially. Keyword occurrences were weighted using the standard inverse document frequency weight. Retrieval was then performed a number of times on different message representations by varying the value of the keyword inclusion threshold. As in Glavitsch and Schaüble's simulation experiments, average precision figures were calculated, and compared with the performance of a *reference run* in which the message representations were composed of the known keyword occurrences.

The spoken message retrieval effectiveness, measured in terms of the ratio between the average precision of the spoken message retrieval and the average precision of the reference run, varies with the choice of wordspotter output threshold. It peaks at a particular threshold value which seems to represent the best trade-off of keyword detection rate against false alarm rate. If the threshold is set too high, there is an insufficiency of keyword occurrences to discriminate relevant from non–relevant documents; if too low, the utility of the correct keyword detections is swamped by an overabundance of false alarms. For the best keyword detection threshold, the spoken message retrieval performance is 87.6% of that obtained for text.

Jones and his co–workers plan to extend their work into a complete message retrieval system, employing speaker–independent and keyword–independent HMM models, incorporating automatic methods of query expansion, and the use of a sophisticated front–end for message browsing and querying.

4.3.3 Spoken Indexing into Textual Document Collections

Kupiec, Kimber and Balasubramanian [29] have attempted to integrate speech recognition and information retrieval in a manner diametrically opposite to that of Glavitsch and Schaüble, or Jones. Instead of a textual (but potentially spoken) interface to a large database of spoken messages, they have proposed a spoken interface to a large textual collection, the widely–available *Grolier Academic American Encyclopaedia*. This shifts the speech recognition problem from the content of the collection to the content of the query. However, their retrieval system, also reversed with respect to those already discussed in this chapter, exhibits some similar behaviour.

The recognition component of Kupiec's retrieval system is a Viterbi recogniser which generates the best sequence of HMM phone models for any input utterance (in this case, query terms, spoken in isolation). The fundamental *indexing* unit, however, is the word. Since phone sequences output by the HMM recogniser are by no means guaranteed to correspond exactly to some word of English, a method is required to map from an arbitrary phone sequence to a single word, as described in an English pronunciation dictionary. This mapping is performed by constructing a discrete HMM to represent the allowable variation between the "correct" pronunciation of a word, and the output of the phone recogniser. This allowable variation is learnt by using the speech recogniser to recognise words from its own training data. The statistics are learnt independently for each phone, so pronunciation HMMs are simply concatenated from the corresponding phone statistics when required.

Every time an isolated query word is recognised as a sequence of phones, the contents of a large phonetic dictionary of English are matched against the recogniser output, and the 30 best guesses at the identity of the word are obtained by ranking each dictionary sequence by the estimated probability that the recogniser output could be produced by the discrete pronunciation HMM associated with each word. A Boolean query is then presented to the retrieval system for matching against the contents of the document collection. This query is constructed by ORing the 30 hypotheses for each query term together and then ANDing the ORed groups. The idea behind this technique is as follows. For a query composed, say, of two words uttered in isolation, the query generation system described above will generate 900 possible word combinations. Only a very small proportion of these word pairs will be detected in a document within a permitted level of proximity of each other. Typically, only those pairs of words which are *semantically* related in some way will occur together in the document collection.

Kupiec offers, by way of illustration, a putative query of the form {precedent OR prescient OR president OR resident} AND {kennerty OR kennedy OR kemeny OR remedy}. Of the 16 possible phrases generated by this query, one particular phrase, "president kennedy", is far more likely to appear in a large collection of documents than any of the others. The range of the query is thus restricted by the semantic content of the document collection being searched. Kupiec labels this phenomenon *semantic co–occurrence filtering*, and proposes it as a general technique for post–processing multiple word sequences output by a general continuous speech recogniser.

Experiments are carried out by constructing 100 *ad hoc* queries, each containing a few words, and defining query "success" as the appearance in the top 25 retrieved documents of at least one relevant document. From an IR perspective, this is a rather loose definition of success, since it corresponds to the retrieval system operating at a precision–recall tradeoff of only 4% precision and some relatively small level of recall (which cannot be calculated unless the precise number of relevant documents in the collection is known). 83 of the queries are "successful" with respect to this definition; the success of 51 of these is owed to the use of the 30 best hypotheses for each word in the queries, rather than the single best. Kupiec *et al* announce their intention to improve the performance and extend the scope of this initial system by employing improved, speaker–independent phone models.

4.4 Conclusion

The review of related work contained in this chapter has concentrated on the major differences between the methodologies of classical Information Retrieval and of the topic classification experiments which have predominated in the speech recognition literature. It has also, however, pointed out such similarities as exist, and described the unified approaches which are starting to emerge.

Chapter 5

Baseline Retrieval System

It was shown in chapter 4 that work has begun on the application of automatic speech recognition techniques to the spoken information retrieval problem. However, each of the classical retrieval systems (as opposed to topic classifiers) described in the literature is limited to some extent. Glavitsch and Schäuble's results are confined to retrieval performed on a *simulated* sub–word recogniser, operating at varying feature detection and false alarm rates; Kupiec's experiments are based on spoken query representations indexing a large textual document collection, using a very weak definition of retrieval success; and Jones and his colleagues' work has so far used artificial queries and relevance assessments and a controlled vocabulary of query terms. It seems appropriate to consider how the state of the art in spoken message[1] retrieval can be advanced by relaxing these constraints.

Firstly, it is important that the experimental collection must be composed of *real* spoken messages, instead of text from which representations are obtained for retrieval purposes by simulating speech recognition. Also, the spoken message collection must be big enough to be partitioned into non–overlapping sets, one for acoustic model training and one consisting of unseen messages on which retrieval experiments may be performed.

No existing corpus regularly used in speech recognition is suitable for retrieval experiments; the corpora generally used in the speech processing community during the late 1980's, such as TIMIT and the RM database, were either designed to provide experimenters with a balanced collection of different sounds in differing acoustic contexts, or composed of utterances of sentences generated automatically from a very strict vocabulary and grammar of English [70, 71]. More recently, these databases

[1] In this and subsequent chapters, a *message* is taken to be an item in the specific collection on which experiments are performed here; the word *document* will be used more generally, to refer to an item in an arbitrary IR test collection.

have been supplanted as the primary benchmark of recogniser performance by corpora of news stories from the *Wall Street Journal* newspaper [13].

While recordings of *entire* articles from the newspaper would constitute a valuable resource for experimentation in spoken message retrieval, the *WSJ* speech data for acoustic model training and testing are released to the speech recognition community in the form of individual sentences. Although a number of sentences from one news article uttered by an individual speaker may occasionally appear in sequence in the collection, it is not generally possible to reconstruct the whole of the spoken form of an newspaper article from the sentences from it that appear in the speech corpus. The Cambridge VMR group has recently developed a corpus of acoustic training data and spoken messages from the domain of general office correspondence, on which their own experiments in retrieval are performed, but this corpus is not yet generally available to the research community, and was collected too late to be used for the experiments presented in this thesis [72].

Also, it is important that requests and relevance assessments should be obtained, as much as possible, from users unconnected with the actual running of the experiments, rather than by the principal experimenters themselves, or generated artificially. As Tague points out, it is very likely that the experimenters, if working with a document collection for a long time, will become quite familiar with its general content, to the detriment of their ability to supply requests and relevance assessments of the kind that might be expected of members of a general user population.

Finally, there is an extremely important factor in spoken message retrieval, which does not figure in text retrieval. This is the speed at which message representations can be extracted, by whichever recognition paradigm is adopted, from the spoken message. Current state–of–the–art experimental continuous speech recognisers, such as the Cambridge University Engineering Department Speech Group 64,000 word system based on the HTK toolkit [10], require many minutes of CPU time to recognise a single sentence, even on a powerful desktop workstation. This enormous computational overhead currently makes this recogniser impractical for the generation of message representations. Even if a single recognition vocabulary were available, consisting of a large set of terms chosen to cover almost every conceivable query, this state–of–the–art recogniser would take an impractically long time to process every hour of speech.

The recognition task involved in implementing spoken message retrieval is characterised by lack of advance knowledge of the message collection vocabulary. This means that the continuous speech recognition approach, which typically combines detailed acoustic models of a fixed vocabulary of words with a first–order or second–order probabilistic language model, is not an entirely suitable paradigm. A wordspotter is far more appropriate, since wordspotting allows for the detection of an arbitrary set of important words in a spoken message. Whole–word keyword models can be concatenated from acoustic sub–word models, in accordance with keyword phonetic decompositions which can be obtained either automatically, or by look–up in an on–line phonetic dictionary. Also, wordspotting is, in theory, a perfectly suitable recognition paradigm for spoken message retrieval, since it is only necessary to detect query terms in the spoken messages; as mentioned earlier, IR has generally not been shown to benefit from higher–level content representations, for which it would be necessary to generate a more complete transcript of each spoken message.

However, although wordspotting typically involves a lot less computation than large vocabulary word recognition, it would have to be performed over the whole message collection every time a new request was submitted. Flexibility of query term choice would be obtained at the expense of time–consuming query–dependent acoustic matching. Query term detection accuracy would also drop, since a continuous speech recogniser, when used as a wordspotter, has a much better model of non–keyword speech than a conventional wordspotter, and therefore generates far fewer false alarms.

Therefore, neither the continuous speech recognition approach nor the wordspotting approach to the acoustic decoding problem represents a suitable tradeoff between the three factors of flexibility of keyword selection, query–dependent acoustic processing time, and query term (keyword) spotting. It may, however, be possible to trade off a further amount of query term detection performance, to obtain both total flexibility of query term choice and a high speed of wordspotter operation.

In Glavitsch and Schäuble's retrieval paradigm, the problem is avoided since the pre–determined vocabulary of VCV–features does not vary between queries. All the recognition needed to construct the indexing representations of the speech messages can consequently be carried out, once and for all, before any retrieval is performed. Another word–independent approach, which would yield word level indexing units, would be to recognise the contents of each message as a sequence of phones and then to search for subsequences corresponding to query terms. Although many factors, some of which were mentioned briefly at the end of chapter 2, preclude the production of error–free sequences by a phone recogniser, the technique could still be made to work; the adoption of a "fuzzy" matching procedure, like that proposed by Kupiec and illustrated in Figure 5.1, could detect phone sequences in the recogniser output by explicitly allowing for recogniser insertions, deletions and substitutions of phones.

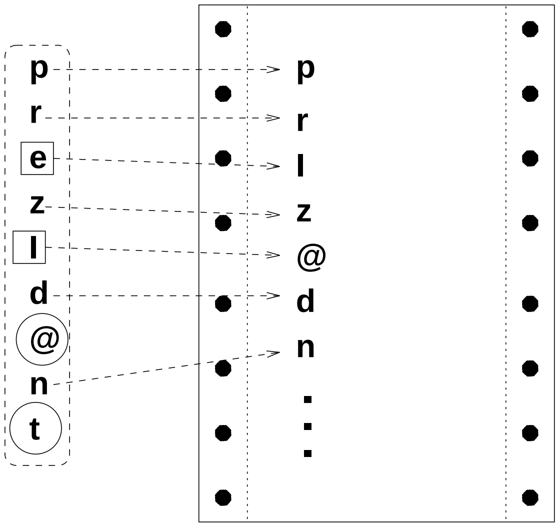


Figure 5.1: The fuzzy phonetic match of the word *president* against the errorful output of a putative phone recogniser. Boxes indicate the keyword phones which are substituted by the recogniser; circles indicate those deleted.

Alternatively, the recogniser could be modified to generate a phonetic representation of each message consisting of multiple differing phone sequences through the message. In practice, the size of this representation could be controlled by enforcing an upper limit on the number of different phone hypotheses that could converge and diverge at any time throughout this message. Query term detection could now be performed by searching the phone–level representation of the message for exact phone strings. Retrieval which involved a query–dependent step of searching for terms in the message collection need no longer involve any time–consuming acoustic matching.

This chapter describes a baseline spoken message retrieval system based on this solution to the query term detection problem. The message collection consists of radio news stories collected on separate occasions in September 1993 and January 1994, representing a total of two–and–a–half hours of speech. Phone models are trained on a larger corpus of training data obtained from the same source. The content of each message in the test collection is obtained and stored in the form of a so–called phone *lattice*, and queries are matched against the messages by searching the set of lattices for the phone sequence corresponding to each query term. Each term hypothesis is given a score reflecting its relative likelihood; a threshold on this score can be altered to vary the message representation, from a few very likely detections of terms to a larger number of terms whose correct detection is less certain. Subsequent sections of this chapter describe the acoustic training corpus and message collection, acoustic model training and testing, the collection of requests and relevance assessments, and the initial retrieval experiments.

5.1 Data Collection and Parametrisation

Many factors suggested the use of a collection of BBC Radio 4 news broadcasts in spoken message retrieval experiments. Firstly, from a data gathering point of view, a large amount of speech can be collected for a fairly small effort. In fact, all the speech used in the experiments in this thesis was recorded using a conventional quartz–locking radio tuner, an ordinary compact cassette deck with automatic tape reverse, Dolby "B" noise–reduction and chrome tape capability, and a domestic electronic timer to allow unsupervised recording of news broadcasts. From a speech recognition point of view, the regular rotation of newsreaders means that the same relatively small number of speakers appear in the acoustic model training set and the message collection, and the high quality FM radio signal and strictly controlled studio acoustics guarantee a clean signal free of cross–talk (where two speakers are talking at the same time) and ambient noise. In addition, the news reports are read, as opposed to being spontaneously spoken, and since all the newsreaders are professional announcers, their microphone manner is relaxed and their diction extremely clear.

Finally, the news bulletin format is quite rigidly adhered to. The newsreader gives a concise summary of each story, introducing, for some of these stories, a more in–depth or "on the spot" report from a professional journalist. Since these reports vary greatly (with differing speakers, differing acoustic channel conditions, for example telephone or studio speech, and potential cross–talk and noise), they are not suitable for current speech recognition techniques. However, the acoustically consistent, studio–based summaries can be thought of, from an IR point of view, as "abstracts" for these reports. The spoken message collection on which experiments were performed can consequently be thought of as indexing a message collection effectively many times larger.

The acoustic model training data was collected over two periods, the first from the middle of August 1993 to the beginning of the following September, the second over four days in January 1994. Speech was recorded in mono on chrome compact cassette from news bulletins broadcast daily at midnight, 6am (in the "BBC News Briefing" programme), 8am (in the "Today" programme) and 10pm (during "The World

Tonight") [73]. Over these two recording periods, the news was read at these times by seven different newsreaders, 4 male and 3 female. The recordings were sampled at a rate of 16kHz into a Sun SPARCstation 2, using an Ariel ProPort 656 A/D–converter. The same cassette deck was used for recording and playback, in order to preserve the effect of the tape transport mechanism on the acoustic characteristics of the recordings. Each sentence was manually endpointed. A total of 3650 individually sampled sentences, representing 6 hours 32 minutes of speech, were obtained in this way. In addition, an orthographic transcription of each sentence was produced manually. An acoustic model training data set was selected by extracting 47 minutes of speech from each speaker, totalling 3030 sentences, from the whole training data set. Around an hour of speech from the whole training set was unused in model training.

Before acoustic model training could take place, it was necessary to extract, from the set of sampled waveforms, a parametric form of the speech suitable for the HMM acoustic modelling. This was in order to perform data reduction and extract important acoustic features that allow for the comparison of trained HMM models and unknown speech. A method which has been shown to be very successful in speech recognition compared to other parametrisations is the Mel–Frequency Cepstrum (MFC) [74]. One of the advantages of using MFC coefficients is that they are relatively uncorrelated, which means that HMM modelling can use diagonal covariance matrices. MFC coefficients were extracted from the digitised speech by first performing a log filter–bank analysis on the contents of a "window" on the sampled speech, using logarithmically–spaced triangular filters, chosen to reflect the human perception of speech. Here, 24 Mel–frequency filter banks were used. Next, a discrete cosine transform was applied to the log filter–bank outputs to yield, every 10ms, 12 MFC coefficients, along with first and second differential coefficients and a log energy value. Appendix A gives the precise parameters of the data coding, as supplied to the *HCode* computer program supplied in Version 1.5 of the HTK Toolkit.

The initialisation and training of phone HMMs generally depends on the availability, for each utterance in the training data collection, of a *phone label file*, which is simply a list of the phones occurring in the utterance in their exact order. A fraction of the training set (200 sentences is generally regarded as sufficient) must be labelled with timings for the start and end of each phone. Model training typically begins with isolated Baum–Welch training techniques used to initialise and re–estimate the models on the 200–or–so utterances with timed label files (these label files usually being obtained by hand–adjustment of phone start and end times using a special label file editing program). This results in a set of models that is sufficiently well trained to allow embedded re–estimation, in which accurate phone start and end times are not required, to proceed over the whole training collection.

However, hand–labelling of any subset of the training data collection is tiresome and difficult. It is possible to initialise phone models with a set of dummy values and perform multiple iterations of embedded re–estimation. However, as was mentioned in chapter 2, the Baum–Welch algorithm is only guaranteed to converge to a *local* minimum, so it is preferable to start embedded re–estimation with good initial models. An alternative approach is to use a set of models trained on another data collection as initial models for the iterative embedded re–estimation process. This method depends on the use of the same speech waveform parametrisation for both training collections and roughly the same acoustic channel characteristics prevailing in both. So long as these conditions hold, factors such as minor differences in the phone labels (such as the difference between sets of phones for American English and British English) do not matter, since the constraint enforced by the phone sequence should ensure that the "imported" models generate sufficiently good model–state to speech–frame alignments when the Baum–Welch algorithm is used to update the model parameters.

Similar thinking underlies the approach to model initialisation adopted in this chapter. Firstly, untimed phone labellings were obtained for the model training set by looking up word pronunciations in a phonetic dictionary derived from Mitton's machine–readable version of the *Oxford Advanced Learner's Dictionary* (OALD) [75, 76]. The dictionary used here contained one pronunciation for every word occurring in the training data. Next, a set of monophone models trained on the American English TIMIT database was obtained, and each model renamed according to a simple mapping of phone labels from TIMIT to the OALD British English phone set, shown in Table 5.1.

A set of timed label files for the entire data collection was now produced by generating a Viterbi alignment of the the phone sequence against the data. This was done by performing standard Viterbi recognition on each training utterance, using a recognition network which forced the recognition of the desired phone sequence. The whole of the training data collection could now be used to create well–trained initial phone models.

5–state[2], single Gaussian density phone models were initialised from model *prototypes* which defined a strict left–to–right model topology. The initialisation was based on the uniform segmentation method as described in chapter 2. The initial models

[2] that is, having 3 emitting states.

Phone Class	OALD Label	Example Word	TIMIT Phone	Phone Class	OALD Label	Example Word	TIMIT Phone
Stops	b	bee	b	Affricates	dZ	joke	jh
	d	day	d		tS	choke	ch
	g	gay	g	Vowels	i	bean	iy
	p	pea	p		I	bid	ih
	t	tea	t		e	bet	eh
	k	key	k		eI	bait	ey
Fricatives	s	sea	s		&	pat	ae
	S	sea	sh		A	barn	aa
	z	zone	z		aU	bow	aw
	Z	azure	zh		aI	buy	ay
	f	fin	f		V	putt	ah
	T	thin	th		O	bought	ao
	v	van	v		0	pot	ao
	D	then	dh		oI	boy	oy
	h	head	hh		@U	boat	ow
Nasals	m	mat	m		U	good	uh
	n	noon	n		u	food	uw
	N	sing	ng		3	bird	er
Semi–Vowels / Glides	l	lay	l		@	about	ax
	r	ray	r		I@	beer	iy
	w	way	w		e@	bear	eh
	j	yacht	y		u@	poor	uw

Table 5.1: The OTA phone set and the mapping from the TIMIT phone set.

were re–estimated in accordance with isolated Baum–Welch re–estimation methods, and then two cycles of embedded re–estimation. The embedded re–estimation process was sped up by setting a *pruning threshold*, which allowed the forward–backward re–estimation process to ignore the most unlikely frame–state alignments during the calculation of state occupation likelihoods. These models then had their output probability distributions enriched to two mixture densities by cloning the single Gaussian mixture, setting equal mixture weights and perturbing the identical mean vectors by adding a multiple of the standard deviation vector to one and subtracting the same multiple from the other. Another two cycles of embedded re–estimation were then carried out on the new 2–mixture models. The number of Gaussian mixture densities were increased and re–estimating continued until a set of 12–component Gaussian mixture models was produced.

5.2 Test Data Collection

A test collection of radio news stories was obtained using exactly the same methods that were used to produce the training data collection. Radio news bulletins were recorded daily, at midnight and 6pm, for a week–long period in September 1993 and over ten days in January 1994. Those bulletins spoken by any of the seven newsreaders whose speech appeared in the model training set were sampled into the SPARC-station 2 and the remainder were discarded. The set of bulletins was edited so that each single waveform file corresponded to one individual news story, and where a news story had occurred identically in two consecutive bulletins (*i.e.* where there had been no developments in a news item between a 6pm bulletin and the subsequent midnight one), the recording from the subsequent bulletin was discarded. The application of these techniques resulted in a test collection 2 hours and 27 minutes in length containing 337 separate news stories, with a mean of 22 minutes of speech from each newsreader. Table 5.2 illustrates the contribution of each newsreader to the test collection. Although these contributions vary a great deal in size, the collection contains roughly similar amounts of speech from male and female speakers.

Newsreader	Gender	Amount of Test Speech (Minutes)
AC	Male	21.0
AJ	Male	4.2
BP	Male	27.3
CG	Female	53.5
CY	Female	8.7
LM	Female	18.3
PD	Male	19.7

Table 5.2: Each newsreader's contribution to the spoken message collection.

Thinking of the test data collection as consisting of surrogates for the entire collection of news bulletins from which it was extracted, and assuming a mean length of 30 minutes for the 6pm bulletins and 25 minutes for the midnight ones, the "effective" amount of speech that can be indexed in the spoken message retrieval task is $10\frac{1}{2}$ hours. Appendix B illustrates a number of these stories.

Orthographic transcriptions were input for the entirety of the test data collection. These were used, in conjunction with the phonetic dictionary, to obtain untimed phone labellings for the test set. These labellings would subsequently be used in recognition performance evaluation. The set of test waveform files was parametrized and speech vector files containing MFC coefficients extracted using the exact method used for the training speech.

As an initial gauge of the ability of the trained monophone HMMs to model the test speech, phone recognition was performed on the whole test collection, using a simple *phones–in–parallel* recognition network, illustrated in Figure 5.2, in which the recognition path was allowed to pass through any phone at any time. The transition between any two phones was weighted by a "phone bigram" probability estimated from the training data. This probability was transformed into the log domain and multiplied by a scaling factor (in this case, 5.0). Such a scaling factor is typically chosen to select a tradeoff between the number of phone insertions and deletions made by the recogniser, and is added to the path log likelihood every time a token exits from a phone model and is injected into the entry state of a potential successor. The set of recognised phone transcriptions was compared with the correct phone sequences for the 337 news stories, using the HTK tool HResults to generate the best alignment of each correct and recognised sequence. Measuring recogniser performance in terms of phone correctness and accuracy, two widely–used measures of recogniser performance corresponding strongly to recall and precision respectively, phone correctness was 67.92% and phone accuracy 65.29%. The relative ease of recognition on the spoken message collection is illustrated by comparison with the best phone recognition rates that have been obtained on the test corpus of the speaker–independent TIMIT collection by Robinson *et al* [77]. Their IPA system depends on explicit durational modelling and a hybrid connectionist/HMM strategy, and performs with 74.2% correctness and 71.6% accuracy. The fact that the simple recognition strategy used here can perform with 65% accuracy reflects of the use of a fixed set of trained speakers throughout, as opposed to differing untrained speakers for model training and testing.

5.3 Lattice Generation

A *lattice* is defined formally as a connected loop-free directed graph. If each node of the graph is associated with a point of time during a message, and each edge labelled with a phone hypothesis and a score representing the likelihood of that hypothesis, then the lattice can be used to store multiple potential phone sequences output by a speech recogniser. Wordspotting can then be performed by searching the contents of the lattice.

Lattice generation in the experiments presented here is based on an extension of

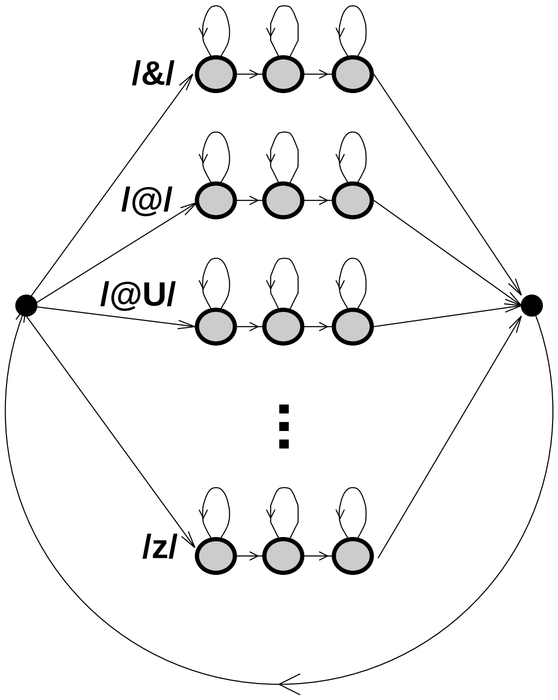


Figure 5.2: The parallel phone network used in initial model tests and subsequent lattice generation.

the Token Passing Paradigm. The extension, proposed by Lucke [78], is based on the use of a phones–in–parallel grammar and the storage, at every time t throughout the unknown speech, of the tokens exiting the N most likely models, as opposed to the single best model. When all the unknown speech has been observed, the standard Viterbi "traceback" procedure is modified so that multiple sequences of the recognised phones are traced through and output, in such a way that for every time t, at most N lattice edges, each corresponding to a single phone hypothesis, converge, and at most N diverge. Figure 5.3 shows a example section of a phone lattice for which $N = 2$. Each lattice edge is labelled with the phone it represents and the log likelihood score for that phone.

Phone lattices represent one approach to a speech recognition task for which no vocabulary of recognisable words is available in advance. It was noted above that the Viterbi recognition performance of the monophone speech models was 67.92% correctness and 65.29% accuracy. Many of the errors made by this speech recogniser are confusions between two members of a single broad acoustic–phonetic class, for example nasals or fricatives. The lattice approach offers a number of alternative phone

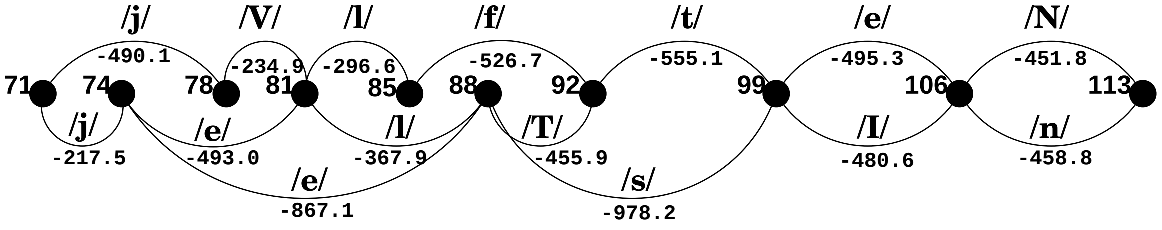


Figure 5.3: A section of lattice of *degree* 2 for the word *yeltsin*.

hypotheses where a "best–sequence" phone recogniser might make a substitution. It should therefore be possible to detect, in a lattice, the phone sequence making up a particular word, whereas it would not typically be possible to detect the same word in a single phone sequence.

Kupiec's approach to constraining phone matches in his isolated word recognition task was to estimate, from the acoustic model training data, the probabilities of the Viterbi recogniser inserting, substituting or deleting specific phones [29]. In the lattice–based wordspotter, the amount of flexibility in the matching procedure is controlled by varying the *degree* of the lattice. This is defined as the maximum number of phone edges allowed to converge and diverge at any point throughout the speech. A lattice of degree N will be referred to as an N–lattice. A lattice with a high degree contains a greater number of differing phone hypotheses for any speech sound, so there is a greater chance of correctly spotting any keyword, although with a greater probability of generating a false alarm for another keyword during the same segment of speech.

The matching procedure adopted here consists simply of finding the correct phone sequence within the phone lattice, with no possibility for phoneme insertion, deletion or substitution, so that the end of any required lattice edge and the beginning of the next one are at the same lattice node. A score can be obtained for a keyword occurrence detected in this way by adding up the log likelihood scores of the correctly detected edges, subtracting the score from the score for the corresponding section of the most likely (*i.e.* Viterbi) phone path, and dividing by the length of the hypothesized keyword. The resulting score is a durationally–normalised log likelihood ratio (DNLLR) score, of the kind first suggested by Rose and Paul, and shown by them to perform well in discriminating correctly–detected keywords from false alarms [59].

5.4 Queries and Relevance Assessments

Eight members of the Speech Vision and Robotics Group of the University of Cambridge Engineering Department agreed to supply requests and relevance assessments for speech message retrieval experiments on the spoken news message collection. A total of 40 requests and associated relevance assessments were obtained for message retrieval experiments on the radio news database. Although a test collection comprising around 340 documents and 40 requests is extremely small by text retrieval standards, this collection is comparable in size with the other corpora on which message retrieval experiments have been proposed or performed [7, 69].

To avoid the overuse of the word *topic* and underline the distinction between spoken message retrieval and topic classification, the word *thread* will be used to describe a set of individual news messages, all addressing the same broad subject, but recorded over a number of days. In contrast to the "topics" of topic classification experiments, the set of threads does not comprise an exhaustive partition of the message collection; some messages may be members of more than one thread, and some may not be members of any.

Where the collection of requests and relevance assessments is strongly user–dominated, it is common for the document collection to contain items from a specific field of academic enquiry in which the user is expert. The user should thus be able to issue requests without any intervention from the experimenter. However, although general familiarity with the format and content of radio news broadcasts was quite high amongst the eight users, the content of the collection had unavoidably aged by the time experiments were carried out. It could consequently not be guaranteed that the users could produce a sufficient number of usable requests without some assistance. Therefore, the radio news collection was studied in detail by the experimenter and 40 threads identified which it was thought would yield usable requests. Requests were obtained by prompting each user with a fairly terse phrase referring to each news thread. The request prompts are given in Table 5.3.

For each thread the user was prompted to type a list of content words that he or she judged to be related to the thread. On average, each request contained 6.73 words. Example requests for a number of threads are illustrated in Table 5.4.

Relevance assessments were supplied for each request by the user who generated that request. It was decided that assessments should be obtained as exhaustively as possible for each request. To limit user fatigue, however, a first pass was made through the collection by the experimenter, in order to remove, for each thread, the most grossly irrelevant messages from the set of messages to be assessed for that

Political Crisis in Russia	Football
The Troubles in Northern Ireland	The UK Press
The Home Secretary's plans to reform the Police	UK Job Losses
The former Soviet Union	Industrial disputes
The Labour Party in the House of Commons	Train crashes
United Nations Operations in Bosnia	The British economy
Russian Economic Reform	Football Results
Terrorist Activity in Northern Ireland	Football (off the pitch)
Separatist violence in Georgia (former Soviet Union)	The City
The National Health Service	News about Russia
Politics in the United States	UK Jobs
The 1993 Labour Party Conference	Golf
The Arms To Iraq Enquiry	Iraq
The Northern Ireland Peace Process	Sport
The Conservative Party in the House of Commons	Drug seizures/deaths
The Middle East Peace Process	UK Job creation
Bosnian peace talks in Geneva	Reform in South Africa
The Los Angeles Earthquake of January 1994	Earthquakes
War in the former Yugoslavia	The British Labour Party
Primary and Secondary Education in the UK	Fraud trials in the UK

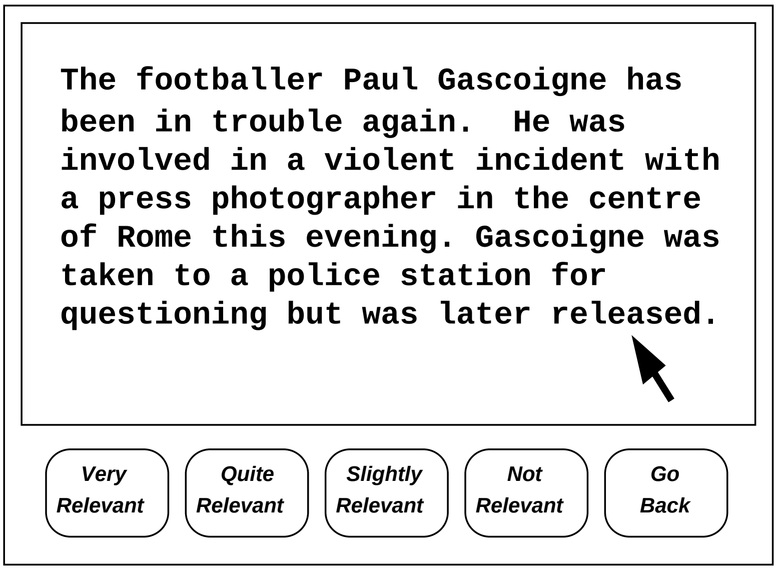
Table 5.3: The 40 prompts used to assist request generation.

request. This was felt to give the best reduction in user effort without compromising the overall quality of the relevance assessments. For each thread, the remaining stories to be assessed were displayed in text form to the user using a graphical tool based on the HGraf module of the HTK Toolkit. Figure 5.4 illustrates this program in operation. For each message, four soft buttons were displayed, each representing a certain degree of assessed relevance, from "Very Relevant" to "Not Relevant". Clicking on any of these recorded a decision about the displayed story and displayed the next in the sequence of unassessed messages. A "Go Back" button allowed the assessor to correct any errors. For experimental purposes, the assessments for each

Prompt	Request
Train Crashes	*accidents br crashes derailed railways trains*
Reform in South Africa	*africa deklerk elections mandela reform south*
Earthquakes	*disasters earthquakes faultlines natural richter tsunamis*

Table 5.4: A number of requests with the prompts that generated them.

of the 40 requests were merged to create a simple relevant/non–relevant distinction, yielding an average of 11.75 news stories relevant to each request.



The footballer Paul Gascoigne has been in trouble again. He was involved in a violent incident with a press photographer in the centre of Rome this evening. Gascoigne was taken to a police station for questioning but was later released.

Figure 5.4: The operation of the computer program used to obtain relevance assessments from users.

5.5 Initial Retrieval Experiments

The majority of message retrieval experiments described in this chapter were performed using phone lattices as the only acoustic representation of the test collection messages. The queries were taken to be identical to the supplied requests. Query term occurrences were detected in the spoken messages using the lattice wordspotting method described in the previous section. Message lattices of varying degrees were generated using the 12–mixture Gaussian monophone HMM models. The scores output by the wordspotter for each hypothesized keyword were subject to a varying cross–keyword threshold so that the size and acoustic accuracy of each message representation could be altered and the resulting effect on retrieval performance observed.

It will be assumed here that the accuracy of spoken message retrieval is bounded above by the accuracy obtained when the correct textual representation of the entire message collection is available. However, this is not necessarily true, as it cannot be assumed that the queries obtained from the human users of the system are optimally able to retrieve the messages that the users have judged to be relevant to their information need. If, for example, a wordspotter were to detect all the occurrences of a recall–enhancing, "good" term in the message collection, but miss all the occurrences of a precision–reducing "bad" term, then the retrieval performance on wordspotter output would be *better* than that obtained, using the same query, on the exact text of the message collection. Although a situation such as this may conceivably occur for one or two short queries, it is certainly not typical behaviour over a set of 40 queries of varying size and discriminability.

5.5.1 Textual Reference Run

Firstly, retrieval was performed using the exact textual transcripts of the items in the message collection, in order to generate *best–case* results against which the performance of the spoken message retrieval could be compared. The matching of queries against textual messages can be performed quickly if the textual message representations are stored in a data structure which allows individual messages to be searched instantaneously for query term occurrences. An *inverted file* is just such a data structure [4]. In an inverted file, the entire textual message collection is stored in such a format that an indexing function, if given a specific word as its argument, can instantly return a set of pointers to all its occurrences in the message collection. The inverted file system used in these experiments was implemented using a hash table, a simple hashing function based on the initial letters of a word, and lists of word occurrences throughout the collection, as shown in Figure 5.5. In addition, function words were removed from the textual messages, in accordance with van Rijsbergen's *stop list* [4], and the remaining terms stemmed, to increase the number of query term matches.

Matching was performed by searching for the occurrences of each query term and adding an appropriate term weight to the running total for each message for the given query. A list of message identifiers and their query–matching scores was output by the matching program in a form readable by the TREC scoring software [46], a widely recognised standard tool used in the yearly ARPA TREC evaluations. The TREC software reads in the retrieval output and the lists of relevant messages for each query, and calculates a number of performance measures, including a set of precisions after the retrieval for each query of N documents, for varying N, and a non–interpolated average precision after the retrieval of all relevant documents for all queries. These measures were described at the end of Chapter 3.

The results for the text–based retrieval experiments are shown in Table 5.5. Two strategies for the calculation of term "base" weights were combined, multiplicatively, with three term frequency methods, generating a total of six differing weighting schemes.

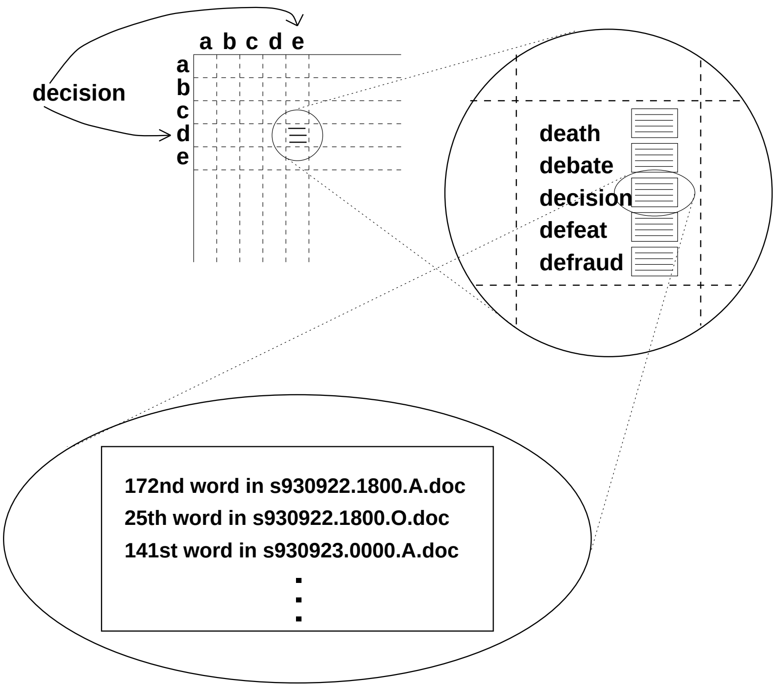

a b c d e
a
b
c
d
e
decision
death
debate
decision
defeat
defraud
172nd word in s930922.1800.A.doc
25th word in s930922.1800.O.doc
141st word in s930923.0000.A.doc

Figure 5.5: A schematic diagram of an inverted file.

Each strategy is given a single letter label. As in Salton and Buckley [35], terms are assigned either a binary base weight (x), reflecting simply their presence or absence in a message, or Croft and Harper's probabilistic IDF weight (p) [44]; they are also given either a binary within–document frequency component (b), the full term frequency (t), or the normalised term frequency (n). Performance is given in terms of the non–interpolated average precision and the interpolated average precisions recorded after the retrieval for each query of 5, 10 and 20 messages.

	bx	tx	nx	bp	tp	np
AvePrec	0.6448	0.6475	0.6936	0.6752	0.6889	0.7038
Prec@5	0.6800	0.7100	0.7150	0.7050	0.7400	0.7300
Prec@10	0.5675	0.5825	0.6075	0.5675	0.5850	0.6000
Prec@20	0.4000	0.4062	0.4113	0.4200	0.4200	0.4238

Table 5.5: Performance for differing term weighting schemes of reference retrieval run.

Firstly, it must be stated that the retrieval of news stories from the collection was relatively easy, with average precisions around the 0.7 mark. This was due mainly to the small size of the collection, numbering only 337 messages, and the small numbers of documents relevant to each query (an average of around 11). To illustrate this, the precision at a recall level of 0.3 was found to be 0.8423. In comparison, the yearly IR evaluation task, TREC, is based on a collection of around a million documents, with relevance sets of the order of hundreds of documents in size. As a result, precision on *ad hoc* retrieval at a recall level of 0.3 is typically around 0.5, indicating that retrieval on the TREC collection is considerably harder.

It can be seen from the table that both forms of term weighting led to improvements in average precision and in the precision at 5, 10 and 20 messages. The average precision for np–weighted terms was almost six *precision points* (where a precision point is simply a difference in precision of 0.01) better than for the quorum (bx) matching approach. The IDF weight performed well because it increased the relative importance of less frequent items; for example, in one of the queries, which in its unstemmed form contains the terms *georgia*, *government* and *protest*, the contribution of *government* towards the query–message scores was reduced owing to its high occurrence rate across a number of differing threads concerned with government in many countries.

Regarding the term frequency component, only small improvements in average precision were obtained when the base weight of a term (either 1.0 or the IDF weight) was multiplied by its frequency of occurrence in a message to give the term's contribution to the query–message score. To assume that a query term occurring, say, 3 times as frequently as another in a message is three times as "important" in the calculation of matching scores seems too broad a generalisation. It was possible, however, to exploit term frequencies without making such gross assumptions. Salton's within–document normalised term frequency weighting scheme compressed the range of term frequency values so any frequently occurring term in a message was now at most twice as "useful" as any infrequently occurring term. The use of this weighting scheme increased most precision values significantly compared to the simple binary term frequency value.

5.5.2 Viterbi wordspotting

The first spoken message retrieval experiments were performed using a conventional implementation of wordspotting to detect query terms. The 40 queries were merged to produce a single query vocabulary consisting of 206 separate words. Word models were built by concatenating the appropriate subword models, and a garbage model was built by an agglomerative clustering of the model states, producing a set of garbage monophones which would be deliberately worse at modelling the speech [79]. The models were placed in a conventional wordspotter recognition network, as illustrated earlier in Figure 4.2, and the HTK Viterbi recogniser HVite used to decode each spoken message into a string of query term and garbage–word occurrences. The recognition output was rescored using the output of the unconstrained phone recognition pass, which was performed earlier, to obtain DNLLR scores for each query term. The message representations to be used in retrieval were varied, by setting a single word–independent threshold on the wordspotter output scores. Retrieval was performed a number of times for a range of wordspotter threshold values, applying the weighting schemes introduced in the earlier section, and at a threshold of $-\infty$ (*i.e.* no threshold).

Table 5.6 and Figure 5.6 illustrate the results obtained for this experiment. The most striking feature of this table is the peaking of retrieval average precision at a threshold value of -1.3 for the x results and -1.5 for the p results. At these values the message representations were optimal with respect to the wordspotter output, the set of queries and relevance assessments, and the weighting schemes used.

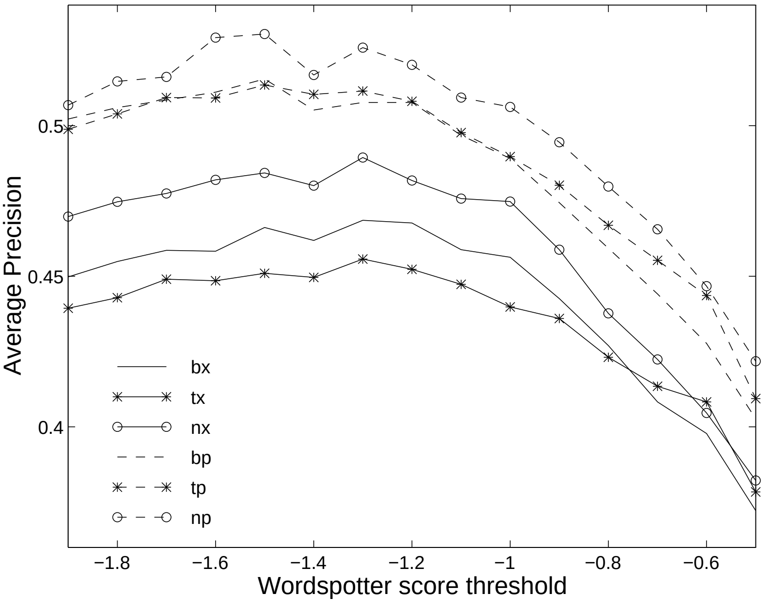


Figure 5.6: Viterbi wordspotter retrieval performance.

The differing optimal threshold values for the two forms of the term base weight may be explained as follows. Without the use of inverse document frequency weight-

Threshold	bx	tx	nx	bp	tp	np
-0.5	0.3722	0.3784	0.3822	0.4027	0.4094	0.4218
-0.6	0.3978	0.4083	0.4046	0.4277	0.4436	0.4467
-0.7	0.4083	0.4135	0.4224	0.4441	0.4553	0.4656
-0.8	0.4270	0.4231	0.4377	0.4593	0.4669	0.4798
-0.9	0.4426	0.4360	0.4588	0.4744	0.4802	0.4945
-1.0	0.4563	0.4398	0.4748	0.4891	0.4897	0.5062
-1.1	0.4588	0.4473	0.4758	0.4968	0.4977	0.5093
-1.2	0.4677	0.4523	0.4818	0.5077	0.5081	0.5202
-1.3	0.4686	0.4557	0.4894	0.5077	0.5115	0.5259
-1.4	0.4619	0.4496	0.4801	0.5052	0.5104	0.5168
-1.5	0.4662	0.4510	0.4843	0.5155	0.5135	0.5304
-1.6	0.4583	0.4485	0.4820	0.5111	0.5092	0.5292
-1.7	0.4586	0.4490	0.4775	0.5085	0.5093	0.5162
-1.8	0.4549	0.4429	0.4747	0.5060	0.5039	0.5147
-1.9	0.4498	0.4394	0.4698	0.5022	0.4988	0.5068
$-\infty$	0.4064	0.3915	0.4426	0.4690	0.4642	0.4788

Table 5.6: The average precisions of retrieval on Viterbi wordspotter output.

ing, the retrieval system was not able to discriminate between infrequent occurrences of useful query terms and frequent occurrences of false alarms. The precision without IDF weighting dropped between thresholds of -1.3 and -1.5, because lowering the threshold caused the "contamination" of the message representations by false alarms for several query terms (typically, the shortest ones). These false alarms "swamped" the correctly spotted terms. With IDF weighting, however, terms with a large number of false alarms were assigned a low base weight; term detections incorporated into the message representations by the threshold decrease were now either poorly–weighted false alarms, which did not interfere with the retrieval process to a great extent, or useful term occurrences, which were assigned higher weights. IDF weights for a number of terms are shown in Table 5.7. As the threshold was further decreased, smaller and smaller numbers of correctly recognised terms were added to the message representations, at the expense of including low–scoring false alarms for a wide range of terms. The "filtering" performed by the IDF weighting scheme therefore began to fail, and precision values decreased.

It was remarked in chapter 4 that the Figure of Merit was limited as a practical metric of wordspotter performance. It is not entirely suitable here, as the spoken message retrieval performed in this experiment has depended only on a single thresh-

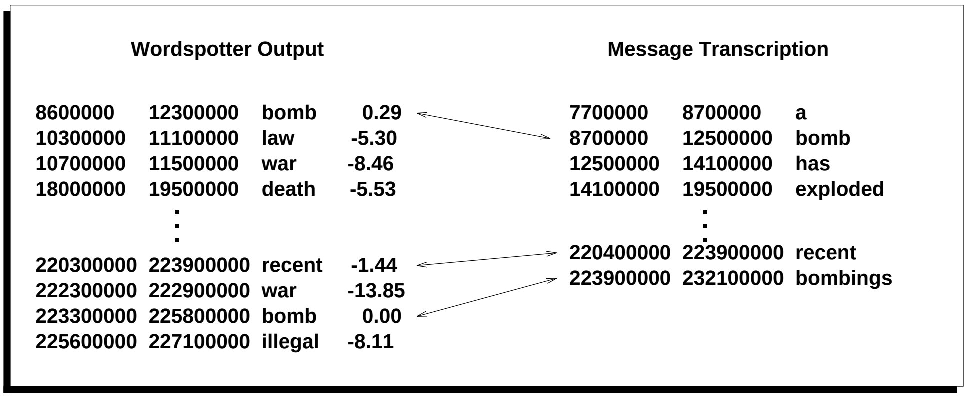
Term	Number of messages	IDF weight
war	171	0.2946
british	27	1.0963
conservative	10	1.5276

Table 5.7: Sample query terms with the number of messages they are hypothesized in, at an acoustic threshold of -1.5, and the corresponding IDF weights.

old for all the keywords. In conventional wordspotting, a posterior score threshold can be approximated by training a keyword–dependent transition penalty. This can be done by training whole–word keyword models and then iteratively recognising the contents of the training speech and adjusting model–specific transition penalties until the "best" tradeoff between correct word detection and false alarm generation is achieved [59]. This approach can obviously not be employed here, however, since there is no relation between the acoustic training data vocabulary and the set of query terms. Another problem here with the Figure of Merit is that owing to its formulation, it is a function only of the detection performance on query terms occurring *at least once* in the message collection. The hypothesis of a large number of false alarms of a totally spurious query term (such as *tsunamis* from Table 5.4), which could significantly affect retrieval effectiveness, is not reflected in its calculation.

As long as its deficiencies are borne in mind, however, the Figure of Merit can serve as a simple characterisation of the ability of the wordspotter to detect query terms and suppress high–scoring false alarms. The FOM of the Viterbi wordspotter was measured and found to be 67.79%. It was calculated by comparing the wordspotter output with a set of timed textual transcriptions of the message collection, counting as a "hit" the hypothesis of a query term where what actually occurred in the test data was a obvious variant of that term, as illustrated in Figure 5.7.

Spoken message retrieval using standard wordspotting techniques, it can be seen, can perform quite effectively, given the simplicity of the acoustic models and the lack of a grammar to constrain query term detections. If, as in Schäuble and Glavitsch's work, spoken message retrieval effectiveness is expressed as the ratio between the highest average precision achieved on spoken message retrieval, with that obtained that obtained on textual transcriptions of the messages, then an *effectiveness ratio* figure of 75.36% was obtained between text and spoken message retrieval with the np–weighting scheme, at a score threshold of -1.5.



Wordspotter Output

8600000	12300000	bomb	0.29
10300000	11100000	law	-5.30
10700000	11500000	war	-8.46
18000000	19500000	death	-5.53
⋮			
220300000	223900000	recent	-1.44
222300000	222900000	war	-13.85
223300000	225800000	bomb	0.00
225600000	227100000	illegal	-8.11

Message Transcription

7700000	8700000	a
8700000	12500000	bomb
12500000	14100000	has
14100000	19500000	exploded
	⋮	
220400000	223900000	recent
223900000	232100000	bombings

Figure 5.7: The comparison of wordspotter output with text transcripts of the messages. The double–headed arrows indicate correct term detections.

5.5.3 Lattice wordspotting

Conventional wordspotting, as has already been explained, is unfortunately not a suitable method for spoken message retrieval, since the wordspotter would have to be rerun on the entire $2\frac{1}{2}$ hour message collection for every newly–submitted query. The wordspotting performed for the experiments in the previous subsection was timed at 16 CPU hours on a fast 32–bit desktop workstation.

The advantage of the lattice–based wordspotter is that the lattice generation depends solely on the phones of the modelled language and not on any particular words. This means that recognition need only be performed once, well in advance of the specification of any queries, and that wordspotting on the intermediate lattice representation of the speech has the potential to be be extremely fast, provided that the lattice can be loaded and stored in a data structure, similar in design to the *inverted file* of text IR, which would allow for the fast location of required phone occurrences.

The first lattice–based retrieval experiment was intended to determine whether the intermediate wordspotting approach could be made to work in the case where the phone lattice was simply the single phone sequence output by a conventional Viterbi phone recogniser. Although the correctness and accuracy of these phone sequences, as already obtained, were 67.92% and 65.29% respectively, the potential utility of these sequences in retrieval was still not known. The Viterbi phone transcriptions were loaded into the lattice wordspotter along with the phonetic decomposition of each query term. Wordspotting was performed by detecting, for each query term, the exact string of phones occurring continuously in the 1–lattices. Of the 206 query terms, pronunciations for all but 33 were obtained immediately from the on–line version of the Oxford Advanced Learners' Dictionary. The remainder were supplied by the experimenter. it was found that the large majority of the unknown words were proper nouns and abbreviations. Recent experiments have shown the effectiveness of *letter–to–sound rules*, usually used in speech synthesizers, to generate phone sequences for use in recognition [80].

Since the lattices used were the phone sequences output by the Viterbi phone recogniser, it should be clear that the DNLLR score of each term hypothesis was zero, so no thresholding could be performed on the set of query term detections. Table 5.8 shows the precisions obtained here, for the usual term weighting schemes. The query–dependent search time was negligibly small, but retrieval was unacceptably ineffective, with precision values at around half the level achieved by the system based on Viterbi wordspotting, and the best effectiveness ratio value being 36.18%. The wordspotter figure of merit was equally poor, at 21.02%.

	bx	tx	nx	bp	tp	np
AvePrec	0.2248	0.2333	0.2295	0.2546	0.2631	0.2534
Prec@5	0.3300	0.3300	0.3350	0.3650	0.3750	0.3500
Prec@10	0.2675	0.2650	0.2550	0.2700	0.2675	0.2625
Prec@20	0.1638	0.1750	0.1650	0.1900	0.1875	0.1863

Table 5.8: Performance for differing term weighting schemes of retrieval based on 1–lattice wordspotting.

Each of the two approaches described so far obviously represents an unacceptable tradeoff between retrieval effectiveness and search time. The Viterbi wordspotter delivered acceptable retrieval performance but took far too long to detect query terms in the unknown speech; on the other hand, the 1–lattice system retrieved messages almost instantaneously, since the method depends on spoken message representations obtained only once and well in advance, but was much less able to satisfy the user's information need.

In the next experiment, wordspotting was performed on 4–lattice representations of the spoken messages. As outlined earlier, the extension of the Viterbi paradigm to generate phone lattices instead of single phone sequences is quite simple and the one–off acoustic search time required to generate phone lattices is no greater than that required to generate phone sequences. However, the time required to search the lattice representations for query terms increases as the number of possible phone paths at each time increases. The results for retrieval on 4–lattices are shown in Table 5.9 and Figure 5.8, for varying values of the log likelihood ratio score threshold and the np–weighting scheme, which consistently outperformed the other 5 weighting schemes tested in this experiment. It can easily be seen that the 4–lattices were far more able to encapsulate the query term content of the unknown messages than the 1–lattices. The effectiveness ratio on np–weighted retrieval rose to 63.13%, and the wordspotter FOM more than doubled, to 48.87%.

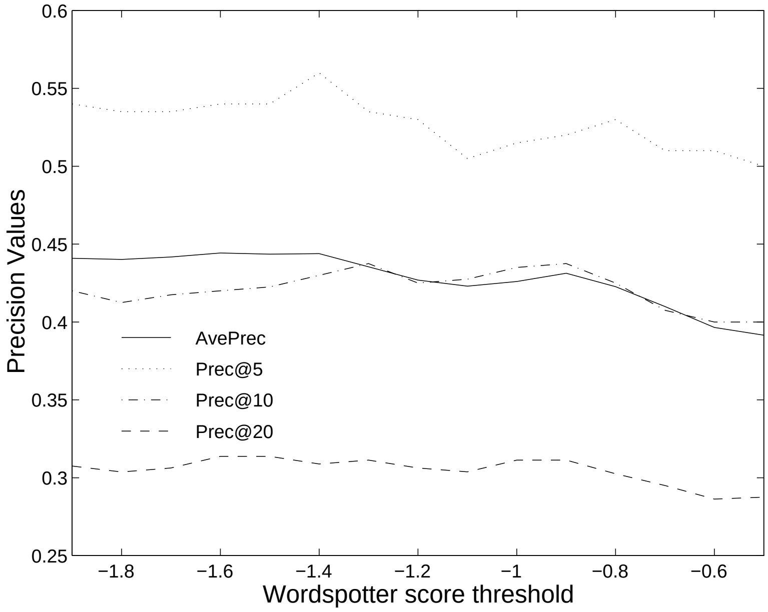


Figure 5.8: 4–lattice wordspotter retrieval performance.

Finally in this section, wordspotting was performed on 8–lattices. The results are shown in Table 5.10 and Figure 5.9. Comparison between the average precisions given in Table 5.10 and the final column of Table 5.6 shows that the 8–lattice system delivered precision values close to those achieved for by the Viterbi wordspotting system. The np–effectiveness ratio here was 70.63%, and the wordspotter figure of merit 60.46%.

5.6 Analysis of the results

5.6.1 The effect of thresholding and term weighting

Firstly, it is noticeable that in each of the thresholded lattice wordspotters, the best performance is generally located at threshold values of around -1.5, and there are only

	Acoustic Keyword Thresholds															
	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	$-\infty$
AvePrec	0.3915	0.3965	0.4099	0.4227	0.4313	0.4260	0.4230	0.4269	0.4354	0.4439	0.4436	0.4443	0.4417	0.4402	0.4409	0.4035
Prec@5	0.5000	0.5100	0.5100	0.5300	0.5200	0.5150	0.5050	0.5300	0.5350	0.5600	0.5400	0.5400	0.5350	0.5350	0.5400	0.5600
Prec@10	0.4000	0.4000	0.4075	0.4250	0.4375	0.4350	0.4275	0.4250	0.4375	0.4300	0.4225	0.4200	0.4175	0.4125	0.4200	0.4050
Prec@20	0.2875	0.2863	0.2950	0.3025	0.3113	0.3113	0.3038	0.3063	0.3113	0.3088	0.3137	0.3137	0.3063	0.3037	0.3075	0.2863

Table 5.9: Performance for differing keyword thresholds of np–weighted retrieval based on 4–lattice wordspotting.

small variations in the average precisions around this value. All the results tabulated in the previous section show that, as for the text case, retrieval performance improves when within– and across–document frequency weighting is used. Specifically, IDF weighting is shown to be of universal benefit, and the compressed term frequency weighting, n, outperforms full term frequencies, t, on almost every occasion. In only the naïve 1–lattice experiment did full term frequency weights perform better; this was presumably because the word detection accuracy was so poor that no improvement could be expected by compressing the already–low range of term frequencies. It would clearly be a mistake to attach too much importance[3] to this result.

Just as the effectiveness ratio measures the average precisions of retrieval for speech and text relative to each other, a similar comparison can be performed for the "Precision after N messages" figures. For the most part, speech/text ratios between these values are consistently higher than the corresponding effectiveness ratio. Spoken messages which contain a large number of query terms are obviously relatively easy to retrieve, since there are multiple opportunities for the wordspotter to detect at least a few occurrences of the required terms. Where a query term only occurs once in a relevant message, this term may be missed entirely by the wordspotter. It is therefore difficult for the spoken message system to retrieve "outlying" relevant messages.

It should be mentioned here that experiments were also performed in which query–

[3] or, indeed, weight.

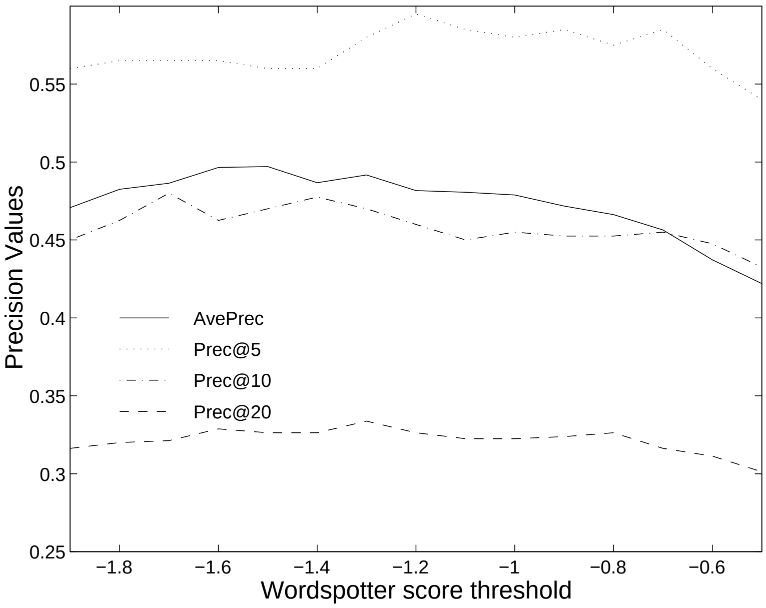


Figure 5.9: 8–lattice wordspotter retrieval performance.

message scores were normalised with respect to message length. However, this was not found to improve retrieval effectiveness. It is a particular attribute of the spoken news collection that those messages derived from the news *headlines* were longer than average. Since these headlines typically addressed "running" news stories, they were typically chosen by the experimenter as news threads for the retrieval experiments. Of the 23 messages corresponding to news headlines, 22 were assessed as relevant to at least one query; of the 337 messages in the entire collection, 234 were assessed as such. Consequently, since most long messages are relevant to queries, and quite easily retrieved on account of their length, message–length normalisation did not improve performance. This should, of course, not be taken as a general attribute of spoken message collections.

5.6.2 Varying the wordspotter threshold

The results presented in the previous section demonstrate that retrieval precision peaks at some value in the middle of the range of values of the log likelihood ratio threshold. The nature of the relationship between precision and threshold is not obvious, however, since any observed improvement in performance depends on several distinct and unpredictable factors, such as relevance assessments, the choice of terms

	Acoustic Keyword Thresholds							
	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2
AvePrec	0.4220	0.4373	0.4564	0.4663	0.4717	0.4789	0.4806	0.4817
Prec@5	0.5400	0.5600	0.5850	0.5750	0.5850	0.5800	0.5850	0.5950
Prec@10	0.4325	0.4475	0.4550	0.4525	0.4525	0.4550	0.4500	0.4600
Prec@20	0.3013	0.3113	0.3163	0.3263	0.3238	0.3225	0.3225	0.3263
	Acoustic Keyword Thresholds							
	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	$-\infty$
AvePrec	0.4917	0.4867	0.4971	0.4965	0.4864	0.4825	0.4707	0.3422
Prec@5	0.5800	0.5600	0.5600	0.5650	0.5650	0.5650	0.5600	0.4450
Prec@10	0.4700	0.4775	0.4700	0.4625	0.4800	0.4625	0.4500	0.3350
Prec@20	0.3338	0.3263	0.3263	0.3288	0.3213	0.3200	0.3163	0.2375

Table 5.10: Performance for differing keyword thresholds of np–weighted retrieval based on 8–lattice wordspotting.

in each query specification, and the exact nature of the acoustic modelling process.

As was explained in Chapter 4, the *ad hoc* nature of querying in information retrieval does not permit, in the speech case, the use of pre–determined acoustic thresholds or functions to transform a log likelihood score into a more useful measure of acoustic uncertainty. Nor is it possible to adopt the approach taken by McDonough *et al* , and obtain a word detection probability (in their case, the forward–backward probability of word–final state occupation), directly from the recognition pass. This would entail the selection of a fixed word recognition vocabulary, thereby sacrificing wordspotter ability to detect any query term.

The only approach to term–dependent thresholding that can be taken here is the addition of a fixed penalty to the log likelihood ratio scores output by the wordspotter before the scores are normalised by term duration to produce the DNLLR score. Since the penalty is fixed, it penalises the scores of short query terms more than long query terms [56]. This method has the advantage that it can be implemented simply in the retrieval stage and that a number of differing penalties can be tried without the need for re–running the wordspotter. Since the penalty does not change the ordering of detections of each individual term, but instead has the effect of varying the threshold value across terms, such a penalty does not affect the wordspotter Figure of Merit.

An experiment was performed using the method outlined above, setting the fixed penalty to -5.0, -10.0 and -15.0 in turn. It was found that although the use of the penalty had the general effect of improving the co–ordination level (bx) matching, no improvement in np–weighted retrieval was obtained. This suggests again that the use of term weighting is sufficient compensation for noisy term detection.

The single term score threshold method can be thought of as a limiting case of Rose, Chang and Lippmann's sigmoid functions [48], with all detected query terms with a score below the threshold assigned a "retrieval utility value" of 0, and all others given a value of 1. Its advantage is that it does not need to be trained; the threshold is simply varied across a range of values. The main disadvantage of the single threshold system is that differing words are hypothesized, above a fixed threshold, at widely differing rates of correct detection and false alarm; if an "optimal" *a posteriori* threshold were to be set on each keyword in accordance with some criterion (for example, the optimal threshold could be defined as the highest threshold after which the next three detected terms are false alarms), the thresholds would exhibit great variation across the set of keywords. As can be seen from Table 5.11, which illustrates the Viterbi wordspotter detection rates, at a threshold of -1.0, for a small number of the query terms, the detection of short terms is particularly inaccurate. The importance of these terms in message retrieval can be determined simply by removing from the queries the set of terms, illustrated in Table 5.12, whose pronunciation consists of only three phones, and performing retrieval again on textual message representations. The np–weighted average precision without the short terms is significantly lower, at 0.6615, than the figure of 0.7038 obtained when they are included.

Term	#Occs	#Detected	#False Alarms	Accuracy
arms	7	1	28	-385.7
bcci	4	2	0	50.0
city	36	31	17	38.89
downing	8	6	4	25.0
exchange	15	12	0	80.0
fein	11	10	18	-72.72
geneva	6	6	0	100.0
home	38	28	28	0.0
israeli	5	4	0	80.0
john	37	24	7	45.94
kuwait	2	1	0	50.0
los	11	9	33	-218.18

Table 5.11: Viterbi detection rates for a number of query terms.

In their recent paper on initial retrieval experiments on the Cambridge VMR corpus, Jones *et al* generated message representations by a wordspotting and score–thresholding procedure similar to that used here [69]. They observed that retrieval

bomb	fein	john	new	shot	south
cup	home	los	night	siege	wing
death	job	match	peace	snow	work

Table 5.12: Short Query Terms.

performance peaked at a threshold value very near the acoustic threshold at which the wordspotter *accuracy*, defined as the number of correct keyword hits minus the number of false alarms divided by the number of actual keywords present, was maximised. Plotting a similar graph for the Viterbi wordspotting experiment described earlier results in Figure 5.10.

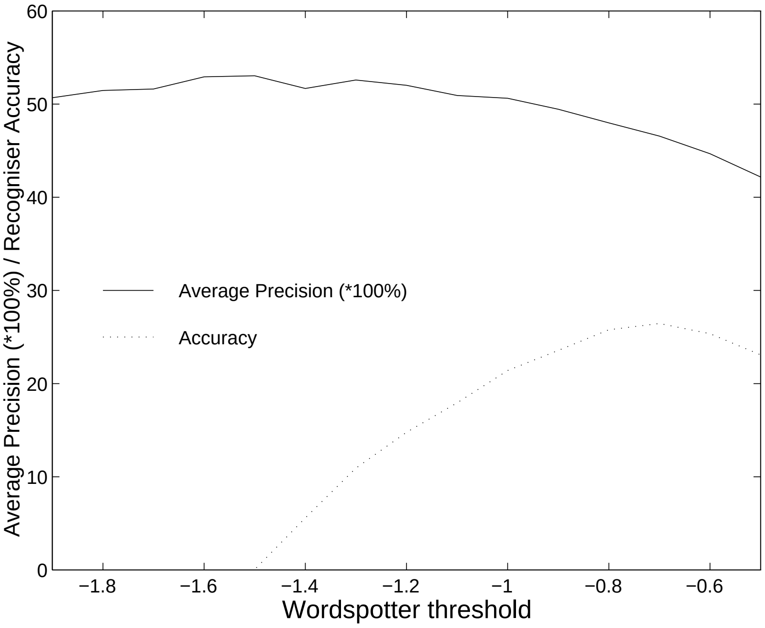


Figure 5.10: Viterbi wordspotter accuracy and average precision of retrieval plotted against the acoustic threshold.

It can be seen from the figure that a similar correlation does not occur for the retrieval systems here. In fact, the average precision of the retrieval system is maximised, in each case, for a much lower acoustic threshold than that which optimises the wordspotter accuracy. Inspection of the correct detection and false alarm rates for each of the individual query terms shows, unsurprisingly, that the short query terms are responsible for a large number of false alarms, disproportionately with the rate of actual occurrence of these terms in the message collection.

Accuracy does not seem to be a useful statistic of wordspotter performance for spoken message retrieval for a number of reasons. Firstly, it is based on a linear relationship between the number of correctly detected terms and the number of false alarms, and so remains unaltered if the lowering of the acoustic threshold on putative term detections admits identical numbers of newly detected terms and false alarms into the set of message representations.

Also, as mentioned in Chapter 4, Kupiec and his co-workers identified a phenomenon they described as *semantic co-occurrence filtering* in their experiments on the use of spoken query terms to index a textual document collection. They observed that ANDing the multiple word hypotheses output by an N-best isolated word recogniser to form a Boolean query resulted in satisfactory retrieval performance. Although the query represented many putative word combinations, only one combination would be sufficiently meaningful that it appeared in the document collection.

A similar phenomenon could account for the optimal retrieval performance occurring at a lower threshold than the most accurate wordspotter output. For example, suppose that a query consists of at least a pair of terms, that some message assessed as relevant to the query contains an occurrence of each term, but that neither term is detected in this message at the threshold maximising the wordspotter accuracy. Suppose also that lowering the acoustic threshold leads to the detection of each of these specific occurrences of the query terms but the inclusion of a larger number of false alarms for each term. Despite the decrease in wordspotter accuracy, retrieval performance may *improve* when the acoustic threshold is reduced, since the distribution of detections of semantically related terms are related whereas the distributions of false alarms of these terms are independent. This means that, in practice, false alarms of these terms are unlikely to occur in the same message.

To illustrate, a simple artificial query, consisting of the terms *northern* and *ireland*, was produced. It was also assumed that every message in the collection in which the phrase "Northern Ireland" occurred was relevant to this query.

For the Viterbi wordspotter, the maximum accuracy obtained on detections of these two words is at a score threshold of -0.7, for which word accuracy is 5%. At this threshold, an average precision of 55.15% is recorded for retrieval with the bp weighting scheme. When the acoustic threshold is lowered to -1.5, word accuracy drops to -80% (corresponding to 14 keyword hits out of 20, at the expense of 30 false alarms) but, owing to the operation of co-occurrence filtering, the average precision under the same retrieval model *increases* to 75.5%. Although this is a fairly simplistic example based on an artificial query, it does illustrate that no obvious correlation between wordspotter accuracy and retrieval performance exists.

In practice, the issue of choosing some acoustic cut–off at which to operate the wordspotter is complicated by user–dominated factors, such as the choice and number of query terms and the precise nature of the relevance assessments, and acoustic factors, most notably the ability of the speech recogniser (be it a small vocabulary, null–grammar wordspotter, a large vocabulary continuous speech recogniser, or somewhere in between) to model the acoustic content of the messages. The only conclusion that can be drawn here is that if the speech recognition component of the message retrieval system is a wordspotter, then too high a threshold on word detections omits a significant amount of information from the message representations, and conversely, too low a value swamps the correctly detected terms with false alarms, to such an extent that even co–occurrence filtering fails.

5.7 Implementational considerations in lattice wordspotting

The implementational factors of spoken message retrieval are equally as important as the retrieval performance obtained. Any useful system incorporating an acoustic query–dependent stage clearly has to complete this stage as quickly as possible. The lattice wordspotter here uses sparse data structures to store the pre–computed lattice edges. For any message, any time during the message and any phone, the list of occurrences of that phone commencing at that time is *instantly* available. Also, the storage of already detected query terms in an inverted file means that, for any query, only terms which have not appeared in a previous query need be detected in the lattices. The average amount of CPU time taken per query, to search the entire set of phone lattices, and output the term detections, is around 8 seconds on a Silicon Graphics Indigo workstation.

It should be pointed out that this figure represents one extreme of the tradeoff between processing speed and the required memory. The sparse arrays required to index the whole $2\frac{1}{2}$ hour message collection occupy 160MB of RAM. Reducing the memory requirements of the lattice wordspotter will necessarily increase the query–dependent lattice search time.

The speed/memory tradeoff could be altered by varying the size of the array of pointers to phone occurrences. In the current system, a pointer is allocated for the occurrence of each phone at each speech frame throughout the message. The mapping from the set of speech frames to the set of pointers is therefore the identity mapping, and the required information is instantly available. To reduce the size of the pointer array, a hash function must map from the index of a speech frame, to the pointer which points to the lattice edges commencing at that speech frame. Since each pointer is now shared by several speech frames, it is necessary to search through the set of lattice edges to find the subset commencing at the speech frame of interest. Varying the size of the pointer array would allow the desired speed/memory tradeoff to be set.

As an example, consider Figure 5.3. In the implemented system, the data corresponding to the two /j/ edges which commence at speech frame 71 is instantly available, since a pointer has been allocated to point exclusively to this data. However, pointers have also been allocated to point to occurrences of /j/ from all other edges, which accounts for the high memory requirement of the system. Sharing the same phone–pointer between x consecutive speech frames, for all phones, would reduce the memory requirement of the system by a factor of x, but it would be necessary to use an efficient search method to locate the exact required subset in the set of pointed–to lattice edges.

5.8 Conclusions

This chapter has described the implementation of a baseline spoken message retrieval system, based on the traditional IR concepts of unconstrained query formulation and subjective message relevance assessment. Phone lattices, as intermediate representations of the message content, were shown to allow for acceptably fast, yet reasonably accurate detection of any query term. Conventional statistical term weighting methods were demonstrated to improve the ability of the systems to retrieve relevant messages. The range of factors affecting the relationship between wordspotter performance and retrieval accuracy was also discussed.

There is some room for improvement in a number of areas. It was seen that the lattice wordspotter is quite poor at detecting short query terms. Since the utility of short terms in retrieval is not in doubt, some effort must be made to improve the accuracy of their detection.

A factor which has not yet been addressed fully is the *term variant* problem. In the text message reference run performed in this chapter, the messages were processed to strip the derivational and inflexional suffixes from each word in order to increase the number of term matches between the text message transcripts and the queries. However, the wordspotters used here only allow for the identification of semantically related but dissimilar terms when the query term is a "phonetic substring" of the message term. They do not, generally speaking, match similar terms where there is a significant acoustic mismatch. It is now appropriate to extend the speech recognition component to include explicit matching between query terms and their acoustic variants in the message collection.

Chapter 6

Improvements to Language and Acoustic Modelling

At the end of chapter 5, it was suggested that a number of improvements should be made to the acoustic modelling in the baseline retrieval system. In particular, it was noted that the lattice wordspotter was not able to detect short query terms reliably. In addition, no effort had been made to identify terms which differed in explicit acoustic form though represented the same underlying concept. In text retrieval this can be accomplished through the use of a stemming algorithm. As will be seen later, although this helpful identification occurs automatically to a limited extent in spoken message retrieval, it does not occur between arbitrary pairs of related terms.

In this chapter, both these areas are addressed by the adoption of what shall be termed a *hybrid* approach to term detection. This approach will involve the use of conventional continuous word recognition techniques to detect, before any message retrieval is performed, occurrences of a some pre–chosen vocabulary of words. When a query is put to the retrieval system, putative occurrences of query terms which are in the pre–chosen vocabulary will be known immediately; the remainder will be detected in the message lattices in accordance with the methods described in the previous chapter.

The hybrid method addresses the two problems outlined in the introductory paragraph as follows. Firstly, so long as the pre–chosen vocabulary contains a significant number of short terms, short term detection will be improved. This is because the Viterbi recognition method, unlike the lattice method, guarantees that term detections do not overlap. In addition, Viterbi continuous word recognition, as opposed to conventional wordspotting, ensures the recognition of each utterance as a string of adjacent words. Even speech that is uninteresting from a retrieval point of view will be modelled well. This should sharply reduce the number of "false alarms" for each term.

Secondly, as long as the preselected vocabulary is sufficiently big and estimated from a suitable source, it should contain a number of acoustic variants of each "underlying" term. This will allow the explicit acoustic modelling of these variants in spoken messages. The use of conventional stemming methods should permit the matching of related but dissimilar terms.

The new hybrid technique should offer improved term detection and enhanced query–message matching. In addition, query–dependent lattice search time is reduced, since a sizeable proportion of the query terms should already have been detected by the word recogniser.

For continuous word recognition to be adopted in message retrieval, it must be assumed that all the spoken messages are from the same language *domain*, which in this experiments presented here,is the domain of news reports. This assumption allows a vocabulary (and optionally a language model) to be adopted, thereby limiting the retrieval system to operation in a single domain. However, it is a general property of text retrieval collections that they consist of items from a single domain, or in the case of the TREC collections, a small number of domains. Similarly, most experimental continuous speech recognisers can only recognise sentences from a particular domain (text from the *Wall Street Journal* newspaper, for example). Therefore, the adoption of the hybrid term detection model should not be seen as a significant break with common practice in either information retrieval or speech recognition.

This chapter describes a number of retrieval experiments based on the adoption of the hybrid term detection method. Firstly, the implementation of the stemming algorithm is briefly discussed. Next, experiments with a 1000–word *no–grammar* model, estimated from the training data, are detailed. Results are presented on a retrieval system employing a more sophisticated approach, with a 3000–word vocabulary estimated from a large corpus of British newspaper text, and a *bigram* language model. Finally, experiments are performed with a hybrid term detection system based on sophisticated *state–clustered* biphone and triphone models.

6.1 Stemming

As was mentioned in chapter 3, the aim of stemming is to improve recall by identifying differing terms with similar meaning. This is based on the underlying assumption that two words with the same word–stem refer to the same underlying concept. The aim of a stemming algorithm can be thought of as the construction of a mapping from the set of all English words into a smaller set of permissible word stems, thereby defining an equivalence relation between words. This is illustrated in Figure 6.1. Where stemming is useful in allowing the identification of related terms, it can, as pointed out earlier, blur the subtle intent of an original phrase. However, the general success of the method makes it a widespread and popular technique in textual information retrieval.

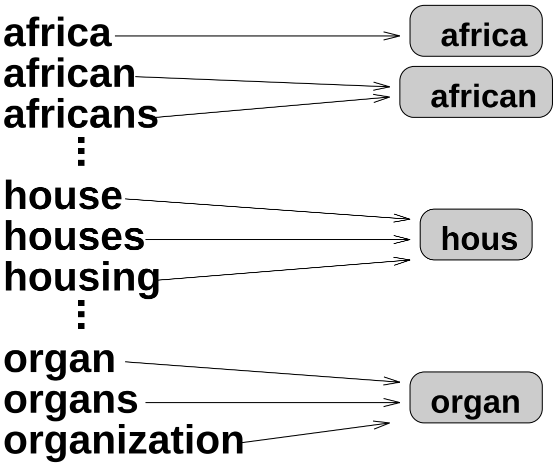


Figure 6.1: Stemming as a mapping from a set of English words to a set of word equivalence classes. Note the overstemming of the words *organ* and *organization*, and the mapping of *africa* and *african* into separate classes.

The computer program used to perform stemming in the experiments presented in this chapter and subsequently is an ANSI C implementation of Porter's algorithm [27]. This algorithm, unlike some, does not depend on a stem dictionary, but is instead simply based on a small list of suffixes and a set of empirical rules under which the removal of suffixes from words may take place. The accuracy of a stemming algorithm can be measured by its ability to approximate a desired "correct" word–stem mapping for a fixed vocabulary of terms. Van Rijsbergen reports typical error rates of 5% for a number of differing approaches to the problem [4].

On the manually obtained transcriptions of 10 randomly chosen news reports from the Radio 4 test collection, a negligible number of errors were produced by the Porter algorithm. These were typically adjectival forms of proper nouns that were not reduced to the same stem as the related noun, for example *africa/african* and *europe/european*. However, as Porter points out, any attempt to improve accuracy by extending the rule–base so that specific words are treated correctly, tends to lead to a drop in accuracy for other words.

In subsequent sections, the phrase *stem–equivalence* will be used to describe the relation between two words w_1 and w_2 that are mapped to the same word stem by the stemming algorithm. In addition, w_1 and w_2 will be said to be *stem–equivalent*.

6.2 No–Grammar Language Modelling

6.2.1 Fixed–Vocabulary Retrieval

The first retrieval experiment employing the new hybrid strategy was based on an extremely naïve approach to language modelling. A vocabulary of 1116 words was selected from the 3030 model training sentences, by ranking words by their frequency and selecting all those occurring at least 8 times. This vocabulary was, on inspection, composed of general task vocabulary (function words and potential query terms, for example *britain*, *controversy*, *injured*, *etc* and words, mostly proper nouns, referring to "on–going" items of news at the time, such as *bosnia* and *palestine*). However, a number of terms were included in this vocabulary which occurred only in individual news stories, such as *bilsthorpe*[1], *croeserw*[2] and *taylforth*[3]. The coverage by this vocabulary of the difficult short query terms was good; it contained 14 of the 18 terms listed in Table 5.12.

Probabilistic language models are typically estimated from large amounts of suitable text data [80]. However, the only text immediately available for the estimation of a language model were the word–level transcriptions of the acoustic model training data. Since this collection was far too small to allow for the construction of a well–trained bigram language model, words and silence were simply assumed to be equally likely, and assigned an occurrence probability of $\frac{1}{1117}$. This probability penalised the insertion of word hypotheses into the recognised sequence. The adjustment of a scaling factor on this probability controlled the rate at which the recogniser inserted or deleted words from the output sequence, in comparison with the exact transcription. A factor of 5.0 was found to offer a suitable tradeoff between word insertions and deletions and this value was fixed for subsequent word recognition experiments.

A words–in–parallel recognition network, illustrated in Figure 6.2, was built by concatenating the monophone models to build the required word models, with respect to the single pronunciation for each word obtained from the phone dictionary.

[1] The scene of a colliery accident.

[2] The scene of a murder.

[3] An actress who brought an unsuccessful libel action against a tabloid newspaper.

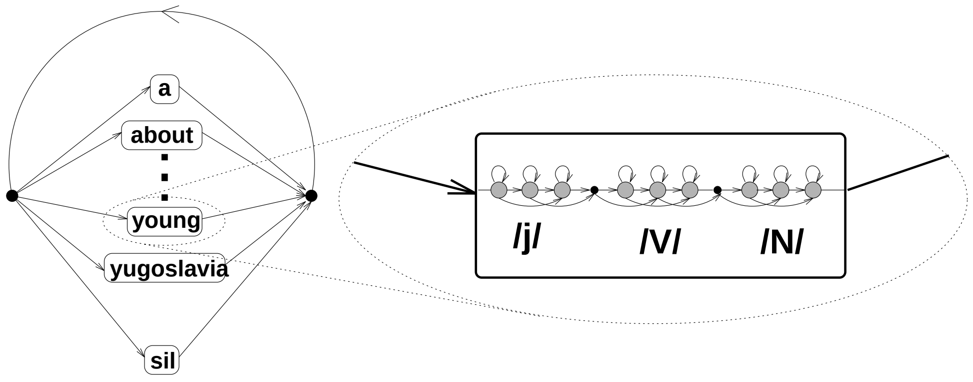


Figure 6.2: The recognition network for the initial word recogniser.

Recognition of the $2\frac{1}{2}$ hour message collection took around 14 hours of CPU time on an SGI Indigo workstation. The word recognition performance on each spoken message was scored with respect to the transcribed content of the message, and in accordance with the HTK alignment procedure [10]. Out–of–vocabulary words were not excluded from the scoring procedure. The correctness figure obtained was 41.58% and the accuracy 26.77%. The output of the recogniser for a sentence from an example spoken message is given in Table 6.1.

```
LAB: the footballer      paul gascoigne has been  in   trouble again
REC: the football  were  paul gas colin has he'll into bonn    began
```

Table 6.1: Word recogniser output for a sample sentence aligned against the sentence transcription.

Several factors contribute to the relatively poor performance of this word recogniser. As the example illustrates, a 1116–word vocabulary is far too small for an speech recognition task featuring a large number of proper nouns, including personal and place names, whose occurrence cannot be predicted. In addition, the lack of a probabilistic language model means that the recognition of a word or phrase can have no influence on the detection of the subsequent word except by fixing a common word boundary.

It can be argued, however, that the use of standard recogniser correctness and accuracy measures to analyse the potential utility of transcriptions for message retrieval is misleading. The function words are certainly not interesting from a retrieval point of view; they are solely useful as part of a language model, of whatever complexity, to constrain the over–recognition of the potential query terms. In addition, since subsequent stemming will allow the matching of certain differing terms, substitutions between stem–equivalent terms are actually immaterial.

Therefore, all the function words in van Rijsbergen's stop list were removed both from the recogniser output and from the exact transcription files against which recogniser output is scored. Rescoring gave a correctness figure of 43.95% and an accuracy of 20.64%. The drop in accuracy indicates that the recogniser has been better at recognising function words than content words; this is not surprising, as almost all the function words appearing in the test data are in the 1116–word vocabulary, in contrast to the content words. Further, stemming the stop–word–free recogniser output and message transcriptions gave rise to correctness of 48.40% and accuracy of 24.94%. The increase in both correctness and accuracy reflects the useful identification of differing words that is performed by stemming. Table 6.2 illustrates the new alignment for the same sample sentence.

```
LAB: footbal paul gascoigne trouble
REC: footbal paul gas colin bonn began
```

Table 6.2: Comparison of stemmed, stop–word–free recogniser output and sentence transcription.

Inspection of the recogniser output revealed that most of the substitutions between stem–equivalent word pairs were those in which the recogniser vocabulary word was a *base* form, *i.e.* the singular of a noun or the uninflected infinitive of a verb, and the word occurring in the message collection was the noun plural, inflected verb, or an adjectival or adverbial form. Example term identifications are given in Table 6.3.

Acoustic substitutions of this kind are quite unsurprising since all the required acoustic evidence for detecting the base word is frequently contained in the occurrence of the inflected word. Only rarely does the recogniser perform a substitution between two differing *inflected* word forms, such as *included/including* or *possible/possibility*. This suggests that in general, it will not always be possible to detect occurrences of a query term in the spoken message collection using an acoustic word model corresponding to an arbitrary stem–equivalent word. The extent to which this problem interferes with retrieval will be determined later.

The first retrieval experiments were performed solely using the word recognition output, after the removal of suffixes and stop words, as the message representations. No query–dependent acoustic or lattice matching took place. 114 of the 206 query terms had at least one stem–equivalent word in the recogniser vocabulary. There

Spoken Word	Recognised as
agree	agreed
america's	americans
championship	championships
chancellor's	chancellor
clinton's	clinton
deaths	death
detective	detectives
economically	economic
eleventh	eleven
girls	girl
hoping	hope
leadership	leader
operation	operations
problem	problems
reporter	report
returned	return
supported	support
terrorists	terrorist
thousands	thousand

Table 6.3: Recogniser substitutions of stem–equivalent word pairs.

were an average of 3.8 terms from the recognition vocabulary in each query, but two queries contained none of these terms. Where no query information was available, the retrieval system assigned every message the same score; here, retrieval precision was $\frac{1}{337}$, or roughly 0.003, for all levels of recall.

From a retrieval point of view, recognition using the 1116–word vocabulary is quite well motivated. This is because the vocabulary consists of words estimated to occur with high or medium frequency in the message collection; the medium–frequency words are the potential indexing terms for the messages, whereas the models of the high–frequency words act as acoustic filler. Moreover, since the continuous word recogniser can be seen as a wordspotter designed to detect content words, with the set of function word models as the garbage model, the usual log likelihood ratio scoring method can be used to rescore the putative detections of each interesting word.

In the first experiment, the message representations were varied by sweeping the usual single threshold over the DNLLR scores. The message representations and queries were all stemmed to allow the matching of acoustically variant terms. The results for np term weighting are displayed in Table 6.4, and illustrated in Figure 6.3.

	Acoustic Keyword Thresholds							
	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4
AvePrec	0.3699	0.3937	0.4172	0.4309	0.4376	0.4418	0.4516	0.4577
Prec@5	0.4500	0.4750	0.5100	0.5350	0.5550	0.5550	0.5600	0.5650
Prec@10	0.3725	0.3925	0.4100	0.4075	0.4150	0.4200	0.4325	0.4325
Prec@20	0.2625	0.2763	0.2888	0.2963	0.3000	0.2950	0.3013	0.3038
	Acoustic Keyword Thresholds							
	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.2
AvePrec	0.4574	0.4680	0.4657	0.4681	0.4714	0.4705	0.4726	0.4789
Prec@5	0.5750	0.5950	0.5900	0.5900	0.5900	0.5750	0.5850	0.5850
Prec@10	0.4350	0.4400	0.4400	0.4375	0.4325	0.4350	0.4375	0.4400
Prec@20	0.3013	0.3050	0.3013	0.3025	0.3038	0.3025	0.3013	0.3050
	Acoustic Keyword Thresholds							
	-2.3	-2.4	-2.5	-2.6	-2.7	-2.8	-2.9	$-\infty$
AvePrec	0.4801	0.4761	0.4754	0.4738	0.4742	0.4741	0.4737	0.4627
Prec@5	0.5850	0.5800	0.5800	0.5750	0.5850	0.5850	0.5750	0.5600
Prec@10	0.4400	0.4400	0.4400	0.4400	0.4375	0.4400	0.4425	0.4400
Prec@20	0.3075	0.3075	0.3075	0.3075	0.3075	0.3050	0.3025	0.2975

Table 6.4: Performance for np–weighting of retrieval on word recogniser output only.

These results compare remarkably well with the results of the Viterbi and lattice wordspotting systems described in chapter 5, especially since retrieval is based on detections of only 114 of the 206 query terms. Performance was good for such a limited set of terms since term detection was now far more certain; the word recognition vocabulary supplied a detailed acoustic model of non–query–term speech that was lacking up to now. The wordspotter Figure of Merit on the *entire* set of query terms, even including those *not* appearing in the 1116–word vocabulary, was 66.38%. This compares with a FOM of 67.79% obtained on the entire set of query terms by the Viterbi wordspotter in the previous chapter. It can be seen that the new approach represented a much improved general model of the unknown speech, since it modelled the "uninteresting" non–query–term speech far better. Alternatively, judging the wordspotting performance of the word recogniser on its ability to detect only the query terms contained in its fixed vocabulary, performance rose to 80.10%. This compares with a FOM value of 68.13%, obtained by the Viterbi wordspotter on the same 114–word subset of the query terms. Also, as in the text case and for the wordspotting experiments, inverse document weighting and term frequency weighting were both found to improve retrieval effectiveness significantly, with the np–weighting scheme

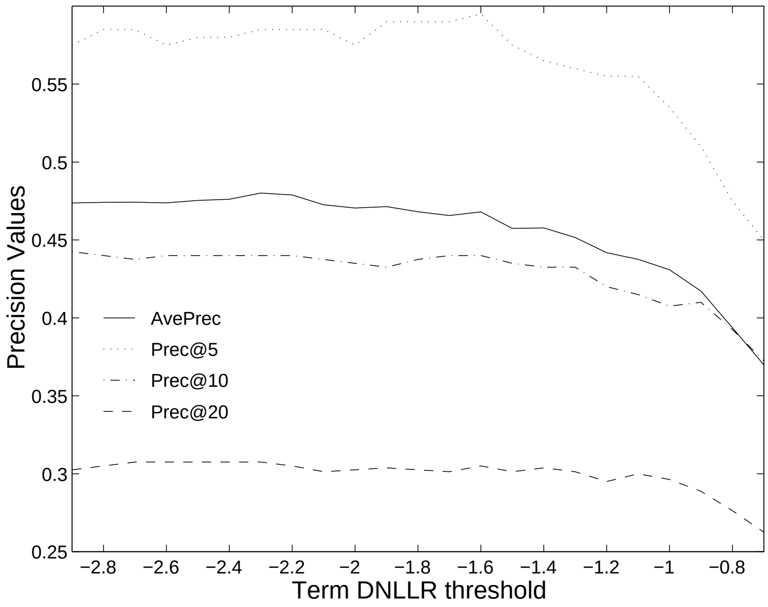


Figure 6.3: Retrieval precisions on word recogniser output only.

typically outperforming the co–ordination level match by between 7 and 8 precision points.

However, the curve relating retrieval effectiveness to the acoustic threshold did not behave the same way as it did for the wordspotting experiments. Using a word recogniser, instead of a wordspotter, to account for the spoken messages considerably reduced the number of term false alarms while largely retaining the correct term detections. This means that the term rejection threshold could be set much lower than before, as the lower–scoring correct term detections were now more "useful"; their incorporation into the message representation no longer meant having to include a large number of misleading false alarms. It can be seen from the graph that the word recognition system reached its peak at a much lower threshold, and that performance tailed away far more slowly than for the wordspotting–based retrieval; in fact, the difference between the precision at the optimal score threshold and the precision obtained without any threshold at all, was less than two precision points. Ratio scoring and thresholding was therefore much less effective for the word recogniser than for the wordspotter.

6.2.2 Retrieval on Full Queries

In the next experiment, retrieval was performed on the full content of each query. Lattice wordspotter term detections were incorporated into the message representations for the query terms not included in the word recogniser vocabulary. The core assumption made in this experiment was that a stem–representative, in the word recogniser vocabulary, of a required query term, would also act as an *acoustic* representative for that term, and that the vocabulary word model would match occurrences of the required term. This assumption is, of course, a simplification, since the empirically derived rules on which the stemming algorithm is based cannot guarantee the identification of related terms. In addition, stem–identical terms can turn out to be too acoustically dissimilar for the required matching to take place. In practice, for a query term Q, one of the following cases holds.

1. Q is contained in the recogniser vocabulary. The word model of Q is useful in detecting occurrences of term Q in the spoken messages, including possibly some variant forms.

2. Q is stem–equivalent, but not acoustically identical, to at least one recogniser vocabulary word W. Occurrences of term Q may be detected using the model of word W so long as Q and W are not too dissimilar (for example, Q = "accidents", W = "accident"). However, the acoustic forms of Q and W may stretch the assumption — for example, if Q="negotiations" and W="negotiator". One approach to this problem would be to enrich the word recogniser vocabulary so that it contained a greater number of related words for each word stem, thereby increasing the chance that the word recogniser has a suitable model for query term Q.

3. Q is not stem–equivalent to any recogniser vocabulary word, and consequently must be detected in the lattice. Q might be highly related and acoustically similar to some vocabulary word W (for example, if Q="african" and W="africa"); however, if the stemming method cannot identify these terms, the available term detection information for word W cannot be used. This also happens for acoustically dissimilar pairs terms of English, such as "creating" and "creation", which are related but not stem–equivalent under the Porter algorithm. This, however, is as much a problem for text–based retrieval as for speech, as is the fact that the stemming algorithm often makes identifications between terms that are not always useful in enhancing recall, such as "capital" and "capitalism".

A drawback of the hybrid approach adopted here was that lattice wordspotting was carried out over the whole length of each spoken message, irrespective of the accuracy of word recognition. This approach seems quite wasteful, since all the speech in the message collection must be recognised twice. An alternative approach would involve setting a rejection threshold on the ratio scores output by the word recogniser. This would identify areas of the unknown speech which matched poorly against the recogniser vocabulary; only these areas need then be searched by the lattice wordspotter for those query terms not included in the recogniser vocabulary. However, as was seen in the *africa* example above, the stemming method cannot guarantee that a word detected correctly by the continuous word recogniser will be useful in retrieval. Therefore, query terms were detected in the lattice regardless of the word recogniser output.

In the first hybrid experiment, the ratio score threshold on the word recogniser output was fixed at -2.3, the value for which average precision on the earlier, fixed–vocabulary experiment was maximised. 92 query terms were detected in the lattices of degree 8, and the term hypotheses thresholded at the usual range of values and incorporated into the inverted file. As can be seen from Table 6.5 and Figure 6.4, np–weighted retrieval was far more effective than for any of the previous systems. The best average precision here is 0.5793, which corresponds to an effectiveness ratio of 82.31%, and is to be found at roughly the same threshold as the best lattice wordspotter performance in chapter 5. Between the thresholds of -0.5 and -1.9, the average precision varies between 0.5572 and 0.5793, a variation of 2.2 precision points. In comparison, the average precisions vary over 7.5 precision points in the full 8–lattice system in Subsection 5.5.3. The fixing of the word recogniser output has clearly reduced the dependence of retrieval performance on the setting of the lattice wordspotter threshold.

The hybrid system has some potential for improvement. For a start, it is based on the central assumption that stem–equivalent terms are acoustically confusable, which is by no means always true. In addition, lattice wordspotting has so far been performed on single, arbitrarily inflected query terms, instead of more general, word stem–like units, which could potentially match a wider range of related words. It seems sensible to improve the lattice wordspotting to allow for the detection of multiple term variants, instead of the single term specified in the query. For example, one of the lattice query terms is "employment"; it would notionally be useful to match this term against related terms such as "employ", "employer" and so on. This approach could take one of two forms;

	Lattice Wordspotter Threshold							
	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2
AvePrec	0.5575	0.5659	0.5669	0.5648	0.5668	0.5653	0.5606	0.5642
Prec@5	0.6450	0.6450	0.6400	0.6450	0.6450	0.6500	0.6400	0.6350
Prec@10	0.5125	0.5100	0.5100	0.5100	0.5050	0.5050	0.5050	0.5100
Prec@20	0.3525	0.3575	0.3600	0.3550	0.3538	0.3488	0.3487	0.3512

	Lattice Wordspotter Threshold							
	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	$-\infty$
AvePrec	0.5713	0.5760	0.5767	0.5793	0.5693	0.5663	0.5572	0.5099
Prec@5	0.6350	0.6400	0.6400	0.6500	0.6400	0.6300	0.6350	0.6000
Prec@10	0.5200	0.5200	0.5125	0.5075	0.5050	0.5050	0.5025	0.4700
Prec@20	0.3550	0.3563	0.3588	0.3625	0.3600	0.3575	0.3562	0.3300

Table 6.5: np–weighted retrieval performance for hybrid system with 1116–word recogniser threshold fixed and wordspotter threshold varied.

- Reducing each query term to a stem–like unit, obtaining a phonetic form of this unit, and then detecting it in the message lattices, or
- Replacing each query term by the members of its stem–equivalence class, taken from a dictionary or a similar source, such as the list of training data words, and searching for each such word.

6.2.3 Improvements to lattice term detection

The first method of improving term detection in the lattices was implemented by manually selecting 12 specific lattice query terms for which it seemed likely, from visual inspection of the message collection, that acoustic reduction would improve the detection of query term variants. These query terms are given in Table 6.6. This reduction was performed manually, since this would provide an upper limit on the retrieval improvement that could be obtained via an automatic approach to the problem. Such an automatic approach might be to extract from the members of the stem–equivalence class of some query term, the maximal "phonetic substring" common to each.

The manual term reduction was also able to circumvent stemming errors. For example, whereas the Porter algorithm is unable to identify the noun "creation" with any of the inflected forms of the verb "create", these words can be identified acoustically by detecting the manually–determined stem "crea", although it is of course known that the shorter an acoustic unit, the more unreliable its detection by the lattice wordspotter.

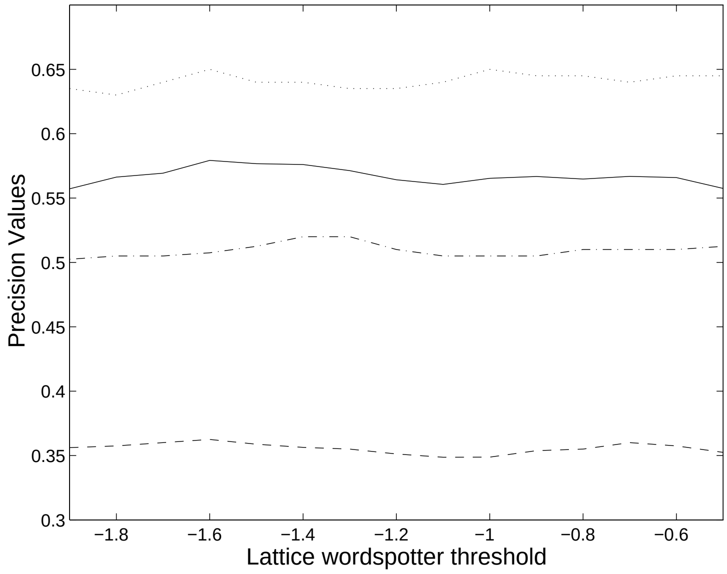


Figure 6.4: Retrieval precisions on first *hybrid* system.

This technique was found to improve retrieval effectiveness, although only by just over a single precision point. The best average precision for np-weighting was now 0.5905, compared to the earlier figure of 0.5793. This was despite a large increase in the wordspotter Figure of Merit for these terms, from 29.68% to 68.92%. For the message collection and query set on which experiments have been performed here, it would seem that the number of terms (an average of 6.73) in each query already allows for a large enough number of term matches, so that an improvement in word detection for an arbitrary subset of the query terms does not necessarily increase retrieval effectiveness significantly. Since, in this experiment, the query terms were selected specifically because they seemed likely to benefit from this approach, the results represent the upper limit of performance improvement. Consequently, it seems unlikely that a totally automatic approach to this problem could improve retrieval effectiveness. This is not, however, to rule out the potential of this approach in a different spoken message retrieval task, especially one in which the queries are poorer, and where it would be vital to increase the number of query term matches.

The second method of acoustic query expansion looks as if it might improve retrieval performance by a greater degree than the acoustic stem approach, since it is based on the modelling of longer acoustic units. However, in practice, it can only improve the modelling where the manually derived acoustic stem of a query term is not a word itself. For example, in the earlier approach, the query term *crashes* was

Query Term	Acoustic Stem
activities	activity
crashes	crash
creation	*crea*
croatia	*croa*
disasters	disaster
earthquakes	earthquake
elections	election
employment	employ
fundholding	fundhold
goals	goal
management	manage
proceeding	proceed
railways	railway
redundancies	*redundan*
trains	train
violence	*violen*
wards	ward

Table 6.6: Query Terms and their corresponding "acoustic stems". Italics indicate acoustic stems that are not words themselves.

reduced to the stem *crash*, which then matched on occurrences of *crashed* and *crashing* in the spoken message collection. However, lattice wordspotting on all three term variants will not improve detection, since false alarms for *crash* will obviously be generated at the same rate as for the earlier "acoustic stem" method. The new approach may only help for an original query term such as *violence* which was reduced by the earlier method to a non–word acoustic unit, *violen*. This can now be expanded to the two terms *violent* and *violence*. Searching on these may deliver improved detection rates in comparison to the "acoustic stem" approach. This experiment was performed, again manually producing revised queries, so the results can again be thought of as the upper limit of performance to be gained by this technique. However, the technique made almost no difference to the wordspotter Figure of Merit or retrieval effectiveness, compared to the "acoustic stemming" method.

It is important to establish the extent to which the assumption about acoustic confusability of stem–equivalent terms fails. Examination of the 1116–word vocabulary and the set of query terms shows that there were no inflected query terms for which the only stem–equivalent word in the recogniser vocabulary was a differing inflected form. This is largely due to the selection of the recogniser vocabulary by a frequency criterion, so that only rarely did an inflected word appear in the vocabulary without the corresponding "base" form also being present. It seems that the general inability, mentioned earlier, of Viterbi word recognition to make acoustic confusions between *two inflected terms* was not a problem in this task.

On the other hand, as a result of the acoustic confusability assumption, an acoustic model of an inflected word was occasionally required to match a stem–equivalent, uninflected word occurring in the message collection. Inspection of the query term set and the spoken message transcriptions indicated that here, there were only 4 such query terms out of a total of 206 terms. Table 6.7 lists these terms and shows the ability of the word recogniser to perform the word substitution required under the main assumption of the hybrid term detection system.

Spoken Term	#Occs	Acoustic Word Model	#Hits
exchange	7	exchanges	3
magistrate	1	magistrates	1
sport	2	sports	1
tory	1	tories	1

Table 6.7: The ability of the word recogniser to perform some "difficult" term substitutions.

It can be seen that about half the required confusions were performed. Unsurprisingly, the greatest problem occurred for the most significant term inflection, namely *exchanges/exchange*. This problem could be solved either by relaxing the core assumption, thereby allowing the detection of the uninflected query term using the lattice wordspotter, or by extending the vocabulary of the word recogniser.

In conclusion, the first attempt at a hybrid term detection method provided an improved approach to spoken message retrieval. Although the central assumption of acoustic confusability of stem–equivalent terms has its drawbacks, it is for the most part a sensible solution to the problem. Since the hybrid method reduces the burden on the lattice wordspotter, it decreases the query–dependent lattice search time, from eight seconds down to around four seconds per query on the SGI Indigo workstation.

Within the hybrid model, a suitable approach to improving retrieval effectiveness again would be to adopt a bigger, better–estimated language model, thereby extending the core vocabulary and further relieving the burden on the lattice wordspotter.

6.3 Bigram Language Modelling

In a speech recognition task for which a fixed recognition vocabulary is assumed, a well–estimated probabilistic *n–gram*[4] language model typically improves recogniser accuracy. This is because the identities of the previous $n - 1$ words in the recognition path are always taken into consideration, thereby constraining the output sequence. Arguably the best trade–off between recogniser performance and language model trainability and size is given by a 2–gram (*bigram*) language model. In addition, a bigram language model can be incorporated quite easily into a Viterbi recogniser based on the Token Passing Paradigm. It is simply necessary to add an appropriately scaled log probability, $s.\log(\Pr(w_y|w_x))$ to the path score of a token exiting word w_x and being propagated into the model for word w_y. Unfortunately, since the acoustic training data collection, consists only of 58849 occurrences of 6305 separate words, it is not sufficiently big to allow for the generation of a well–trained bigram model for any meaningful vocabulary size. A much larger source of suitable data must be found.

It was thought that, if available in sufficient quantities, the text of a British "quality" broadsheet newspaper would be a suitable source of data for the construction of a bigram language model for the spoken news reports. During the orthographic labelling of acoustic training and test data, the *Independent* newspaper was regularly used to provide spellings of foreign proper nouns, and it was also observed that the journalistic style of the newspaper appeared very similar to that of the Radio 4 news reports. On some occasions, the wording of a spoken news report was identical to that appearing the following day in the newspaper, undoubtedly since the story text was provided to both newsrooms by the same agency. A large, machine–readable source of such text was therefore thought likely to be useful in the production of a language model for the word recogniser. Luckily, such a source of data exists, since the text of the *Independent* is now available on CD–ROM.

The entire editorial content (that is, articles, as opposed to adverts, photographic material *etc*) of the *Independent* between August 1989 and December 1990 was obtained in individual headered article form on a single CD–ROM [81, 82]. The article headers were used to partition the collection into news and "feature" subsets, and the feature articles discarded, leaving a collection of around 77,000 articles totalling approximately $30\frac{1}{2}$ million words. The data files were then manipulated into a form suitable for processing by the HTK labelling tools. Words were ranked by their frequency of occurrence in the data collection and all those occurring at least a thousand

[4] an n–gram is simply an arbitrary sequence of n words.

times were selected for inclusion in the new language model, yielding a total of 3043 words. This set was chosen as the new recognition vocabulary. The data were further processed by the replacement of each non–vocabulary word with a single out–of–vocabulary (abbreviated to *OOV*) label.

A bigram language model was trained from the data using Katz's "back–off" method [83]. This procedure generates improved estimates of the probability of unobserved or infrequent word n–grams by *redistributing* some of the probability "mass" associated with more frequent n–grams. No attempt was made to construct an acoustic model intended to match all non–vocabulary speech, and all transitions to the *OOV* label were ignored. In his wordspotting experiments with the Credit Card subset of the Switchboard collection, Weintraub found that a wordspotter that combined an acoustic *OOV* model with a large vocabulary and a bigram language model, had a lower Figure of Merit than a system without such an *OOV* model, since the *OOV* model had tended to match on poor acoustic realisations of keywords.

In a word recogniser with an n–gram language model, the recognised word score is a combination of the acoustic log likelihood and the scaled log bigram probability. It was unclear whether the usual technique of ratio scoring and thresholding could be applied in this case. Successful ratio scoring could certainly not be achieved by subtracting the language model score and obtaining a ratio of acoustic scores; with the use of a probabilistic language model, a word may be correctly recognised *despite* a poor acoustic realisation, if its occurrence is highly probable given the previously hypothesized word or phrase. Since ratio scoring could not be relied on here, there was no simple method of rejecting word substitutions that occur in the message representations output by the word recogniser. It therefore seemed reasonable to attempt to favour "harmless" substitutions, for example, the recognition of non–vocabulary words as known vocabulary words which were guaranteed not to interfere with the retrieval process.

The contents of a "stop list" is a suitable set of such words. Since they have no intrinsic content by themselves, they do not appear in the queries. In addition, the function words are, on average, quite short, and sufficiently acoustically dissimilar to suggest they might provide a more accurate acoustic filler for the *OOV* words than a more usual single or parallel phoneme model. A second bigram was now estimated using the back–off method, but now *redistributing* the probability "mass" of bigrams of the form $\{w_i, OOV\}$ amongst the set of bigrams $\{w, w_f\}$, with w_f varying over the set of function words. The probability mass redistribution was based on the relative frequencies of the function words, so that a more frequent function word was allocated a greater proportion of the probability mass.

Recognition was performed using the new 3043 word vocabulary and each of the two bigram models. The bigram probability scaling factor was left at the earlier value of 5.0. The word recogniser correctness and accuracy scores using the first bigram were 61.47% and 51.33% respectively. Removing function words and stemming the remaining output, these figures rose to 65.40% and 55.21% respectively. For the bigram with the *OOV*–redistribution, the correctness and accuracy were 61.04% and 53.64%, rising to 64.88% and 57.43% after stop word removal and stemming. Thus, *OOV*–redistribution increased recogniser accuracy by over 2% at the cost of a small drop in correctness. However, it remained to be seen which set of message representations would give rise to the better retrieval performance. Table 6.8 illustrates the output for a couple of sentences of recognisers based on both bigram models.

```
LAB:    in  bosnia      the muslim parliament...
REC-1:  in  bath near  over muslim parliament...
REC-2:  him by us near over muslim parliament...
```

```
LAB:   the footballer    paul gascoigne   has been in trouble again
REC-1: the football were paul gas calling has been in trouble again
REC-2: the football were paul gas calling has been in trouble again
```

Table 6.8: Comparison of sentence transcriptions with recogniser output for each of the bigram models. REC–1 is the output of the recogniser with the ordinary bigram, REC–2 with the *OOV*–redistribution bigram. The recogniser outputs are actually identical for the 2nd sentence.

A number of differing retrieval experiments were performed using the new sets of word recogniser output obtained using the two bigrams. In the first experiment, retrieval was performed using only the word recogniser output, to determine the effect of the usual ratio–scoring method, and to find out which of the two bigram language models was responsible for the better retrieval performance. Table 6.9 illustrates the np–weighted average precision values obtained for a number of differing score thresholds, and without any threshold altogether.

It can be seen from this table that the usual ratio–scoring method did indeed fail to exhibit its usual effect. The method was no longer appropriate, since no meaningful comparison could be made between the combined acoustic and language score now being output by the Viterbi recogniser and the acoustic–only normalisation score. It can also be seen that the recogniser with the *OOV*–redistribution language model produced marginally poorer message representations than the original bigram model. As

Threshold	Ordinary bigram	*OOV*–redistribution bigram
-2.0	0.4387	0.4423
-3.0	0.4856	0.4912
-4.0	0.5034	0.4979
-5.0	0.5075	0.4984
-6.0	0.5044	0.5017
-7.0	0.5028	0.5022
-8.0	0.5048	0.5010
-9.0	0.5051	0.5016
$-\infty$	0.5078	0.5029

Table 6.9: Average Precisions at a selection of thresholds for both bigram model–constrained recognisers.

in chapter 5, it is more important to detect a slightly larger number of correct query terms than a smaller number of more accurate hits. The peak average precision obtained from these experiments was 0.5078 for the np–weighting scheme. In comparison, the average precision obtained on the 1116–word, no–grammar experiment was 0.4801, at a ratio score threshold of -2.3.

At first it seems counter–intuitive that the use of the much larger and hopefully more accurate language model should lead to such a small improvement in retrieval effectiveness when retrieval is performed only using the output of the word recognisers. After all, the larger vocabulary has far better coverage of the set of query terms — it contains 152 out of the 206 terms, compared to 114 terms for the earlier vocabulary. The reason for this result is simply that the language models come from two differing sources. The 1116–word vocabulary was estimated directly from the training data and therefore contained, in addition to the general vocabulary of news reporting, a number of the discriminating proper nouns that occurred in "on–going" stories during the collection of both acoustic training and message test data. So, while the bigram language model was a far better *general* model of the language of news reports, it lacked specifically the more up–to–date terms occurring in the previous model. Inspection of the sets of query terms detected in the lattice in each experiment showed that for the 1116–word vocabulary, it was necessary to search for some general task vocabulary, such as *disaster*, *employment*, and *parliament*. In contrast, all these words were contained in the 3034–word newspaper vocabulary, and lattice wordspotting was performed mostly for proper nouns that were present in the 1116–word vocabulary, such as *bosnia*, *geneva* and *palestine*.

Retrieval results were finally obtained on the full set of query terms by the usual method of incorporating lattice term detections for the remaining terms. The output of the word recogniser with the first bigram model was combined with the output of the lattice wordspotter for the 54 remaining terms, which were detected at a Figure of Merit of 62.36%. Although the word recogniser output was unthresholded, the wordspotter term detections were subject to a varying score threshold.

	Lattice Wordspotter Threshold															
	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.2	$-\infty$
AvePrec	0.5865	0.5855	0.5879	0.5880	0.5880	0.5916	0.5928	0.5931	0.5949	0.5946	0.5955	0.5923	0.5930	0.5893	0.5877	0.5699
Prec@5	0.6350	0.6250	0.6450	0.6450	0.6450	0.6500	0.6500	0.6550	0.6600	0.6550	0.6550	0.655	0.6550	0.6500	0.6500	0.6200
Prec@10	0.5300	0.5300	0.5275	0.5250	0.5250	0.5250	0.5250	0.5300	0.5300	0.5300	0.5300	0.5300	0.5325	0.5300	0.5300	0.5200
Prec@20	0.3675	0.3650	0.3650	0.3650	0.3638	0.3650	0.3663	0.3688	0.3688	0.3675	0.3688	0.3675	0.3675	0.3675	0.3688	0.3650

Table 6.10: Retrieval performance for hybrid system with 3043-word recogniser threshold fixed and wordspotter threshold varied.

The best hybrid performance with a bigram language model can be seen from Table 6.10 to be 0.5955, which corresponds to an effectiveness ratio of 84.61%. Interestingly, this was only $1\frac{1}{2}$ precision points better than the best no–grammar hybrid performance, which was 0.5793. It can be seen quite clearly from Figure 6.5 that the larger, bigram–model–constrained recogniser has flattened the average precision curve even further, reducing the system's dependence on a threshold value that is not easily to estimate. It has also allowed another reduction in query–dependent search time, now down to 3.2 seconds on the Indigo workstation. However, it should not be forgotten that the bigram language model is a far better general model of messages from the general news domain than the no–grammar vocabulary estimated from the training data, and that it would be just as useful in the production of message representations on a totally new spoken message corpus.

6.4 Improved Acoustic Modelling

The experiments described so far have been based on the acoustic modelling capability of the set of 47 monophone Hidden Markov Models, each of which modelled the set

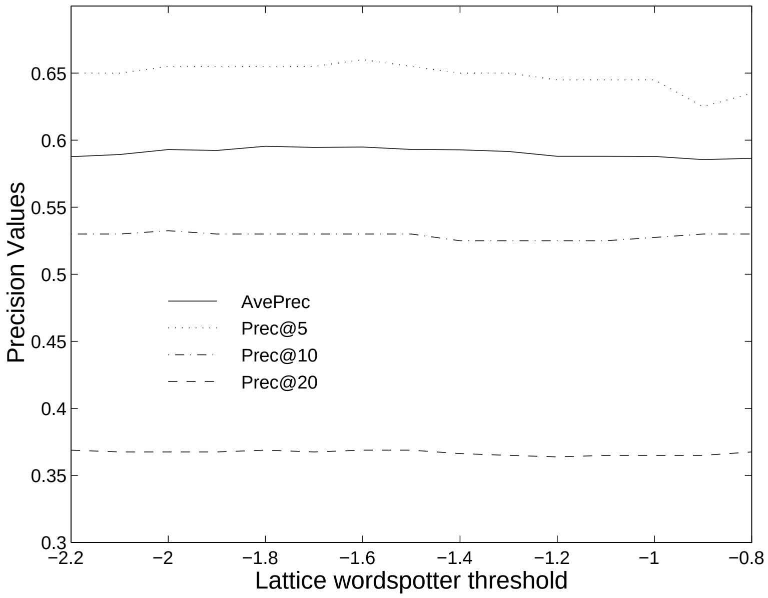


Figure 6.5: Retrieval precision on second hybrid system.

of acoustic data observed for each phone in the training set as a mixture of 12 multivariate Gaussian distributions. While such models are relatively simple to train, they are not sufficiently discriminating for most speech recognition applications, since each phone model is in practice trained on phone occurrences in many differing acoustic contexts and with different allophonic realisations. Monophones represent one extreme of the tradeoff between ease of model training and subsequent performance.

Recognition accuracy can be improved by the use of separate models for a single phone according to its acoustic context. Models for a phone which take into account the identity of the *left* (previous) and *right* (subsequent) phones are known as *triphone* models. Models dependent on only one of these contexts are called *left* or *right biphones*, as appropriate. Model training is the central problem of so–called *context dependent* phone modelling. Any corpus of acoustic training data, labelled in terms of n distinct phones, is extremely unlikely to contain enough occurrences of the n^2 left or right biphones, or even the n^3 triphones, needed to train the entire population of context–dependent models that might be required in a subsequent recognition task. Therefore, a number of techniques have recently been proposed for the *clustering* of model parameters, in order to generate well–trained populations of context–dependent models [9, 84, 65].

This section describes the creation of sets of triphone and biphone models for the spoken message retrieval task using a decision–tree method for the clustering of similar phonetic contexts [85]. These models are then used in word recognition and lattice generation, and message retrieval performance with the output of these recognisers is investigated, to yield the final results obtained by the hybrid method.

6.4.1 Decision Tree–based Model Clustering

The training of context–dependent phone models, involves the sharing of model parameters over a number of models, in order to make best use of limited training data. "Traditional" approaches to the production of robustly–trained context–dependent phone models, for both continuous speech recognition and wordspotting, have tended to be *data–driven*. They have typically involved the sharing of model parameters across differing acoustic contexts, whether by the interpolation of mixture Gaussian weights, or the clustering of similar models [9, 86, 65].

The general shortcomings of data–driven techniques have been commented on before [85]; in particular, model–based interpolative approaches offer no satisfactory way of generating models for triphones which are *unseen* in the training data. This is a crucial problem here, because the hybrid detection model adopted at the end of chapter 6 was based on a language model and acoustic models estimated from unrelated sources of data; if the acoustic modelling in this system is to be improved, a large number of models for unseen triphones will be required.

More recently, linguistic *knowledge* has been exploited to generate clusterings of similar acoustic contexts [84, 60, 85]. These methods have been based on the use of Yes/No questions about phone acoustic contexts to construct binary *decision trees* which correspond to linguistically well–motivated clusters of acoustic contexts. Each "leaf" node of a decision tree corresponds to a cluster of triphones which are *tied* together in subsequent model re–estimation, thereby pooling the training occurrences of these triphones. Although phonetic decision trees have largely been used to cluster entire phone models, a method of clustering model *states* was recently proposed by Young *et al* [85]. This method allows left and right acoustic contexts to be treated separately, which avoids the potential clustering of unlike acoustic contexts, and also provides a simple method for synthesizing entirely new models for triphones unseen in the training data. Young *et al* illustrated the power of this approach in experiments on the annual ARPA speech recognition evaluations on the Wall Street Journal corpus [85, 80].

6.4.2 Training of new Acoustic Models

The training of new acoustic models for both parts of the hybrid recognition component of the spoken message retrieval system proceeded as follows. A set of single mixture Gaussian monophone models was cloned to generate initial estimates of models for the 6607 distinct triphones[5] observed in the acoustic training data. The transition matrices of the initial triphones were "tied" across all differing acoustic contexts of the same "base" phone. This means that the cloned copies of a monophone transition matrix were replaced by one single matrix shared across all acoustic contexts of the base phone. This was allowable since there is typically little durational variation to model across triphones.

Two cycles of embedded Baum–Welch re–estimation were performed and the state occupancy likelihoods stored for use by the decision–tree building program. The set of linguistic questions used in Young *et al*'s experiments was obtained and edited to reflect the different set of phone label names being used in the spoken message retrieval experiments [87]. Phonetic decision trees were created for each of the 44 base phones and used to cluster the set of triphone states. The members of each state cluster were then assigned a "tied–state" single Gaussian output distribution.

The clustering generated a total of 1765 distinct tied–states. The list of triphones and biphones required for the 3043–word vocabulary was obtained and the entire set of models created. This resulted in a total population of 6857 triphones, of which 3699 were distinct. The number of Gaussian mixture components in each tied–state was incremented to 12 by a procedure involving mixture–splitting and embedded re–estimation, similar to that described in chapter 5 to generate the monophone set. In message retrieval experiments, however, the set of 8 Gaussian mixture models were used, since these were found to give marginally better recognition performance. Figure 6.6 illustrates the decision tree clustering and model synthesis process.

A similar procedure was used to generate improved models for phone lattice generation. Triphone/biphone populations trained for word recognition are unsuitable for word–independent phone recognition for a number of reasons. Firstly, unconstrained triphone recognition (*i.e.* using a phones–in–parallel recognition network) on a number of utterances from the message collection performed surprisingly poorly, and took an impractically long time, owing to the huge size of the recognition network. Secondly, there was no potential to speed up the matching process and improve phone accuracy by forcing the acoustic contexts of phones in the output transcriptions to match (*i.e.* forcing the hypothesis of a phone in right context y to be followed by the

[5] including right and left biphones, for the beginning and end respectively of each word.

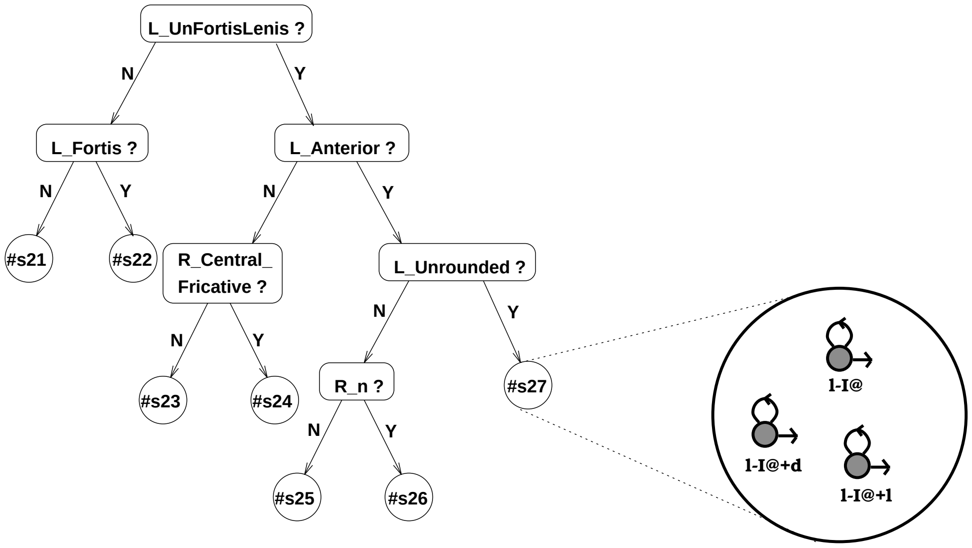


Figure 6.6: A sample decision tree, for state 2 of the vowel I@. The set of questions has been selected and the tree grown to make optimal use of the training data. The circles represent the clustered states that result. Simply "dropping" an unseen triphone down the corresponding set of trees, by asking the relevant questions about its acoustic contexts, allows for the synthesis of a model for that triphone.

hypothesis of a biphone or triphone of phone y). This is because triphone populations are typically selected in order to model some *specific* vocabulary, and forcing acoustic contexts to match would cause a recognition error every time the unknown speech contained a triphone for which no model existed. Since the thinking behind the lattice wordspotting is to offer some method of detecting words not occurring in the vocabulary of the word recogniser, it does not seem sensible to restrict the modelling of these out–of–vocabulary words in such a manner.

It was decided to generate the entire set of 1936 *right biphones*[6] and 44 word–final monophones, using the decision tree state–clustering method and the same set of linguistic questions as was used in triphone state–clustering. (A population of biphones could in theory have been synthesized using the existing phonetic decision trees. However, this would not have been an optimal use of the training data, as these trees were built using triphone and biphone data, where biphones only occurred at the word boundaries. Therefore, only a fraction of the clustered states would be used to generate the biphones.) The existing set of single mixture Gaussian

[6] Left biphones would have been an equally valid choice — the point is that the models were dependent on only one context.

monophone models were cloned to generate initial models for the biphones using the same method as for the triphones. After two cycles of biphone re–estimation, using newly–generated biphone label files, state clustering generated 1060 distinct tied–states. The addition of unseen biphones resulted in a set of 1980 right biphones and monophones, of which 711 models were distinct. These models were subject to the same embedded re–estimation and mixture–splitting procedure as the triphones until a set of 8–mixture models was produced.

6.4.3 Retrieval experiments

State–clustered model accuracy

The two new sets of models were now used to recognise the contents of the spoken message collection. Word recognition was performed using each of the two backed–off bigram language models described in chapter 6, and the same grammar scale factor of 5.0. The right biphones were used to generate lattices of degree 6 using a recognition network consisting of a biphone loop in which acoustic contexts were forced to match. The lattices were postprocessed by the removal of acoustic contexts from the output biphone labels and the removal of duplicate edges. This made the biphone lattices more *compact* than the monophone lattices. Where the identity of a phone in the unknown speech was very certain, the biphone lattice would contain 6 edges with identical start and end times, labelled as hypotheses of that phone, and differing only in the acoustic context of the phone hypothesis. Context–stripping and duplicate edge removal would result in one edge, thereby constraining the set of phone sequences detectable at this point.

Table 6.11 shows a comparison of the performance of the triphone–based 3043–word recogniser against that of the monophone system. It can be seen that the recogniser correctness figures have consistently increased after the switch to triphones, but the triphone output only seems marginally more accurate than the monophone output after the removal of stop words. This is because the acoustic scores output by the recogniser are now uniformly higher, and so the fixed grammar scale factor represents a different tradeoff between word insertions and deletions. At the scale factor of 5.0, the triphone–based system inserts far fewer short function words than the monophone system, because the language model makes a proportionally greater contribution to the combined path scores in the triphone–based recogniser than in the monophone–based recogniser. After the removal of function words from the triphone–recognition output, unwanted insertions of content words remain. This decreases the accuracy of the message representations.

Models	Monophone		Triphone	
	Corr.	Acc.	Corr.	Acc.
Raw output	63.43	37.40	71.17	43.63
After removal of stop words	61.47	51.33	66.24	52.59
After stemming	65.40	55.20	70.62	56.92

Table 6.11: Recogniser correctness and accuracy for the monophone and triphone recognition systems and the ordinary bigram language model. The *OOV–redistribution* bigram systems behave in similar fashion.

The grammar scale factor could of course be adjusted to vary the insertion/deletion tradeoff at which the new recogniser operates. Such an adjustment would probably not benefit retrieval, since increasing word accuracy would necessarily decrease word correctness; experimental results in Sections 5.6 and 6.3 indicated that spoken message retrieval effectiveness increased when more query term detections were available, albeit at a lower accuracy.

The accuracy of the right biphone models was tested by performing Viterbi phone recognition using a biphones–in–parallel grammar in which the acoustic contexts of biphones were forced to match. Performance was measured by stripping the biphone labels output by the recogniser down to the corresponding monophones and comparing with the known phone sequences for the test data. Phone correctness and accuracy were now 73.57% and 70.02%, compared to their earlier figures of 67.92% and 65.29%.

Retrieval results

Retrieval was initially performed on the 40 queries using only the word recogniser output, after stemming and the removal of stop words, as the message representations. No score threshold was applied. With np–weighting, the average precision for the ordinary bigram system was 0.5378, and 0.5382 for the redistributive bigram. These figures both show a 3 point improvement compared to the results of the corresponding monophone experiment in Section 6.3, in which the average precisions were 0.5078 and 0.5029 respectively. There now seems to be little difference in retrieval performance between the systems involving the use of the two different bigram language models. Adjustment of the grammar scale factor to reflect the new level of the acoustic scores would probably separate the two systems, but this was not done here.

As before, 52 query terms were detected using the lattice wordspotter, but now the new biphone lattices were used. The figure of merit obtained on the output of the wordspotter for the lattice terms was 71.85%; this represented an increase of almost 10% over wordspotting for the same set of terms in the monophone lattices. The new figure of merit could of course be improved by relaxing the assumption about the acoustic confusability of stem–equivalent terms, although attempts in Subsection 6.2.3 to improve wordspotter performance failed to translate to more effective message retrieval.

The wordspotter output was thresholded at the usual range of values and retrieval performed on the hybrid message representations. Table 6.12 and Figure 6.7 show the final results obtained using the hybrid method. The single best average precision figure obtained for spoken message retrieval was 0.6512, for the bigram with *OOV*–redistribution. This corresponded to a final effectiveness ratio value of 92.52%.

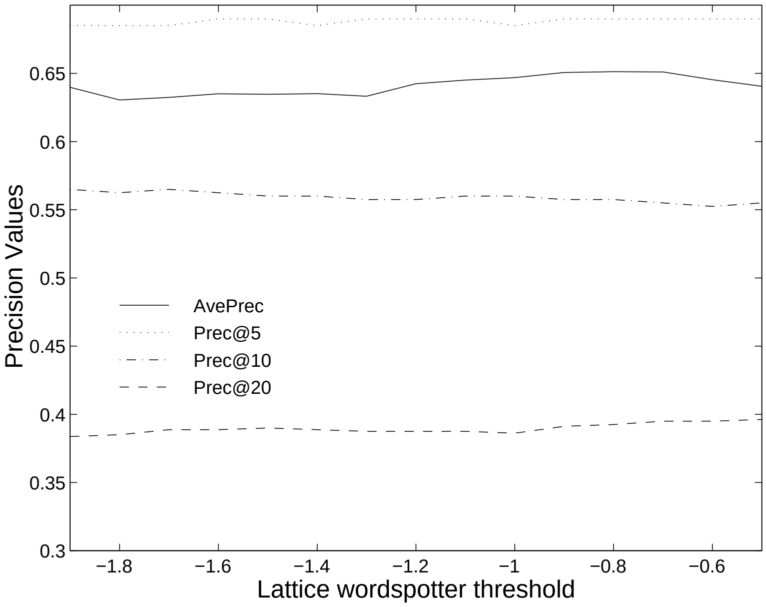


Figure 6.7: Retrieval precision on final hybrid system, with triphone–based word recogniser and biphone lattices.

6.5 Conclusion

The retrieval performances resulting from the hybrid strategies and existing monophone models represented a significant advance on the best results obtained in the previous chapter. This was the result of adopting a core domain–specific vocabulary, designed to include a large number of potential query terms, and using a conven-

	Wordspotter Keyword Thresholds															
	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	$-\infty$
AvePrec	0.6405	0.6454	0.6510	0.6512	0.6507	0.6469	0.6451	0.6424	0.6332	0.6351	0.6347	0.6350	0.6324	0.6305	0.6398	0.6423
Prec@5	0.6900	0.6900	0.6900	0.6900	0.6900	0.6850	0.6900	0.6900	0.6900	0.6850	0.6900	0.6900	0.6850	0.6850	0.6850	0.6750
Prec@10	0.5550	0.5525	0.5550	0.5575	0.5575	0.5600	0.5600	0.5575	0.5575	0.5600	0.5600	0.5625	0.5650	0.5625	0.5650	0.5500
Prec@20	0.3962	0.3950	0.3950	0.3925	0.3912	0.3862	0.3875	0.3875	0.3875	0.3887	0.3900	0.3887	0.3887	0.3850	0.3837	0.3900

Table 6.12: Retrieval performance for hybrid system with triphone–based word models and biphone lattices.

tional stemming algorithm to make useful identifications between terms. The results with the differing language models suggested that the simple no–grammar system was quite able to correct the grossest of the word detection errors made in Chapter 5 by the lattice wordspotter. The bigram model still proved useful, however, since it delivered a small increase in effectiveness, reduced the system's dependence on tunable parameters, and lowered the query–dependent lattice search time. In addition, state–of–the–art techniques were employed to generate biphone and triphone models, for word recognition and fast lattice wordspotting. These improved acoustic models were responsible for the final effectiveness ratio value of 93%.

Chapter 7

Experiments with the Retrieval Model

The hybrid technique has now essentially reached the upper limit of its performance, with the use of of a well–estimated language model and state–of–the–art acoustic models. It is therefore sensible to see how it might be extended by the application of a technique from text IR that has not yet been tested — relevance feedback. The improved acoustic models also facilitate the comparison of the hybrid term detection model with a method of acoustic indexing and retrieval that has been proposed in the IR literature, namely the VCV–feature method of Schäuble and Glavitsch [7]. This chapter describes both these experiments, and a further experiment which is motivated by the results of the comparison between the hybrid and VCV–feature methods.

7.1 The Application of Relevance Feedback

As described in chapter 3, relevance feedback is the iterative technique whereby an initial query can be reformulated in the light of new information generated by the assessment of a number of the documents at the top of a ranked list of retrieved documents. It allows both for the re–weighting of existing query terms, to reflect their ability to discriminate between relevant and non–relevant documents, and the expansion of the query by the inclusion of terms not included in the original query formulation.

The method adopted here involves weighting the revised queries with the probabilistic term relevance weight of Robertson and Sparck Jones [42], which was introduced in Section 3.4. This weight was used because it is well–understood and is related to the probabilistic inverse document frequency weight, which has already been shown in the spoken message retrieval experiments to have some considerable utility.

7.1.1 Feedback experiments on textual message representations

Feedback experiments were first performed on the text transcriptions of the spoken messages. It was not clear that the improvement in retrieval effectiveness, if any, would represent an upper limit on the improvement obtained in similar experiments on spoken messages. For example, suppose some discriminating query term is continually misrecognised by the word recogniser. So long as the recogniser *consistently* makes the same substitution, there is no reason why the incorrect word label could not be added to a reformulated query, if it was observed to occur in messages assessed as relevant.

The first feedback experiment was performed on stemmed textual transcriptions of the spoken messages. Ordinary retrieval, with no relevance information, was performed individually on each of the queries, q_j, using the standard np–weighting. The operation of feedback in a real retrieval environment was simulated by selecting the top c messages from the ranked retrieval output and removing these from the set of messages. Since the relevance assessments for these c messages were of course already known, probabilistic term relevance weights for each term t_i observed in the set of "cut–off" messages could be calculated using the standard equation

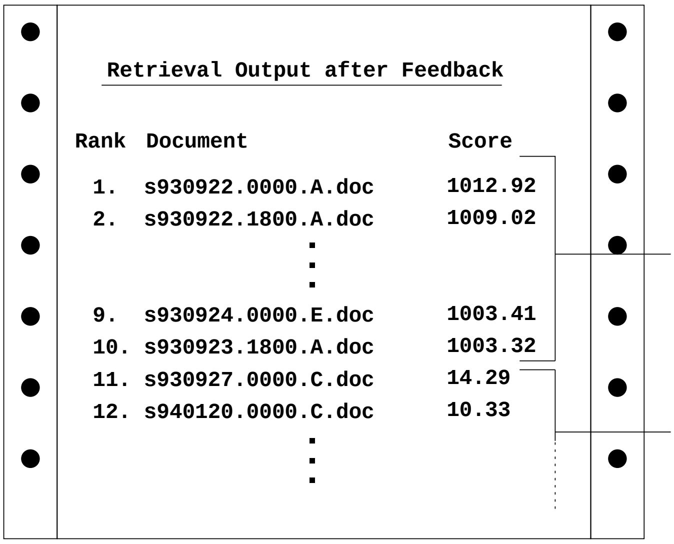
$$w_i = \log \frac{r_i(N - n_i - R + r_i)}{(R - r_i)(n_i - r_i)},$$

where the notation is that introduced in Section 3.4. The weighted terms were now ranked by their *Offer Weight* [40], simply defined as

$$O_i = r_i * w_i,$$

and the top q terms selected for inclusion in the new query.

Messages were now retrieved from the *residual* of the message collection using the newly formulated query. The entire retrieval output, based on the initial and fed–back queries, was ranked by adding a large constant to the scores of the initially–assessed messages, so that the messages were ranked in exactly the order they would have been seen by a user in a real retrieval environment. Figure 7.1 illustrates this method. It is a fair assumption that the number of terms q that are incorporated in the reformulated query should depend on the number of assessed messages used in feedback; therefore, in each experiment, q was set to be equal to c, the number of assessed messages. As in Robertson's work, the term relevance weight replaced the inverse document frequency weight in query–message matching. The term frequency weighting component of the query–message score was the usual normalised value. Table 7.1 shows the precision values obtained for this experiment for differing cutoff values c and compares them with the precision values achieved without the use of relevance information. The results are given in terms of average precision, and precision after the retrieval of $c + 10$ messages, with and without feedback. This measure was chosen since effective relevance feedback should improve the ranking of relevant messages in the top 10 messages retrieved with a second retrieval pass.



```
   Retrieval Output after Feedback

Rank  Document                 Score
  1.  s930922.0000.A.doc       1012.92
  2.  s930922.1800.A.doc       1009.02
               .
               .
               .
  9.  s930924.0000.E.doc       1003.41
 10.  s930923.1800.A.doc       1003.32
 11.  s930927.0000.C.doc       14.29
 12.  s940120.0000.C.doc       10.33
               .
               .
               .
```

Documents assessed by user after retrieval on initial query, with 1000 added to scores.

Documents retrieved by reformulated query after the application of relevance feedback

Figure 7.1: The implementation of the *residual* method of scoring retrieval output with the use of relevance feedback.

$c = q$	Average Precision		Precision at $c + 10$ messages	
	No feedback	Feedback	No feedback	Feedback
10	0.7038	0.7129	0.4238	0.4387
20	0.7038	0.7139	0.3133	0.3267

Table 7.1: The effect of feedback on retrieval precision.

It can be seen that the improvements obtained on retrieval over the whole query set, are only of the order of a single precision point. This increase can be seen not to correspond to a practical improvement in retrieval effectiveness. Where the cutoff is set to 10 messages, the performance of the original system is such that after 20 messages are assessed, an average of 8.48 messages are relevant. In the feedback system, in which queries are reformulated in the light of assessments made on the top 10 retrieved messages, an average of 8.77 of the top 20 retrieved messages are relevant. Thus, feedback only increases the average number of relevant messages in the top 20 retrieved messages by 0.29.

Examination of the performance of individual queries, before and after re–formulation, gives an indication of why, in this case, feedback fails to deliver a useful increase in effectiveness. When feedback is based on the assessment of the top 10 retrieved documents, a 5–point improvement in average precision is recorded for 11 of the queries; however, a similar decrease occurs for 8. An example pair of queries, one improved by the use of relevance feedback, and the other made poorer, are shown in Table 7.2.

The Political Crisis in Russia		The UK Press	
Initial Query	Fed–back Query	Initial Query	Fed–back Query
khasbulatov	bori	britain	accept
polit	build	censorship	bid
russia	dissolv	media	group
soviet	mister	newspap	independ
union	moscow	press	newspap
yeltsin	power	uk	publish
	russian		sharehold
	struggl		stake
	support		sundai
	yeltsin		televis

Table 7.2: Initial and Fed–back queries for two of the query prompts.

The majority of messages in the entire collection, assessed as relevant to the *russia* query, actually referred to the unfolding of a specific news event[1]. This story was reported in successive bulletins early on in the message collection. In retrieval and relevance feedback, a new query was generated which improved matching on relevant unseen items. However, items assessed as relevant to the *press* query were much more heterogeneous in nature, as might be expected by the rather broadly worded prompt. Relevance feedback generated a new query which turned out to reflect the content of a specific news story[2]. The narrow nature of the new query significantly

[1] The storming of the Russian parliament building in August 1993.

[2] The proposed takeover of the *Independent* newspaper in January 1994.

reduced the effectiveness of retrieval on this thread.

The behaviour of each of these two fed–back queries points towards *undersampling* of the message collection, since retrieval has benefited only where message content is reasonably consistent throughout the set of relevant messages. Although more representative queries were later constructed by taking a cutoff of 20 on the set of initially retrieved messages, no greater improvement in retrieval effectiveness was recorded, since a smaller proportion of the averaged precisions are those obtained after feedback. With such a small collection, the robustness of the estimation of a fed–back query has traded off significantly against its future utility — the more items were assessed, the fewer there remained to find.

7.1.2 Feedback experiments on spoken messages

Experiments on the spoken messages were now performed. The message representations that had generated the best effectiveness ratio of 92.52% in the previous experiments were used, and the message cutoffs were set to the same values as for the earlier text–based work. Table 7.3 shows again that the increase in precision values obtained through the use of feedback at either cutoff value is not very significant.

$c = q$	Average Precision		Precision at $c + 10$ messages	
	No feedback	Feedback	No feedback	Feedback
10	0.6512	0.6576	0.3925	0.4025
20	0.6512	0.6638	0.2950	0.3117

Table 7.3: The effect of feedback in spoken message retrieval.

Feedback performance was poor here for a number of reasons. As illustrated by the *press* query in Table 7.2, the small number of messages relevant to each query means that a query reformulated using a subset of the relevant messages may not be representative of the remainder of the set. In addition, with the retrieval effectiveness on the full set of original queries already very high, as it was here, with an average precision of 0.6512, the potential for improvement was clearly limited. The undersampling problem was shown to cause the feeding back of unrepresentative terms into the queries. This can be prevented by introducing a *supervised* selection procedure, in which the user is shown a ranked list of the proposed new query terms, and invited to select a number of them for inclusion in the new query. This procedure was not possible in these experiments, however, because of the delay between query collection and retrieval experimentation.

Despite the poor results described above, it is possible to demonstrate the effectiveness of feedback in spoken message retrieval, simply by starting with a poorer initial set of queries. When requests and relevance assessments were initially collected, two sets of requests were actually obtained; the unconstrained–term requests used in the experiments throughout this thesis, and a set of corresponding natural language sentence requests. On inspection, these requests were found to bear too much similarity to the set of prompts, and to contain an average of only 3.8 query terms each, and so were not used in the experiments described so far. However, they are useful here to illustrate how an initially poor set of queries can be improved by the application of relevance feedback, even in experiments on this relatively small collection. Table 7.4 illustrates that relevance feedback produces significant performance improvements in retrieval on both textual and spoken message representations. Spoken message retrieval for the set of queries derived from these requests, was of course based on the same hybrid term detection approach as has been used throughout.

	$c = q$	Average Precision		Precision at $c + 10$ messages	
		No feedback	Feedback	No feedback	Feedback
Text	10	0.5444	0.5886	0.3425	0.3837
	20	0.5444	0.5892	0.2492	0.2975
Speech	10	0.4737	0.5100	0.3100	0.3487
	20	0.4737	0.5106	0.2325	0.2767

Table 7.4: The effect of feedback when retrieval is initially performed with a poorer set of initial queries.

Where retrieval was performed on spoken messages, inspection of the fed–back queries and the message transcriptions revealed that a number of the new query terms were not actually spoken in any messages relevant to the corresponding query. They appeared in the queries because they were consistently substituted for discriminating terms by the word recogniser. Table 7.5[3] details some of these substitutions. The small collection size here does not really make it possible to determine the usefulness of these terms in subsequent matching. However, it can be seen that the phenomenon would obviously complicate *supervised* query expansion, since users would be asked to make a decision on the inclusion or otherwise into the reformulated query of terms which did not *appear* to be remotely related to the underlying information need.

[3] "The Belfry" is a golf course near Sutton Coldfield in the UK.

New Query Term	Representing
steel	mi*ster yel*tsin
george	*georg*ia
list	loya*list*, nationa*list*
bell, free	*belfry*

Table 7.5: Some recogniser substitutions actually incorporated into the fed–back queries.

7.1.3 Summary of Results on Relevance Feedback

In summary, although the proven methods of relevance feedback performed disappointingly poorly on the main set of queries, a far more significant improvement was observed in experiments in which the initial set of queries was somewhat poorer. A particular problem here was the small size of the message collection and relevance sets, compared to the document collections on which modern text retrieval experiments are performed. This meant that reformulated queries were not always representative of the corresponding set of relevant messages.

The most interesting result here is that feedback has in some cases led to the automatic acquisition of new query terms whose underlying meaning actually differs from that of the word or phrase whose acoustic realisation was matched, in word recognition, by that term. The result suggests that it might be possible to get away from the exact matching of whole words as the basic operation in the indexing and retrieval of spoken messages. This, in fact, has been precisely the thinking behind the work of Schäuble and Glavitsch in their experimental simulations of spoken message retrieval [7]. Since sophisticated acoustic models have already been trained for the hybrid term detection strategy reported hereto in this thesis, it seems sensible to implement the VCV–feature retrieval strategy using these models, and see whether it offers any advantages compared to the word–matching approach.

7.2 Implementation of VCV–Feature Retrieval

The approach to message retrieval proposed by Schäuble and Glavitsch obviates the need to perform any query–dependent acoustic matching. It is based on the selection, for a message domain, of a population of so–called *VCV–features*, using criteria related to the discriminative ability of each such feature and the frequency of feature occurrence in the acoustic training data. The test data is recognised using the population of acoustic feature models, and query–message matching performed by decomposing the query into its constituent features. Schäuble and Glavitsch have presented results demonstrating the effectiveness of the approach on errorful VCV–feature message representations in simulated retrieval experiments [7, 50].

A number of small alterations to the above method were first necessary. Firstly, since acoustic biphone and triphone models had already been trained, and phonetic decision trees built, there was no need to enforce a frequency criterion in the selection of the feature population, since an acoustic model for any feature could be built from existing or synthesizable biphones or triphones. It was also decided to select the population of features from the *test* data rather than the training data. This is because it had been necessary, for the acoustic model training, to segment the training data at the individual sentence level, rather than the "news story" level. A large amount of effort would have been necessary to generate news–story transcriptions of the data in order to use this data to select the feature population. Retrieval results obtained in this experiment should consequently be thought of as *retrospective* to some extent, since the set of features is derived from explicit test data knowledge. Also, the feature detection model proposed by Schäuble and Glavitsch depends on an entire recognition pass through the message collection to detect each single separate indexing feature. In practice, this would of course be extremely computationally wasteful (and of course, significantly more prone to false alarms); here, a more conventional approach is taken, with recognition based on a parallel network of indexing features together with a garbage model.

7.2.1 Feature Selection and Acoustic Modelling

Firstly, it was necessary to select the population of VCV–features that would be used in indexing and retrieval. The text transcriptions of the spoken news reports were parsed and each word converted to its string of component VCV–, CV– and VC–features, as illustrated in Figure 7.2. This generated a total of 3306 VCV–features. Next, the *term discrimination value* was calculated for each feature [88].

The term discrimination value measures the degree to which the inclusion of a term in a document collection increases or decreases the dissimilarity of the documents from each other. Frequently occurring terms (such as function words or here, features derived from them) are too commonplace to discriminate documents from each other; rare terms are also non–discriminating since they do not occur frequently enough to have a significant impact on the collection. Formally, a vector model of retrieval is assumed, with binary term presence or absence, and an Euclidean distance measure

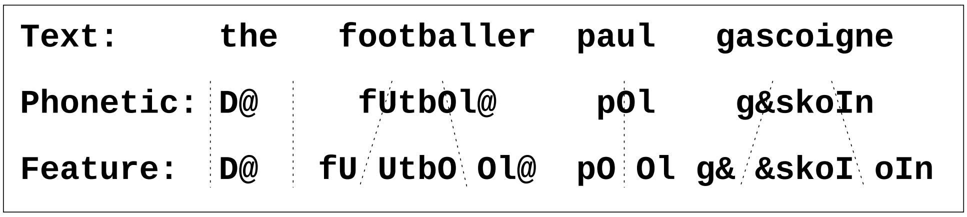

Text: the footballer paul gascoigne
Phonetic: D@ fUtbOl@ pOl g&skoIn
Feature: D@ fU UtbO Ol@ pO Ol g& &skoI oIn

Figure 7.2: The decomposition of a sentence into its component VCV–features.

S between document vectors D_i and D_j where

$$D_i = (d_{i1}, d_{i2}, \ldots d_{in}).$$

The centroid C of a document collection is

$$C = (c_1, c_2, \ldots c_n)$$

where

$$c_j = \frac{\sum_{i=1}^{m} d_{ij}}{m}.$$

The compactness of the collection, Q is defined by

$$Q = \frac{1}{m} \sum_{i=1}^{m} S(C, D_i).$$

If term t_k is now removed from each document in the collection to create a new document collection, in which

$$D_i' = (d_{i1}, \ldots, d_{i(k-1)}, d_{i(k+1)}, \ldots, d_{in}),$$

and C' and Q' are the centroid and collection compactness of this new collection, then

$$tdv(t_k) = \frac{Q' - Q}{Q}.$$

Features were ranked by their term discrimination value and the top 1000 selected as the set of indexing features for the retrieval experiment. As can be seen in Table 7.6, examination of the top of the entire ranked list of features illustrates the link between the best features and the test data vocabulary from which they were derived. As would be expected, the other end of the list contains very many features extracted from function words, as well an inflexional suffix.

The single biggest problem with building acoustic models for the VCV–features is that in Viterbi decoding, a single non–overlapping sequence of acoustic unit hypotheses must be output. Since, as described in chapter 4, adjacent VCV–features share a

Rank	Feature	Corresponding Words
1	faU	found
2	Ots	courts, sports, reports
3	dZV	judge
4	id	freed, leader, media, proceedings
5	laI	airline, flight, life, supply
3301	IN	–ing
3302	t@	to
3303	h&	has
3304	ov	of
3305	In	in
3306	D@	the

Table 7.6: The top and bottom of the ranked list of VCV–features.

common vowel, some method must be found to model each half of the single vowel occurrence separately, so that feature sequences can be detected.

In practice, this problem was dealt with as follows. Acoustic models for the indexing features were built from the set of biphone and word–final monophone models that were used in the production of biphone lattices for the final wordspotting experiments. In these models, the transition matrices had been tied together before model cloning; now, for each base phone, the transition matrices were untied and the biphone matrices all retied, thereby freeing the monophone transition matrix. It was now intended that each vowel occurrence should be shared across two features, with each training data vowel now modelled by the final vowel monophone of the first feature, and the initial vowel biphone of the second. This would allow the construction of non–overlapping VCV–feature models for the forthcoming experiment. Thus, it was necessary to enrich the topologies of all vowel biphone and monophone transition matrices and re–estimate them, to ensure a good "fit" between the acoustic feature models and the data. This was done by adding so–called *skip* transitions to the vowel models, as illustrated in Figure 7.3. A transition probability of 0.3 was added to each of the state transitions illustrated by a dotted line, and the other transition probabilities reduced to ensure that the sum of transition probabilities out of each state was still 1.0. The training data biphone label files were edited to insert a vowel monophone before every vowel biphone, and the new models subjected to four cycles of embedded re–estimation with the new label files. Figure 7.4 illustrates a Viterbi alignment[4] of the new models to the acoustic training data.

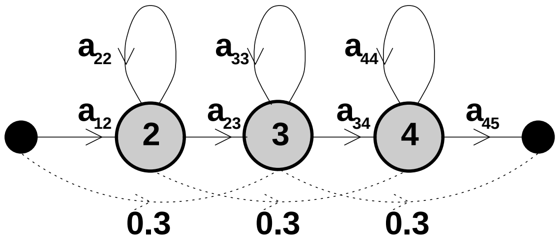


Figure 7.3: The "skip" transitions added to the vowel models to allow their use in VCV–feature modelling.

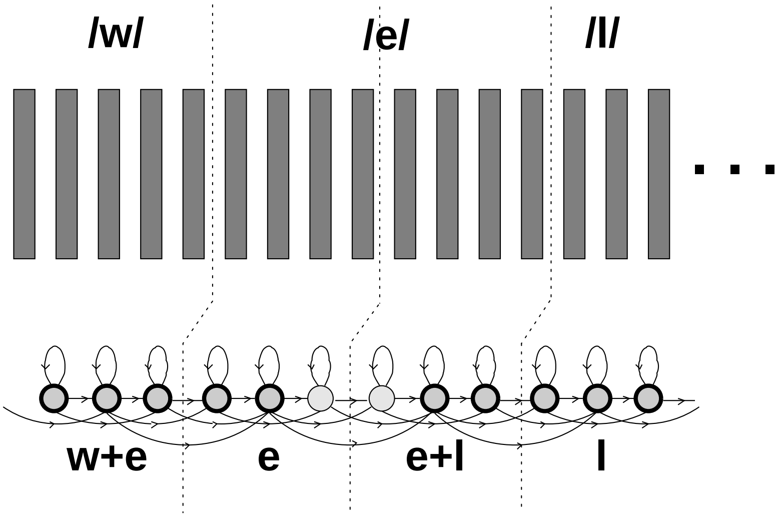


Figure 7.4: The new alignment of models against a training occurrence of the word *well*. The enrichment of the state transition matrices by the addition of skips can be seen, as can the new sharing of the single vowel between two models. Dark circles indicate the model states assigned speech observation vectors by the alignment — it can be seen that the vowel is now modelled by four emitting states instead of 3.

Feature models were now concatenated from the newly re–estimated models, and a null–grammar parallel network of VCV–features built, with a garbage model consisting of the set of monophone models in parallel. Timed feature alignments of the message collection were also generated, so that the performance could be measured in terms of detection and false alarm rates. The detection performance on the entire set of features was 38.55% at a rate of 8.88 false alarms per feature per hour. On the

[4] This is for illustration only – Baum–Welch training is actually used to re–estimate the models.

set consisting only of those features occurring in the queries, the performance was 40.55% correct detection at a rate of 10.7FA/feature/hr. Since Schäuble and Glavitsch's VCV–feature method does not associate acoustic scores with features, it is not meaningful to calculate a Figure of Merit for feature detection. Table 7.7 illustrates feature output for the example "gascoigne" sentence of Section 6.2.

```
LAB: D@ fU UtbO Ol@ pO Ol g& &skoI oIn h& &z bi in In trV Vbl @ge en
REC:  <---garb---> @pO    g&  sko <-----garb-----> Int@   Vbl @ge end
```

Table 7.7: Feature recognition for the sample sentence.

Retrieval was now carried out by decomposing the queries into their component VCV–features and calculating a query–message score in the usual manner. As usual, np–weighting gave rise to the best average precision; however, the best average precision here was 0.3374. Comparing this with the average precision of 0.6693 obtained when matching on feature decompositions obtained directly from the text, this represents an effectiveness ratio of only 50.41% for the method. Retrieval effectiveness was thus disappointingly poor, in comparison with the hybrid, word–matching approaches discussed in this thesis. In fact, the effectiveness ratio obtained here was considerably lower than that observed in the 4–lattice experiments in Section 5.5.

It is instructive to compare the results above with the effectiveness ratios reported by Schäuble and Glavitsch in their work with the *simulated* recogniser output [50]. In experiments on three differing standard text document collections, simulated output at a feature detection rate of 40%, and a false alarm rate of 10 FA/feature/hour, gave rise to effectiveness ratios of 60%, 53% and 30%. Averaging these together gives a figure of roughly 48%, which is near the effectiveness ratio observed in the experiment above. Although this is a rather unscientific comparison, it at least shows that Schäuble and Glavitsch were making reasonable assumptions about the nature of speech recognition in their simulation work.

However, Schäuble and Glavitsch do not seem to have made realistic assumptions about the performance of current speech recognition techniques, as many of the detection rates for which they performed experiments represent recogniser accuracies far in excess of what was actually obtained in the experiment performed here. It would be extremely difficult to improve the feature detection rate to a value as high as, say, 70% whilst keeping the false alarm rate down to an acceptable level that would not result in the swamping of the correctly detected features.

There seems little potential to constrain feature recognition. Whereas conventional word recognition can be constrained by the use of a probabilistic n–gram language model, the feature recognition system cannot really take advantage of such an approach, since it includes a garbage model which has to model too many differing acoustic events. It would be possible to introduce a level of dependency between adjacent features if features were extracted from training word sequences *across* word boundaries, instead of only *word–internally*. This would ensure that all the features in the recognition vocabulary would be bounded on the left and right by a vowel. Obviously, only a limited number of features, or the garbage model, would then be known to follow each feature in the recognition sequence, and this constraint enforced in recognition by the construction of a suitable network. Moreover, the construction of cross–word features would permit the automatic inclusion in the indexing vocabulary of discriminating cross–word features occurring in important phrases.

Another problem posed by word–internal feature decomposition is that acoustic confusability is not a factor in the selection of the feature population. Many word–initial or word–final features are typically of the form CV or VC, and therefore of shorter duration than the word–internal VCV–features. This means that many pairs of features, such as (`&Nk`, `&NkSnz`) and (`&S`, `&S@`) are acoustically overlapping and as such, highly confusable. In fact, of the 310 CV and VC features in the recognition vocabulary, well over *half* are confusable with a VCV–feature in the same vocabulary.

This is a significant problem, as for every word w, a *unique* feature decomposition exists — where two features may match well against some unknown speech, only one of them could subsequently be useful in retrieval. It can also be seen from the table that a number of feature pairs differ from each other solely in the identity of an initial or final *demivowel*. As has already been discussed, there is no obvious way to constrain feature sequences; the hypothesis of a particular feature instead of another can depend solely on a few speech frames. This is a source of considerable inaccuracy in feature detection.

7.2.2 Improvements to Feature Modelling

In an attempt to improve VCV–feature indexing, more accurate feature models were built by synthesizing the required set of left biphones, triphones and right biphones using the set of decision trees used in the earlier triphone–based word recognition experiments. The earlier method was used again to add skips to the transition matrices of the left and right biphones used to model vowels at feature boundaries. However, in a change to the previous procedure, these transition matrices were not subject to any cycles of embedded re–estimation. Tests on the set of biphone and monophone models re–estimated for the earlier feature set, showed that the re–estimation of the transition matrices, after the uncoupling of the vowel transition matrices and the addition of skips, had had a negligible effect on VCV–feature recognition output. This is because in general, the log transition probabilities a_{ij} of the HMM model are dwarfed by the the acoustic log likelihoods, $b_j(o_t)$. It seemed that as long as a reasonable estimate of the skip transitions was used (in this case, 0.3), the computationally expensive embedded re–estimation of the transition matrices could be avoided. This seems to be another example of the poor HMM durational modelling that was discussed at the end of chapter 2.

In addition, the old phones–in–parallel garbage model was discarded. Instead, *function word models* were created for the subset of words from van Rijsbergen's stop list that appeared in the textual transcriptions of the message collection. These models were generated by synthesizing the required triphone models using the decision tree method and then concatenating these models. A parallel network of function word models should be a suitable garbage model, since the function words are those words whose component features should have a low term discrimination value and therefore should not be modelled by the indexing features. In addition, recognition of function words enforces word boundary constraints, which should help to reduce the incorrect detection of word–internal features at word boundaries. Table 7.8 illustrates the output for the usual example sentence.

```
LAB: the fU UtbO Ol@  pO Ol  g& &skoI oIn has been in trV Vbl again
REC: the fU UtbO all @pO Old g&  sko  oIn has been in tr& Vbl again
```

Table 7.8: Improved feature recognition for a sample sentence.

The newly generated feature-based output was scored in the usual way. The detection rate rose from 38.55% to 48.93%, with the false alarm rate increasing slightly from 8.88 FA/feat/hr to 9.87 FA/feat/hr. In retrieval, average precision improved from 0.3374 to 0.3906, with a new effectiveness ratio of 58.36%. Feature detection and retrieval can still be seen to be poor, even with the use of state–of–the–art techniques to generate acoustic feature models.

7.2.3 Conclusions

The VCV–feature method of retrieval has performed disappointingly. Even in the final implementation presented here, involving high–quality acoustic models, the method has offered no performance advantage over a much simpler approach tested in Chapter 5, based only on much poorer acoustic models and the simplest imaginable word grammar. Areas of particular weakness in the method are the modelling of demivowels, especially where these discriminate between similar features, and the use only of *word–internal* features, leading to poor acoustic feature modelling at word boundaries.

One approach to these problems would involve *cross–word* features to deal with the word boundary problem, and the use of acoustic units larger than the demivowel, so that feature discrimination would be based on a greater amount of acoustic data. The final experimental section describes a new retrieval method that attempts these solutions.

7.3 Data–driven indexing

The VCV–feature approach to indexing and retrieval is strongly *knowledge–driven*; the feature set is obtained not from consideration of which features are discriminating and also relatively easy for the recogniser to detect, but solely from a criterion related to feature occurrence in textual transcriptions of the acoustic data, whether training or test. The previous section illustrated the resulting mismatch between what was required for good indexing and retrieval performance, and the ability of the recogniser to detect the chosen indexing units.

McDonough *et al* described similar problems in their paper on topic classification on the Switchboard corpus [47]. Their approach to this problem was to select the vocabulary of words used for classification solely from the output of their speech recogniser on a development test set. They found that the matching of the indexing vocabulary to the recogniser output significantly improved classification performance.

This result suggests an new approach to message indexing and retrieval. Sub–word, indexing features are derived from recogniser output, in this case the phone sequence generated by a phone recogniser. Since the indexing vocabulary is now *descriptive* of the recogniser output rather than *prescribing* what is expected of the recogniser, its size need not be limited, as earlier, by the term discrimination value criterion. In addition, no garbage model is needed to model "non–feature" speech; the use of the standard inverse document frequency weighting will reject all frequently occurring features, mainly those derived from function words and inflexional suffixes, when query–message matching is performed. Further, a simple constraint can be applied to the generation of output label sequences by the use of a phone bigram. The next subsections describe the derivation of the new message representations and the approach to query–message matching, and discuss the retrieval results obtained by the new method.

7.3.1 Data–driven Indexing Features

In theory, a number of differing approaches could be used to construct a message representation from the sequence of labels output by a phone recogniser. Moore [89] recently used dynamic programming (DP) to match similar but non–identical phone sequences produced by an extremely poor monophone recogniser, whose output was found to be only 22.3% accurate. The features were shown to be useful in a 2–class classification problem involving the detection of weather forecasts from continuously running Radio 4 speech, as the forecasts were spotted with 79% accuracy.

In this task, the accuracy of the biphone recogniser is a lot higher than 20% (it is actually 70.02%, as measured in Section 7.3). This suggests that a useful set of message representations could be obtained by fixing the feature width, and simply moving a "window" over the phone recogniser output. Figure 7.5 illustrates this method in operation. A window size of 3 phones should represent a good tradeoff between feature discrimination and correct detection, since the larger the window, the more likely it is to contain an incorrectly recognised phone and consequently be incorrect itself.

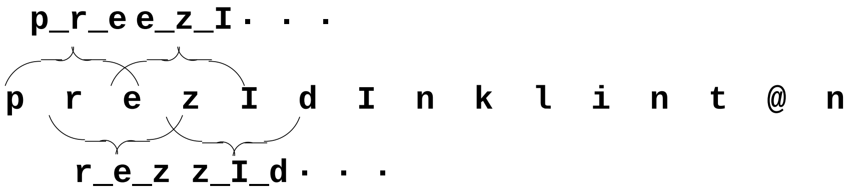


Figure 7.5: The generation of data–driven features by moving a window along phone recogniser output on the unknown messages.

The overall phone trigram detection performance, in terms of correctness and accuracy, was 43.06% correctness and 40.69% accurate. However, this performance is measured over *all* the unknown speech, much of which is not of interest from a retrieval point of view. It is more appropriate to obtain a score that reflects the detection performance on the "interesting" phone trigrams — those occurring in the queries. This score can be measured in terms of the usual wordspotter performance measures of correct detection percentage and false alarm rate. Here, the 656 differing trigrams occurring in the query terms were detected at a rate of 58.20% correctness, and a rate of 6.42 FA/trigram/hour. It can therefore be seen that the query trigrams are easier than average to detect. This is probably because the uninteresting function words are a considerable source of error in the biphone recogniser, but only a small number of trigrams occur in both query terms and function words.

Comparison with the VCV–feature detection rates shows that the detection of phone trigrams was considerably more reliable. In addition, whereas only 257 of the total of 431 query VCV–features appeared in the 1000–feature recognition vocabulary, the coverage of the query trigrams in the message representations was obviously 100%. In other words, 58% of *all* required phone trigrams were detected by the biphone recogniser. In comparison, only 59% of the query features were known to the VCV–feature recogniser, which detected, at best, only 50.96% of the occurrences the features in its vocabulary.

7.3.2 Query–Message Matching and Retrieval Results

The phone trigram method of generating message representations was adopted because it is fairly intuitive, and because the biphone recogniser was known to be sufficiently accurate to allow for the generation of discriminative trigrams. The output of a much poorer phone recogniser would certainly not have been good enough to allow for the generation of sufficiently accurate trigrams. Since it is known that 58% of trigrams are correctly detected, at an acceptable rate of false alarm production, it seemed sensible in this case to perform retrieval by converting the query terms to strings of phone trigrams and matching these against the content of the message representations.

The method was found to work considerably better than either of the VCV–feature approaches. The np–weighted average precision was in fact 0.4916, comparable with that of the best monophone–based word recognition approach without query–dependent wordspotting. Since it did not rely on a pre–chosen vocabulary and language model, or query–time acoustic searching, this system offers a very good tradeoff between vocabulary dependence, query–dependent search time and retrieval effectiveness.

It should be clear that this system has considerable scope for improvement. For example, a more sophisticated technique for extracting indexing features, such as DP matching, would allow the generation of discriminating features of varying lengths. In addition, the cross–word generation of features should readily allow for the automatic incorporation of discriminating phrasal units into message representations. Cross–word phone trigram matching was not performed in the experiment here, since the query terms had been supplied individually, so no phrases were known.

7.4 Conclusion

This chapter described experiments with the established retrieval technique of relevance feedback and performed a comparison between the established hybrid term detection model and two other recognition and indexing strategies. Firstly, experiments with relevance feedback showed that this technique had potential to improve retrieval effectiveness where the initial queries were quite poor. A subsequent experiment saw the implementation of a spoken message retrieval system using Schäuble and Glavitsch's VCV–features. It was found that retrieval effectiveness was poor, owing to the relative acoustic confusability of features whose selection was motivated purely by textual criteria. Finally, an attempt was made to solve some of the problems encountered with VCV–features by extracting sub–word indexing units from label sequences output by a biphone recogniser. This proved quite successful, delivering reasonably good retrieval performance without the need for task–dependent language modelling or any query–dependent search effort.

Chapter 8

Conclusion

8.1 Review of Experiments Performed and Conclusion

This thesis has described the application of well–understood textual IR techniques to the task of locating items in a collection of non–textual documents — in this case, acoustic waveforms sampled from recordings of radio news broadcasts. This was motivated by recent results in experimental text IR, which have shown that good retrieval effectiveness can be obtained by expressing each document in a relatively simple form, such a list of word stems, statistically weighted where appropriate, without the need to perform complex linguistic analyses of document texts or queries.

The experiments described in this thesis avoided the methods typically used in the experimental topic classification work described in the speech recognition literature. In topic classification, individual speech recordings are classified uniquely into some number of known classes, based on the hypothesis in the unknown speech of pre–determined words, and the association with each such word of a pre–trained weight. Here, however, since the IR methodology involves the collection of *ad hoc queries* and *relevance assessments* from potential users of a retrieval system, there was no potential to select a specific vocabulary of useful terms, or assign, in advance, weights related to the occurrence of these terms in the desired recordings. Although the message collection was relatively small, compared to modern textual corpora for the development and testing of retrieval systems, it was estimated that this message collection was sufficiently big to allow for the indexing and retrieval of individual stories from news bulletins totalling 11 hours in length.

It was realised from the very beginning of the experimental work that in a practical environment, the potential for query–dependent acoustic searching was extremely limited, with the consequence that a conventional approach to wordspotting would not be suitable for the detection of query term occurrences in the spoken messages. A new approach to wordspotting, based on the detection of phone strings in the pre-computed output of a phone recogniser, offered unlimited flexibility in the choice of query term whilst also delivering speedy performance. The initial retrieval results demonstrated the utility of the statistical term weights historically employed in textual document retrieval. It was also observed that the standard retrieval methods were relatively robust to wordspotter errors.

Several factors ensured good recognition performance. The use of read speech from the same seven trained speakers in both the acoustic training data and the message collection, the high production values of BBC Radio News, and the availability of good FM radio reception, all had an effect on wordspotter figures of merit, and recogniser correctness and accuracy measures. Speech recognition performance accuracy generally degrades when training and test speakers differ, where speech is spontaneously spoken rather than read, and where bandwidth is limited (for example in telephone speech) and the acoustic signal dirty (such as in the presence of background noise). It is unavoidable that any of these factors would have lowered the retrieval effectiveness of the spoken message systems described here. In addition, it was observed in chapter 5 that the retrieval of messages stories from the collection using the principal set of queries was quite easy, due to the small size of the collection, and the small numbers of documents relevant to each query.

Next, the recognition of frequently occurring, short query terms was improved by the adoption of a hybrid term detection system. Viterbi recognition was performed for a vocabulary of frequent terms selected from the acoustic training data. This more competitive approach to term detection generated far fewer false alarms and reduced the query–time burden of the lattice wordspotter. The use of a conventional stemming algorithm to identify acoustically differing but semantically related terms was also investigated. Extending the recogniser vocabulary and including a probabilistic language model improved retrieval effectiveness further. Performance was finally improved by using state–of–the–art techniques for acoustic modelling in both areas of the hybrid term detection system. Further, an experiment investigating feedback of initial relevance assessments to improve the queries, showed that relevance feedback was of some benefit where the initial queries were quite poor.

The hybrid retrieval system, built up throughout the experimental chapters of this thesis, was finally compared with two approaches employing sub–word acoustic units as the basic indexing feature. In the VCV–feature retrieval paradigm of Schäuble and Glavitsch, which used the same sub–word units for acoustic recognition and message indexing, performance was poor, even with the use of sophisticated acoustic modelling techniques. This was because the population of indexing features was chosen with respect to a criterion related only to the occurrence of features in text. This meant that features were too acoustically confusable to be recognised more accurately. In contrast, a fairly simple, data–driven approach to sub–word indexing, based solely on the output of a reasonably accurate biphone recogniser, was considerably more effective. However, the best performance by far was obtained with the hybrid word recogniser/lattice wordspotter method. The final hybrid system retrieved spoken messages very effectively indeed, performing 93% as accurately as the textual reference method. Appendix C illustrates the progression of retrieval results obtained.

The principal conclusion of this thesis, therefore, is that spoken messages can be retrieved quickly, flexibly and very accurately, at least on a small message collection with good acoustic models, and that the word is, so far, the most successful unit for acoustic recognition and message indexing.

8.2 Further Work

In text IR, many factors (collection size, the nature of document content, the query collection and the retrieval model) preclude the straightforward generalisation of conclusions drawn from a single experiment. Spoken message retrieval adds an even greater number of factors, such as the number of speakers and the acoustic channel characteristics. Despite these factors, which were touched on in the previous section, there are several experimental areas which could be explored using the existing data collection. Firstly, retrieval might benefit from exploiting the term detection scores output by the recogniser. The acoustic score analysis done here was initially limited to thresholding the ratio scores output by the wordspotter. This is the same method used by Jones *et al* [69]. It was used here because roughly similar scores would be assigned to a term hypothesis whether output by the conventional wordspotting method or the lattice wordspotter. Rose and his co–workers [48], and McDonough *et al* [47], directly incorporated acoustic scores into the message–topic scores output by their topic classifiers; however, these systems depended either on the ability to pre–train acoustic score mapping functions, or the assumption of a fixed classification vocabulary, neither of which was possible here.

The results presented in this thesis were based on techniques imported from textual information retrieval, such as stemming and relevance feedback. However, no experiments were performed to determine the effect on retrieval performance of the use of a *thesaurus* to perform query expansion without relevance information. This method, in which a query term is augmented by a number of similes, has been used in experimental text retrieval in attempts to improve recall. In spoken message retrieval, it might enhance retrieval by adding terms more easily matched than an original term, which might be a good descriptor in text but difficult to match in speech, for example for reasons of term length.

Also, as was commented on earlier, sub–word indexing shows a great deal of potential. The very simple approach adopted in chapter 7, involving trigrams derived from recognised phone sequences, achieved surprisingly respectable retrieval performance. The adoption of dynamic programming techniques might allow for indexing robustness even when phone recognition accuracy is quite poor, and a simple extension of the indexing method should facilitate the matching of message representations with *ad hoc* queries. A successful sub–word method would allow phone recognition and message indexing to be carried out in a single word–independent pass, therefore obviating the need for major query–time processing or the adoption of a task–dependent approach to acoustic modelling.

Finally, it should not go without comment that potential speech retrieval researchers face the heavy burden of collecting their own experimental corpora for acoustic training and message retrieval. The provision to the research community of a single acoustic training data corpus and large spoken message collection would significantly advance the state of the art.

Appendix A

The following illustrates the exact values used in the parametrisation of acoustic training and test data for the experiments presented in this thesis. The parametrisation is performed by the *HCode* computer program supplied with Version 1.5 of the *HTK* Hidden Markov Model Toolkit [10].

Option	Description	Value
-e	Append log energy	—
-f t	Set frame period to t milliseconds	10.0
-h	Apply Hamming Window	—
-k f	Set waveform pre–emphasis coefficient to f	0.97
-l n	Set cepstral liftering coefficient to n	22
-m	Output MFCC coefficients	—
-n n	Set number of output parameters to n	12
-p n	Set order of analysis to n	24
-s f	Scale log energy by f	1.0
-w t	Set window duration to t milliseconds	25.0

The subsequent setting of the *HCOERCE* environment variable to *MFCC_E_D_A_Z* during training and recognition expands speech vectors by the addition of first and second order differential information. In addition, for each utterance, the cepstral mean is subtracted from the speech vectors, in order to improve channel robustness.

Appendix B

The following illustrates a number of news items appearing during a single, half–hour evening news bulletin on 22nd September 1993. They are given as an illustration of the general content of each item and of the variation in item length.

At least thirty eight people are feared dead and more than fifteen are missing after a passenger train jumped the tracks as it crossed a bridge in the American state of Alabama. Part of the train plunged into a swamp inhabited by alligators and one carriage is completely submerged. The accident happened in the early hours of the morning in a remote area north of Mobile. The train with more than two hundred people on board, had been en route from Los Angeles to Miami. Up to ten British people thought to be students were on board, but it's thought they were not hurt. Simon Grant, one of the passengers, explained what happened. The water in some parts is believed to be about twenty five feet deep, and rescuers have only been able to reach the area by rail boat or helicopter. A local journalist John Nodar has been following events. It's thought it could be some hours before the scale of the disaster becomes clear. Officials say this is the worst crash in the history of the Amtrak rail company, which was set up in nineteen seventy to run America's long distance services.

At least sixteen people are reported to have died in a collision between two trains in India. More than sixty passengers were injured, some seriously. It happened in the state of Rajahstan, south of Delhi. Officials suggested that a goods train had gone through a stop signal before ploughing into a passenger train.

The Home Secretary Michael Howard has said legal processes must be changed to ensure that the guilty are brought to justice. He was addressing the police superintendents association conference in Torquay which yesterday declared that the criminal justice system had failed the country. Mister Howard angered many sections of the police by using his speech to announce controversial changes to disciplinary procedures.

All the main parties have put law and order high on the political agenda this autumn. The Liberal Democrats have already had two conference debates on the issue. Conservatives will be looking to the Home Secretary to help the party regain its traditional lead on crime at their conference this month.

Ford is to reduce the number of Escort cars made at its Halewood plant in Merseyside. From next month production will be cut back by a third. The company blamed the move on a decline in European sales. It follows announcements of production cuts earlier this week by Nissan at Sunderland and Vauxhall's plants in Britain.

Golf. Bernhard Langer has announced that he will be in the European team for the Ryder cup. There had been doubts about his fitness, but after a practice round he said he would be able to play at the Belfry later this week.

The short list for the twenty fifth Booker Prize has been announced. One of the strongest contenders is the Irish writer Roddy Doyle with his book, Paddy Clarke Ha Ha Ha. The other authors are Tibor Fisher, Michael Ignatieff, David Malouf, Carol Phillips, and Carol Shields. The result will be announced next month.

In addition, the following illustrates the 5 messages assessed as relevant to the request concerning train crashes, which is given in Section 5.4.

At least thirty eight people are feared dead and more than fifteen are missing after a passenger train jumped the tracks as it crossed a bridge in the American state of Alabama. Part of the train plunged into a swamp inhabited by alligators and one carriage is completely submerged. The accident happened in the early hours of the morning in a remote area north of Mobile. The train with more than two hundred people on board, had been en route from Los Angeles to Miami. Up to ten British people thought to be students were on board, but it's thought they were not hurt. Simon Grant, one of the passengers, explained what happened. The water in some parts is believed to be about twenty five feet deep, and rescuers have only been able to reach the area by rail boat or helicopter. A local journalist John Nodar has been following events. It's thought it could be some hours before the scale of the disaster becomes clear. Officials say this is the worst crash in the history of the Amtrak rail company, which was set up in nineteen seventy to run America's long distance services.

At least sixteen people are reported to have died in a collision between two trains in India. More than sixty passengers were injured, some seriously. It happened in the state of Rajahstan, south of Delhi. Officials suggested that a goods train had gone through a stop signal before ploughing into a passenger train.

Divers are continuing to search for three people still believed missing from an express train which plunged off a bridge into a swamp yesterday in the American state of Alabama. Forty four bodies have been recovered from the creek near near Mobile. Visibility underwater is said to be no more than two feet. Those not yet accounted for are the crew, who were in the first of three locomotives. There was thick fog at the time of the accident, and investigators believe a barge collided with the bridge, weakening one of the concrete supports.

At least fourteen people have been killed and seventy injured in a train crash in Morocco. A crowded passenger train was engulfed in flames when it was hit by a train loaded with propane gas.

Police in Northern France have charged a sixteen year old schoolboy with causing a train crash in December in which four people died. He's said to have admitted placing a large piece of metal on the line causing a collision between an early morning commuter train and an oncoming locomotive. The boy is alleged to have told police that he did it to see what would happen. If convicted, he faces a possible life sentence.

Appendix C

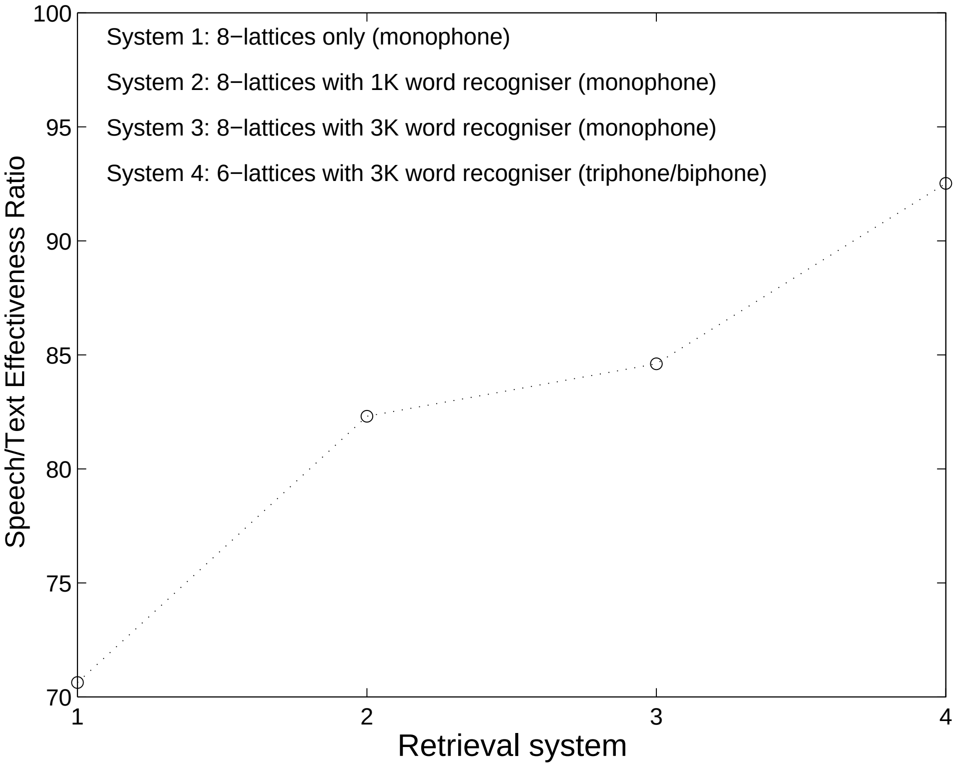


Figure 8.1: The effectiveness ratios for each main retrieval system.

Bibliography

[1] M. Lottor. Internet domain survey. gopher://is.internic.net/00/infoguide/about–internet/domain–surveys/, July 1994.

[2] T. J. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollerman. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 2(1):52–58, Spring 1992.

[3] B. D. Sheth. A Learning Approach to Personalized Information Filtering. Master's thesis, Massachusetts Institute of Technology, 1994.

[4] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[5] P. Willett. *Document Retrieval Systems*, volume 3. Taylor Graham, 1988.

[6] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[7] U. Glavitsch and P. Schäuble. A System for Retrieving Speech Documents. In *Proc. SIGIR*, pages 168–176. ACM, 1992.

[8] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–295, San Mateo, California, 1988. Morgan Kaufmann Publishers Inc.

[9] K.-F. Lee. *Automatic Speech Recognition – The Development of the SPHINX System*. Kluwer Academic Publishers, Boston/Dordrecht/London, 1989.

[10] S. J. Young, P. C. Woodland, and W. J. Byrne. *HTK: Hidden Markov Model Toolkit V1.5*. Entropic Research Laboratories, Inc., 600 Pennsylvania Ave. SE, Suite 202, Washington, DC 20003 USA, 1993.

[11] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, (IT–13):260–269, 1967.

[12] S. J. Young, N. H. Russell, and J. H. S. Thornton. Token Passing: a Conceptual Model for Connected Speech Recognition Systems. Technical report, Cambridge University Engineering Department F.INFENG/TR38, 1989.

[13] D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, and M. A. Przybocki. 1993 Benchmark Tasks for the ARPA Spoken Language Program. In *Proceedings of the Human Language Technology (HLT) Conference*, pages 49–74. ARPA, 1994.

[14] J. N. Holmes. *Speech Synthesis and Recognition*. Van Nostrand Reinhold, Wokingham, Berkshire, England, 1988.

[15] M. M. Hochberg. *A Comparison of State-Duration Modeling Techniques for Connected Speech Recognition*. PhD thesis, Brown University, 1993.

[16] G. Salton. *Automatic indexing and abstracting*, pages 76–114. Prentice–Hall, 1975.

[17] D. K. Harman, editor. *The First Text REtrieval Conference (TREC–1)*. National Institute of Standards and Technology, Gaithersburg MD, 1993.

[18] D. K. Harman, editor. *The Second Text REtrieval Conference (TREC–2)*. National Institute of Standards and Technology, Gaithersburg MD, 1994.

[19] K. Sparck Jones. Reflections on TREC. Technical Report 347, University of Cambridge Computer Laboratory, August 1994.

[20] C. Cleverdon. Optimizing convenient online access to bibliographic databases. In P. Willett, editor, *Document Retrieval Systems*, pages 32–41. Taylor Graham, 1988.

[21] C. D. Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management*, 26(1):171–186, 1990.

[22] D. D. Lewis and K. Sparck Jones. Natural Language Processing for Information Retrieval. Technical Report 307, University of Cambridge Computer Laboratory, July 1993.

[23] S. E. Robertson. The methodology of information retrieval experiment. In K. Sparck Jones, editor, *Information Retrieval Experiment*, pages 9–31. Butterworths, 1981.

[24] K. Sparck Jones and J. I. Tait. Automatic Search Term Variant Generation. *Journal of Documentation*, 40(1):50–66, March 1984.

[25] T. Strzalkowski. Document Representation in Natural Language Text Retrieval. In *Proceedings of the Human Language Technology (HLT) Conference*, pages 364–369. ARPA, 1994.

[26] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

[27] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[28] M. F. Porter. Information retrieval at the Sedgwick Museum. *Information Technology: Research and Development*, 2:169–186, 1983.

[29] J. Kupiec, D. Kimber, and V. Balasubramanian. Speech-based Retrieval using Semantic Co-Occurrence Filtering. In *Proceedings of the Human Language Technology (HLT) Conference*, pages 373–377. ARPA, 1994.

[30] J. M. Tague. The pragmatics of information retrieval experimentation. In K. Sparck Jones, editor, *Information Retrieval Experiment*, pages 59–102. Butterworths, 1981.

[31] K. Sparck Jones. The Cranfield tests. In *Information Retrieval Experiment*, pages 256–284. Butterworths, 1981.

[32] K. Sparck Jones. A Statistical Interpretation of Term Specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[33] C. T. Yu and G. Salton. Effective information retrieval using term accuracy. *Communications of the ACM*, 20:135–142, 1977.

[34] G. Salton. *The SMART Retrieval System*. Prentice–Hall, 1971.

[35] G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[36] S. Robertson and S. Walker. Some Simple Effective Approximations to the 2–Poisson Model for Probabilistic Weighted Retrieval. In *Proc. SIGIR*, pages 232–241, Dublin, 1994.

[37] S. E. Robertson and K. Sparck Jones. Simple, Proven Approaches to Text Retrieval. Technical Report 356, University of Cambridge Computer Laboratory, December 1994.

[38] J. J. Rocchio Jnr. *Document retrieval systems — optimization and evaluation*. PhD thesis, Harvard University, 1966.

[39] G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[40] S. E. Robertson. On Term Selection for Query Expansion. *Journal of Documentation*, 46:359–364, 1990.

[41] P. J. Hayes and S. P. Weinstein. CONSTRUE/TIS: A System for Content–Based Indexing of a Database of News Stories. In *Proceedings of Second Annual Conference on Innovative Applications of Artificial Intelligence*, Georgetown University, Washington DC, 1990.

[42] S. Robertson and K. Sparck Jones. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[43] K. Sparck Jones. Search Term Relevance Weighting Given Little Relevance Information. *Journal of Documentation*, 35(1):30–48, March 1979.

[44] W. B. Croft and D. J. Harper. Using Probabilistic Weights of Document Retrieval without Relevance Information. *Journal of Documentation*, 35(4):285–295, December 1979.

[45] H. P. Frei, S. Meienberg, and P. Schäuble. The Perils of Interpreting Recall and Precision Values. In *Proceedings GI/GMD-Workshop Information Retrieval*, pages 1–10, Berlin, 1991. Springer-Verlag.

[46] G. Salton. TREC. Software distribution.

[47] J. McDonough, K. Ng, P. Jeanrenaud, H. Gish, and J. R. Rohlicek. Approaches to Topic Identification on the Switchboard Corpus. In *Proc. Int. Conf. Acoust., Speech., Sig. Processing*, volume I, pages 385–388, Adelaide, 1994. IEEE.

[48] R. C. Rose, E. I. Chang, and R. P. Lippmann. Techniques for Information Retrieval from Voice Messages. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, pages 317–320, Toronto, 1991. IEEE.

[49] R. C. Rose. Techniques for Information Retrieval from Speech Messages. *Lincoln Laboratory Journal*, 4(1):45–60, 1991.

[50] P. Schäuble and U. Glavitsch. Assessing the Retrieval Effectiveness of a Speech Retrieval System by Simulating Recognition Errors. In *Proceedings of the Human Language Technology (HLT) Conference*, pages 370–372. ARPA, 1994.

[51] J. G. Wilpon, L. G. Miller, and P. Modi. Improvements and Applications for Key Word Recognition Using Hidden Markov Modeling Techniques. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, pages 309–312, Toronto, 1991. IEEE.

[52] J. Bone. Queen's English rings no bells on US phones. *The Times*, page 1, February 9th 1995.

[53] J. M. Baker. Large Vocabulary Speaker–Adaptive Continuous Speech Recognition Research Overview at Dragon Systems. In *Proc. Eurospeech*, pages 29–32, Genoa, 1991. ESCA.

[54] T. Hornstein. Telephone Voice Interfaces on the Cheap. In *Computer Science Research at UBILAB*, pages 134–146, Zürich, 1994. Union Bank of Switzerland, UVK Informatik.

[55] H. Gish, K. Ng, and J. R. Rohlicek. Secondary Processing using Speech Segments for an HMM Word Spotting System. In *Proc. Int. Conf. Spoken. Lang. Processing*, pages 17–20, Banff, 1992.

[56] M. Weintraub. Keyword–Spotting Using SRI's DECIPHER Large–Vocabulary System. In *Proc. Int. Conf. Acoust., Speech., Sig. Processing*, volume II, pages 463–466, Minneapolis, 1993. IEEE.

[57] R. C. Rose. Discriminant Wordspotting Techniques for Rejecting Non–Vocabulary Utterances in Unconstrained Speech. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, volume II, pages 105–108, San Francisco, 1992. IEEE.

[58] National Institute of Standards and Technology. The Road Rally Word–Spotting Corpora, September 1991. Speech Disc 6-1.1.

[59] R. C. Rose and D. B. Paul. A Hidden Markov Model based Keyword Recognition System. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, pages 129–132, Albuquerque, 1990. IEEE.

[60] R. C. Rose. Definition of Subword Acoustic Units for Wordspotting. In *Proc. Eurospeech*, pages 1049–1052, Berlin, 1993. ESCA.

[61] E. Lleida, J. B. Marino, J. Salavedra, and A. Bonafonte. Syllabic Fillers for Spanish HMM Keyword Spotting. In *Proc. Int. Conf. Spoken. Lang. Processing*, pages 5–8, Banff, 1992.

[62] L. D. Wilcox and M. A. Bush. Hmm–based Wordspotting for Voice Editing and Indexing. In *Proc. Eurospeech*, pages 25–28, Genoa, 1991. ESCA.

[63] L. D. Wilcox and M. A. Bush. Training and Search Algorithms for an Interactive Wordspotting System. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, volume II, pages 97–100. IEEE, 1992.

[64] P. Jeanrenaud, K. Ng, M. Siu, J. R. Rohlicek, and H. Gish. Phonetic–Based Word Spotter: Various Configurations and Application to Event Spotting. In *Proc. Eurospeech*, pages 1057–1060, Berlin, 1993. ESCA.

[65] R. C. Rose and E. M. Hofstetter. Techniques for Robust Word Spotting in Continuous Speech Messages. In *Proc. Eurospeech*, pages 1183–1186, Genoa, 1991. ESCA.

[66] J. K. Barkla. Construction of Weighted Term Profiles by Measuring Frequency and Specificity in Relevant Items. In *Proc. Second Int. Cranfield Conf. Mechanized Information Storage and Retrieval Systems*, Cranfield, Bedford, 1969.

[67] W. L. Miller. *The Evaluation of Large Information Retrieval Systems with Application to Medlars.* PhD thesis, University of Newcastle, 1970.

[68] S. Wray, T. Glauert, and A. Hopper. The Medusa Applications Environment. In *Proc. Int. Conf. Multimedia Computing and Systems*, pages 265–273, Boston, May 1994. IEEE.

[69] G. J. F. Jones, J. T. Foote, K. Sparck Jones, and S. J. Young. Video Mail Retrieval: The Effect of Word Spotting Accuracy on Precision. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, Detroit, 1995. IEEE.

[70] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett. The DARPA 1000–Word Resource Management Database for Continuous Speech Recognition. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, New York City, 1988. IEEE.

[71] L. F. Lamel, H. K. Kassel, and S. Seneff. Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 26–32, 1986.

[72] G. J. F. Jones, J. T. Foote, K. Sparck Jones, and S. J. Young. Video Mail Retrieval Using Voice: Report on keyword definition and data collection. Technical Report 335, Cambridge University Computer Laboratory, May 1994.

[73] BBC TV and Radio programme schedules. http://www.bbcnc.org.uk/bbctv/.

[74] L. F. Lamel and J. L. Gauvain. High Performance Speaker–Independent Phone Recognition Using CDHMM. In *Proc. Eurospeech*, pages 121–124, Berlin, 1993. ESCA.

[75] A Computer–Usable Dictionary file based on the Oxford Advanced Learner's Dictionary of Current English. ftp://ota.ox.ac.uk/pub/ota/public/dicts/710/, June 1992.

[76] A. S. Hornby. *Oxford Advanced Learner's Dictionary of Current English, Third Edition*. Oxford University Press, 1974.

[77] T. Robinson, M. Hochberg, and S. Renals. IPA: Improved Phone Modelling with Recurrent Neural Networks. In *Proc. Int. Conf. Acoust., Speech., Sig. Processing*, volume I, pages 37–40, Adelaide, 1994. IEEE.

[78] H. Lucke. *On the Representation of Temporal Data for Connectionist Word Recognition*. PhD thesis, University of Cambridge, 1992.

[79] D. A. James and S. J. Young. A Fast Lattice-based Approach to Vocabulary Independent Wordspotting. In *Proc. Int. Conf. Acoust., Speech., Sig. Processing*, volume I, pages 377–380, Adelaide, 1994. IEEE.

[80] P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, and S. J. Young. The 1994 HTK Large Vocabulary Speech Recognition System. In *Proc. Int. Conf. Acoust., Speech, Sig. Processing*, Detroit, 1995. IEEE.

[81] Bowker Saur. *The Independent on CD–ROM: 1 October 1989 – 31 December 1990*, 1991.

[82] F. Ryan. Searching The Times, The Guardian and The Independent on CD–ROM. *Program*, 25(4):319–337, October 1991.

[83] S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. Acoust., Speech and Sig. Processing*, ASSP–35(3):400–401, 1987.

[84] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny. Context dependent modeling of phones in continuous speech using decision trees. In *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1991.

[85] S. J. Young, J. J. Odell, and P. C. Woodland. Tree–Based State Tying For High Accuracy Acoustic Modelling. In *Proceedings of the Human Language Technology (HLT) Conference*, pages 307–312. ARPA, 1994.

[86] S. J. Young and P. C. Woodland. The use of State Tying in Continuous Speech Recognition. In *Proc. Eurospeech*, pages 2203–2206, Berlin, 1993. ESCA.

[87] J. J. Odell. Private Communication.

[88] G. Salton, A. Wong, and C. T. Yu. Automatic Indexing using Term Discrimination and Term Precision Measurements. *Information Processing & Management*, 12:43–51, 1976.

[89] R. K. Moore. Private Communication.