# Subword-based Approaches for Spoken Document Retrieval

by

Kenney Ng

S.B., Massachusetts Institute of Technology (1990)
S.M., Massachusetts Institute of Technology (1990)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000



Author .....
Department of Electrical Engineering and Computer Science
January 12, 2000

Certified by .....
Victor W. Zue
Senior Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

**Subword-based Approaches for Spoken Document Retrieval**

by

Kenney Ng

Submitted to the Department of Electrical Engineering and Computer Science
on January 12, 2000, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

## Abstract

This thesis explores approaches to the problem of spoken document retrieval (SDR), which is the task of automatically indexing and then retrieving relevant items from a large collection of recorded speech messages in response to a user specified natural language text query. We investigate the use of subword unit representations for SDR as an alternative to words generated by either keyword spotting or continuous speech recognition. Our investigation is motivated by the observation that word-based retrieval approaches face the problem of either having to know the keywords to search for *a priori*, or requiring a very large recognition vocabulary in order to cover the contents of growing and diverse message collections. The use of subword units in the recognizer constrains the size of the vocabulary needed to cover the language; and the use of subword units as indexing terms allows for the detection of new user-specified query terms during retrieval.

Four research issues are addressed. First, what are suitable subword units and how well can they perform? Second, how can these units be reliably extracted from the speech signal? Third, what is the behavior of the subword units when there are speech recognition errors and how well do they perform? And fourth, how can the indexing and retrieval methods be modified to take into account the fact that the speech recognition output will be errorful?

We first explore a range of subword units of varying complexity derived from error-free phonetic transcriptions and measure their ability to effectively index and retrieve speech messages. We find that many subword units capture enough information to perform effective retrieval and that it is possible to achieve performance comparable to that of text-based word units. Next, we develop a phonetic speech recognizer and process the spoken document collection to generate phonetic transcriptions. We then measure the ability of subword units derived from these transcriptions to perform spoken document retrieval and examine the effects of recognition errors on retrieval performance. Retrieval performance degrades for all subword units (to 60% of the clean reference), but remains reasonable for some subword units even without the use of any error compensation techniques. We then investigate a number of robust methods that take into account the characteristics of the recognition errors and try to compensate for them in an effort to improve spoken document retrieval performance when there are speech recognition errors. We study the methods individually and explore the effects of combining them. Using these robust methods improves retrieval performance by 23%. We also propose a novel approach to SDR where the speech recognition and information retrieval components are more tightly integrated. This is accomplished by developing new recognizer and retrieval models where the interface between the two

components is better matched and the goals of the two components are consistent with each other and with the overall goal of the combined system. Using this new integrated approach improves retrieval performance by 28%. We also detail the development of our novel probabilistic retrieval model and separately evaluate its performance using standard text retrieval tasks. Experimental results indicate that our retrieval model is able to achieve state-of-the-art performance.

In this thesis, we make the following four contributions to research in the area of spoken document retrieval: 1) an empirical study of the ability of different subword units to perform document retrieval and their behavior and performance in the presence of recognition errors; 2) the development of a number of robust indexing and retrieval methods that can improve retrieval performance when there are recognition errors; 3) the development of a novel spoken document retrieval approach with a tighter coupling between the recognition and retrieval components that results in improved retrieval performance when there are recognition errors; and 4) the development of a novel probabilistic information retrieval model that achieves state-of-the-art performance on standardized text retrieval tasks.

Thesis Supervisor: Victor W. Zue
Title: Senior Research Scientist

## Acknowledgments

I would like to thank my thesis advisor, Victor Zue, for his guidance throughout this research project. I highly valued the direction and insight he provided for my research and the comments and feedback he gave on my papers. I also thank Victor for assembling an excellent team of researchers that make up the Spoken Language Systems (SLS) group, fostering an exciting atmosphere for research, and ensuring that the necessary financial and computing resources were available so the students could be free to focus on academics and research and not have to worry about other things.

I also want to thank my entire thesis committee, consisting of Victor Zue, Jim Glass, David Karger, and Herb Gish for their time and interest. They provided valuable advice and suggestions in our discussion meetings and helpful comments on drafts of the written thesis. I also want to thank Lou Braida for being my academic advisor.

I was fortunate enough to be a teaching assistant for two really great courses: 6.011 (Introduction to Communication, Control, and Signal Processing) and 6.345 (Automatic Speech Recognition). I want to thank George Verghese for letting me be a TA for 6.011 and Victor Zue and Jim Glass for letting me TA 6.345.

Everyone in the SLS group has been very helpful throughout my four plus years here and I am very appreciative of that. In particular, I want to thank Michelle Spina for her help in putting together the NPR data corpus that I used in my research and Drew Halberstadt and Ray Lau for being terrific officemates.

Finally I would like to thank my wife, Carissa, and my entire family for their love and encouragement throughout my graduate career. Returning to graduate school after being away for five years was not easy and I could not have done it without their support.

This research was supported by research contracts from NYNEX Science and Technology and DARPA under contracts N66001-96-C-8526 and N66001-99-1-8904, monitored through Naval Command, Control and Ocean Surveillance Center

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

With the explosion in the amount of accessible data spurred on by advances in information technologies including increasingly powerful computers, increased data storage capacity, and growing international information infrastructures (e.g., the Internet), the need for automatic methods to process, organize, and analyze this data and present it in human usable form has become increasingly important. Of particular interest is the problem of efficiently finding and selecting "interesting" pieces of information from among the rapidly growing streams and collections of data. This is especially true as more and more people seek to make effective use of these vast sources of information on a routine basis.

The World Wide Web is a good example of this scenario. There is so much data available on the Web that the only way someone can even hope to find the information that he or she is interested in is to rely on automatic methods such as web search engines. Although these automatic methods are very popular and extremely useful, they are still far from being perfect. The tasks of automatically indexing, organizing, and retrieving collections of information items are still open research problems.

Much research has been done, under the headings of document and text retrieval, on the problem of selecting "relevant" items from a large collection of *text* documents given a request from a user (Harman 1997; Rijsbergen 1979; Salton and McGill 1983). Only recently has there been work addressing the retrieval of information from other media such as images, audio, video, and speech (Dharanipragada et al. 1998; Foote et al. 1995; Hauptmann and Wactlar 1997; James 1995; Wechsler and Schauble 1995; Witbrock and Hauptmann 1997).

This expansion into other media raises new and interesting research issues in information retrieval, especially in the areas of content representation and performance robustness.

Given that increasingly large portions of the available data contain spoken language information, such as recorded speech messages, radio broadcasts, and television broadcasts, the development of automatic methods to index and retrieve spoken documents will become more important. In addition, the development of these methods will have a significant impact on the use of speech as a data type because speech is currently a very difficult medium for people to browse and search efficiently (Schmandt 1994).

In this chapter, we first provide some background by giving an introduction to information retrieval. Next, we describe some of the differences between text and speech media and list some of the issues raised by the change in media. Then, we briefly describe some speech information processing tasks related to speech retrieval including topic identification, spotting, and clustering. We then motivate our research and describe the goals and contributions of this thesis. Finally, we give a chapter by chapter overview of the thesis.

## 1.1 Information Retrieval

The term "information retrieval" has been used to describe a wide area of research that is "concerned with the representation, storage, organization, and accessing of information items" (Salton and McGill 1983). The typical scenario associated with information retrieval is that of identifying information items or "documents" within a large collection that best match a "request" provided by a user to describe his or her information need. The user is not looking for a specific fact but is interested in a general topic or subject area and wants to find out more about it. In other words, the request is usually an incomplete specification of the user's information need. An example would be requesting articles about "the blizzard of 1996" from a collection of newspaper articles. The goal is not to return specific facts in answer to the user's request but to inform the user of the existence of documents in the collection that are relevant to his or her request and to return pointers or references to them. This can also include the degree to which the identified documents match the request.

There is no restriction, in principle, on the type of document that can be handled; it can

be a text document, an image, an audio recording, or a speech message. In practice, however, most of the work on automatic information retrieval to date has dealt with collections of text documents ranging in complexity from bibliographical data and abstracts to complete articles and books. Only recently has there been work with other types of media. In the literature, the terms "text retrieval" and "document retrieval" are used to refer to text media; "image retrieval" and "speech retrieval" are used to refer to image and speech media respectively; and "multi-media retrieval" is used to refer to mixed media. In this chapter, we will use the term "document" to refer to items of any media type.

The notion of "best match" and "relevance" is purposely left vague at this point. It will be discussed in more detail when we examine specific retrieval methods in Section 2.4 and Chapter 6. It is generally assumed, however, that documents that match the request are those that are about the same or similar "topic" as the request. This means that the measure of relevance is based, in some way, on the *contents* of the document and request.

### 1.1.1 Information Retrieval Components

All information retrieval systems have the following basic component processes which is illustrated in the system block diagram in Figure 1-1:

- Creating the document representations (indexing),
- Creating the request representation (query formation), and
- Comparing the query and document representations (retrieval).

Each document, upon addition to the collection, needs to be processed to obtain a document representation that is stored and used by the retrieval system. This process is known as *indexing*. The representation must capture the important information contained in the original document in a form that allows it to be compared against representations of other documents and representations of the user requests or queries. Typically, the representation is more compact than the original document which allows for large collections of documents to be handled efficiently. Each user request must also be processed to generate a request representation or "query." This process is known as *query formation*. As with the document representations, the query must be able to capture the important information contained in

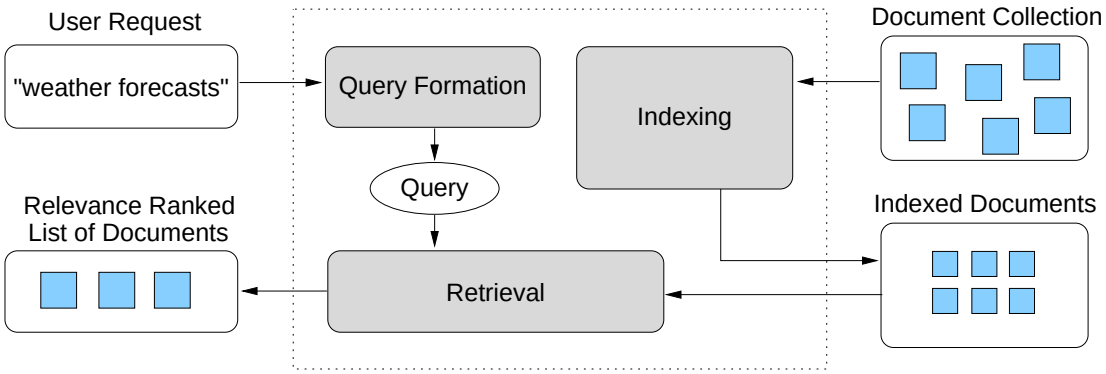


Figure 1-1: Block diagram illustrating the major components in an information retrieval system. The indexing process creates the document representations; the query formation process turns the user request into a query; and the retrieval component compares the query and document representations and returns a relevance ranked list of documents.

the original request in a form that allows it to be compared against the document representations. The query is the expression of the user's information need and is used by the retrieval system to select documents from the collection.

The central process of an information retrieval system is the comparison of the query representation with the document representations. This is the *retrieval* process. In general, a matching function is defined which selects relevant documents from the collection based on the query and document representations. In principle, each document in the collection is matched against the query to determine its relevance.

Information retrieval performance can be measured along many dimensions. In real-world applications, factors such as cost of implementation and maintenance, ease of indexing new documents, and speed of retrieval are important. By far the most popular performance criteria is that of retrieval effectiveness which is usually composed of two measures: *recall* and *precision*. Recall is the fraction of all the relevant documents in the entire collection that are retrieved in response to a query. Precision is the fraction of the retrieved documents that are relevant. By scanning down the list of ranked documents, a precision-recall curve can be traced out. This graph indicates the set of possible operating points that can be obtained by thresholding the list of ranked documents at various points. Recall and precision generally vary inversely with each other. Section 2.4 presents a more detailed description of the information retrieval performance measures that we will use in this thesis.

| Information Process | Goal | Documents | Queries | Training Data |
|---|---|---|---|---|
| **Information Retrieval** | search for and retrieve documents in a collection that are relevant to a user query | static, unstructured | dynamic, incomplete | no |
| **Database Retrieval** | search a database of records and return specific facts or particular records that answer a user request | static, structured | dynamic, complete | no |
| **Information Filtering** | identify relevant documents from incoming streams of documents | dynamic, unstructured | static, incomplete | yes |
| **Document Categorization** | classify documents into one or more predefined categories | unstructured | no queries, known categories | yes |
| **Document Clustering** | automatically discover structure in a collection of unlabelled documents | static, unstructured | no queries, unknown categories | no |
| **Information Extraction** | automatically find and extract domain specific features or facts | unstructured | no queries | yes |
| **Document Summarization** | automatically derive a concise meaning representation of the document | unstructured | no queries | no |

Table 1-1: List of information processes related to information retrieval. A brief description of their goals and some characteristics regarding the document collection, the topics or queries, and an indication of the availability of labelled training data are included.

### 1.1.2 Related Information Processes

There are many information processing tasks that are closely related to information retrieval. These include database retrieval, information filtering/routing, document categorization, document clustering, information extraction, and document summarization. These processes are listed in Table 1-1 along with a brief description of their goals and some characteristics regarding the document collection, the topics or queries, and whether labelled training data is available. In this section, we briefly describe these processes and mention their similarities and differences to retrieval.

Recall that in **information retrieval** the goal is to search for and retrieve documents in a collection that are relevant to a user's request. This task is characterized by a dynamic information need in the sense that user requests can change from session to session and it is not known *a priori* what the user will ask. The request is also usually an incomplete (imprecise) specification of the user's information need. Another characteristic is that the collection of documents is relatively static. There may be additions to and deletions from

the collection but they generally have a very small effect on the entire collection. In addition, no supervised training data is available during collection creation since it is not known what the user requests will be and, hence, which documents are relevant and which are not.

In **database retrieval**, the goal is to search a database of records and return specific facts or particular records that answer or exactly match a given user request. The structure of the records is usually well defined (i.e., the records consist of specific fields filled with particular types of values such as zip codes, dates, etc.) and the request is a complete (precise) specification of the user's information need.

In **Information filtering/routing**, the goal is to identify relevant documents from incoming streams of documents. The filtering is usually based on descriptions of long-term information preferences called "profiles" instead of dynamic queries. Documents that match are then forwarded or routed to the users associated with the profile; those that don't match are discarded. Since the information need is relatively static, documents that have been processed and assessed by the user can serve as training data to improve the profile.

The goal in **document categorization** is to classify documents into one or more of a set of predefined categories. The documents can be in a static collection or can arrive in a data stream (e.g., newswire). There is usually a set of labeled data (document:category pairs) that can be used to train classifiers for the different categories.

In **document clustering**, there is no labeled training data and the goal is to automatically discover structure in a collection of unlabelled documents. Clustering has been used to organize document collections to improve efficiency and performance for information retrieval (Croft 1980; Lewis 1992; Rijsbergen 1979).

In **information extraction**, the goal is to automatically find and extract domain specific features or facts like entities, attributes, and relationships from a document (Chinchor and Sundheim 1995). Some examples of the types of information usually extracted are names of organizations, people, locations, and dates.

The goal in **document summarization** is to automatically derive a representation of the document that captures it's important characteristics succinctly. There has been some work in trying to automatically derive abstracts of text documents for use in information retrieval (Kupiec et al. 1995).

## 1.2 Text vs. Speech Media

There are many differences between text and speech media which raise new research issues that need to be addressed in order to be able to develop an effective information retrieval system for speech messages. Similar issues arise when comparing text and image media.

One issue is that speech is a richer and more expressive medium than text (Schmandt 1994); it contains more information than just the words. With speech, information such as the identity of the spoken language, the identity of the speaker, and the "mood" or "tone" of the speaker, expressed as prosodic cues, are captured in addition to the spoken words. This additional information may be useful in extending a retrieval system; it offers new indexing features. One type of information that remains common to both text and speech documents is the concept of the topic or subject of the document. In this work, we only deal with the topic content of the documents; the use of the other acoustic information contained in the speech signal is beyond the scope of this work.

A second issue is how to accurately extract and represent the contents of a speech message in a form that can be efficiently stored and searched. Although a similar task needs to be done with text documents, the change from text to speech adds an additional layer of complexity and uncertainty. There are many challenges including being able to handle multiple speakers, noisy speech, conversational or fluent speech, and very large (even potentially unlimited) vocabularies. In this work, we investigate the use of subword unit representations as an alternative to word units generated by either keyword spotting or continuous speech recognition. The subword unit indexing terms are created by post-processing the output of a phonetic speech recognizer.

A third issue is the robustness of the retrieval models to noise or errors in transcription. Most of the indexing and retrieval methods that have been developed for text documents have implicitly assumed error-free transcriptions. With text, the words in the documents are assumed to be known with certainty. As a result, there is no explicit mechanism in the models for dealing with errors in the document representations. However, with speech there are currently no perfect automatic transcription methods and there will likely be errors in the transcripts generated by the speech recognizer. The use of dictionaries to spell check

words, thesauri to expand words, and the conflation of words to their stems can all be thought of as approaches that address the issue of robustness to varying degrees. Modifying the retrieval model by generalizing the matching function to allow for the approximate matching of indexing terms is another approach to deal with errorful document representations. In Chapter 5, we investigate a number of robust methods that take into account the characteristics of the recognition errors and try to compensate for them in an effort to improve retrieval performance. We also propose, in Chapter 7, a novel retrieval approach where the recognition and retrieval components are more tightly integrated.

## 1.3 Related Speech Information Processes

Spoken document retrieval is a relatively new area of work not only in the information retrieval community but also in the speech recognition community. However, there has been work done in recent years on the related problems of classifying and sorting speech messages according to subject or topic. These tasks, known as "topic identification" and "topic spotting," are analogous to categorization and routing for text documents. Speech messages, like text documents, can also be automatically clustered to try to discover structure or relationships between messages in a collection.

The task in topic identification is the assignment of the correct topic to a speech message known to be about one of a fixed number of possible topics. This is a "closed set" problem in the sense that the speech message must belong to one of the prespecified topics. Topic spotting is the "open set" variant of this problem. In this case, it is possible for the speech message to not belong to any of the prespecified topics, i.e., it can belong to "none of the above." The term "topic spotting," however, has been used almost exclusively to refer to the two-class discrimination problem where the task is to decide if a speech message is about a certain specified topic or not. Two characteristics of the topic identification and spotting work that make them different from retrieval are the *a priori* specification of the topic classes, and the availability of labeled speech messages for training the classifiers and statistical models used to score the messages. In retrieval, there is no training data and the user queries (i.e., topics) are not known ahead of time. It is only during the relevance

feedback stage that labeled messages (relevant or not relevant) are available. In clustering, there is no labeled training data and the goal is to automatically discover structure in a collection of unlabelled data. Clustering information may be useful directly to a user (e.g., for browsing a document collection) or can be used to improve retrieval system performance by providing some structural organization to the documents in the collection.

Because of the similarities of topic identification, spotting, and clustering to speech retrieval, many of the techniques developed for these speech information processing tasks can be adapted for use in spoken document retrieval. We now briefly review some of the approaches developed for these related speech information processing tasks.

**Topic Identification**

In (Rose et al. 1991), a word spotting approach is used to detect keyword events in a 6 topic speech message classification task. The set of keywords was determined by computing the mutual information between each word and each topic on a set of training messages labeled according to topic, and then selecting the top scoring words per topic. These keywords were input to a neural network based message classifier that takes as input a binary feature vector representing the presence or absence of each keyword in the message and outputs a score for each topic class. Topic identification is performed for a given speech message by finding that topic which maximizes the score. A classification accuracy of 82.4% was achieved with text transcriptions of the speech messages. When a word spotter was used to detect the keywords in the speech message, topic classification accuracy dropped to 50%. Replacing the binary feature vector to the message classifier by a continuous valued vector consisting of the keyword detection confidence measure improved performance to 62.4%.

In (Rohlicek et al. 1992), a system for extracting information from off-the-air recordings of air traffic control communications is described. The goal of this "gisting" system is to identify flights and to determine whether they are "taking off" or "landing." Two different methods were used to perform the topic classification. One is based on training a decision tree classifier (Brieman et al. 1984) to distinguish between the "take off" and "landing" topics. The input to the classifier consist of binary feature vectors with each element indicating the presence or absence of the corresponding keyword in the dialog. The second

method performs topic classification by accumulating likelihood ratio scores of the detected keywords to produce a topic score. In both approaches, the training data consists of a large collection of labeled dialogs that have been processed by the recognizer. Performance is high on this task with correct classification of transmissions and dialogs above 90%.

In (Gillick et al. 1993), a large vocabulary word recognition approach was used on a 10-class topic identification task of recorded telephone conversations from the Switchboard corpus (Godfrey et al. 1992). To classify a speech message, a large vocabulary, speaker-independent, speech recognizer is first used to produce an errorful transcription of the speech message. Then scores for each of the ten topics are generated by evaluating the transcription on unigram language models that have been trained for each topic. Finally, the message is assigned to the topic corresponding to the model with the best score. Experiments showed that even with highly errorful transcriptions produced by the word recognizer (78% word error), reasonable topic identification performance (74% accuracy) can be achieved.

A comparison of using both word recognition and word spotting on the same 10-class topic identification task from the Switchboard corpus was done in (McDonough et al. 1994; McDonough and Gish 1994). In this study, a large vocabulary, speaker-independent, speech recognizer was run in both recognition mode, to generate the most likely word sequence, and in word spotting mode, to generate putative word events with an associated posterior probability of occurrence score. Different message representations are created from the recognition and word spotting outputs. Each element in the recognizer feature vector contains the number of times the corresponding word is seen in the recognition output for the message. In the word spotting feature vector, each element contains the expected number of occurrences obtained by summing the probability scores associated with each putative word hypothesis. These features are used as input to a topic classifier based on a multinomial model of the keyword occurrences. Experiments show that the word spotter feature vector (79.2% accuracy) out-performed the recognizer feature vector (74.6% accuracy).

### Topic Spotting

In (Carey and Parris 1995), a small vocabulary word spotter was used to detect or spot weather reports from recorded BBC radio news broadcasts. The set of keywords used in the

word spotter were selected from labeled training data by computing a "usefulness" measure based on the mutual information between each word and each topic. Topic spotting is performed by accumulating, over a window of speech (typically 60 seconds), the usefulness scores of the detected keywords to produce a topic score for that region of the speech message; regions with high scores are then hypothesized to be about the topic. In (Wright et al. 1995) an alternative method of scoring is presented in which the *occurrence distribution* of the keyword in topic and non-topic speech is modeled instead of just the *occurrence probabilities*. These new distributions can each be modeled by a Poisson or a mixture of Poissons. Also in (Wright et al. 1995), several different topic models such as logistic and log-linear models that try to capture dependencies between the keywords are examined. Experiments show that using a carefully chosen log-linear model can give topic spotting performance that is better than using the basic model that assumes keyword independence.

There have also been subword approaches to topic spotting (Nowell and Moore 1994; Skilling et al. 1995; Wright et al. 1996). The main motivation in these approaches is to try to require as little prior knowledge about the domain as possible. In (Nowell and Moore 1994), a dynamic programming approach was used to select variable length phone sequences generated by a phone recognizer to be used as "keywords" for spotting topics in recorded military radio broadcasts. This approach was extended to the acoustic level in (Skilling et al. 1995), where sequences of vector-quantized acoustic features, instead of phone sequences, are used as the "keywords" for topic spotting. In both of these approaches, "keywords" sets are selected by computing and selecting those with a high "usefulness" measure. Again, topic spotting is performed by accumulating the usefulness scores of the detected "keywords" over a window to produce a topic score for that region of the speech message; regions with high scores are then hypothesized to be about the topic. For the particular task used in these experiments, both of these subword approaches gave reasonable topic spotting performance.

### Topic Clustering

In (Carlson 1996), several approaches to the task of automatic clustering of speech messages from the Switchboard corpus by topic are investigated. The clustering process has three main components: tokenization, similarity computation, and clustering. The goal in tok-

enization is to generate a suitable representation of the speech message which can be used by the other two components. Examples investigated include words from text transcriptions, words generated by a word recognizer, and phones generated by a phonetic recognizer. Next, a measure of similarity needs to be computed between every pair of messages. In this work, a similarity measure based on the ratio of likelihood scores derived from $n$-gram models is used. For each message, an $n$-gram language model is first computed; then each message is scored against the language model of the other messages; and finally, these scores are used to compute the similarity measure between the messages. These similarity scores are then used in the third stage to perform clustering. Two different clustering methods are investigated: hierarchical tree clustering and nearest neighbor classification. Experimental results indicate that all methods work well when true transcription texts are used. Performance is significantly worse when using speech input but still reasonable enough to be useful. The best speech clustering performance was obtained with word recognition output, unigram language models, and tree-based clustering.

## 1.4 Motivation

One approach to the task of spoken document retrieval (SDR) is to perform keyword spotting on the spoken documents to obtain a representation in terms of a small set of keywords (Foote et al. 1995; Jones et al. 1995a; Rose et al. 1991). In order to process the speech messages to create the keyword representations ahead of time, the set of keywords needs to be chosen *a priori*. This either requires advanced knowledge about the content of the speech messages or what the possible user queries may be. Alternatively, the keywords can be determined after the user specifies the query. In this case, however, the user would need to wait while the entire message collection is searched. Even with faster than real-time keyword spotting systems, there may be unacceptable delays in the response time.

Another approach is to first transform the spoken documents into text using a large vocabulary speech recognizer and then use a conventional full-text retrieval system (Hauptmann and Wactlar 1997; Johnson et al. 1998; Witbrock and Hauptmann 1997). In this approach, the main research emphasis is on trying to improve the speech recognition sys-

tem so that it can operate efficiently and accurately in a large and diverse domain. This has been the dominant approach in the recent spoken document retrieval tracks of the NIST (National Institute of Standards and Technology) sponsored Text REtrieval Conference (TREC) (Garofolo et al. 1997; Garofolo et al. 1998). Although this approach is straightforward, it has several drawbacks. One is the decoupling of the speech recognition and message retrieval processes. Although this leads to modularity, it can also lead to sub-optimality; the retrieval process is likely to benefit from information about the uncertainty of the recognized words produced during the recognition process. Another important issue is the growth of the recognizer vocabulary needed to handle new words from growing and diverse message collections. With current technology, there is a practical limit on the size of the recognition vocabulary. There are also the related issues of determining when, how, and what new words need to be added and whether the entire message collection needs to be re-indexed when the recognizer vocabulary changes. Yet another issue is the reliance on large amounts of domain-specific data for training the large vocabulary speech recognition models.

To illustrate the nature of the vocabulary growth, two years of text data from the Los Angeles Times newspaper (1989 and 1990) used in the ad-hoc text retrieval task in TREC-6 (Harman 1997) are analyzed. Figure 1-2 plots the relationship between the size of the vocabulary versus the size of the data set as the data set size is increased. We start with a data set consisting of one day's worth of news stories and continue to add stories incrementally, in chronological order, until all two years worth of data has been added. Milestones are indicated as the data set grows to include one day, one week, one month, six months, one year, and finally two years' worth of data. We observe that the vocabulary size grows with the data set size. Even after the data set contains a significant amount of data (e.g., after the one year mark), new vocabulary words are continually encountered as more data is added. Many of the new vocabulary words, not surprisingly, are proper names, and, of course, these are the words that are important for information retrieval purposes.

An alternative approach that has the potential to deal with many of the above problems is to use subword unit representations for spoken document retrieval. The use of subword units in the recognizer constrains the size of the vocabulary needed to cover the language

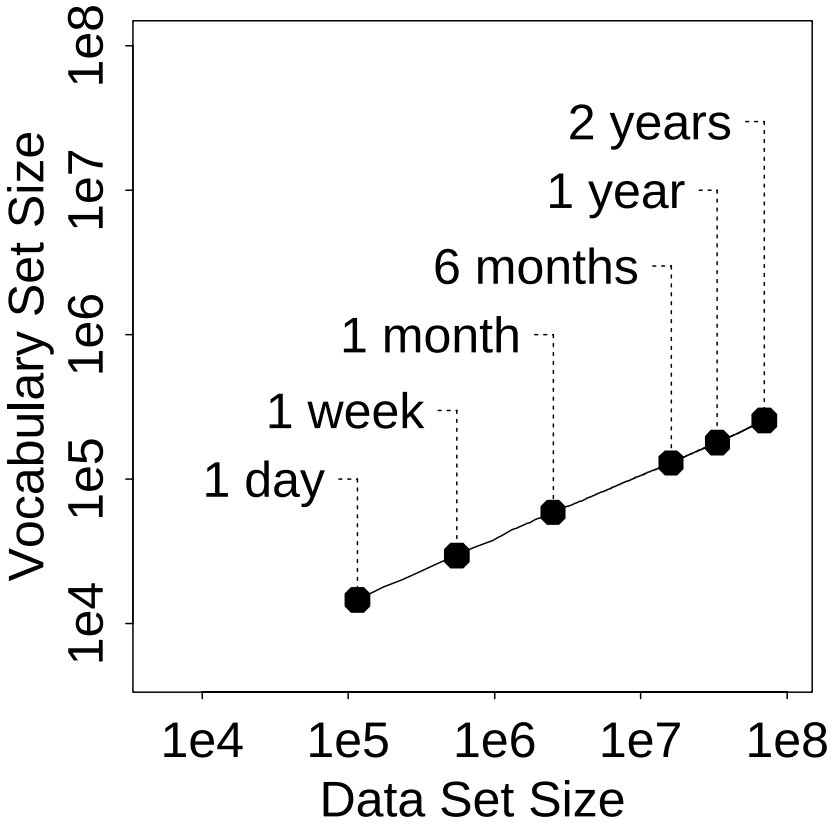


Figure 1-2: The relationship between the growth of the data set size and the vocabulary set size. Two years of text data from the Los Angeles Times newspaper (1989 and 1990) used in the ad-hoc text retrieval task in TREC-6 is used to generate the plot. Milestones are indicated as the data set grows to include 1 day, 1 week, 1 month, 6 months, 1 year, and finally 2 years' worth of data.

and reduces the amount of data needed for training. The reduced training requirements can facilitate the transition to new application domains and different languages. The use of subword unit terms for indexing allows for the detection of new query terms specified by the user during retrieval. Although there is a tradeoff between the size of the subword unit and its recognition accuracy and discrimination capability, some of this can be mitigated by an appropriate choice of subword units and the modeling of their sequential constraints. The effectiveness of relatively simple text retrieval algorithms that essentially match strings of consecutive text characters (i.e., character $n$-grams) (Cavnar 1994; Damashek 1995; Huffman 1995) gives us hope that subword approaches can be successful with spoken doc-

uments. We should note that subword-based approaches can also be used in combination with word-based methods to complement them in situations where it is difficult to train a large vocabulary recognizer or when out-of-vocabulary words occur in the user query.

Several subword based approaches have been proposed in the literature. One makes use of special syllable-like indexing units derived from text (Glavitsch and Schauble 1992; Schäuble and Glavitsch 1994) while others use phone sequences (phone $n$-grams) generated by post-processing the output of a phonetic speech recognizer (Ng and Zobel 1998; Wechsler and Schauble 1995). There are also methods that search for the query terms on phonetic transcriptions or phone lattice representations of the speech messages instead of creating subword indexing terms (Dharanipragada et al. 1998; James 1995; Jones et al. 1996; Wechsler et al. 1998). Some of these methods combine subword and large vocabulary word approaches to address the issue of new words in the queries (James 1995; Jones et al. 1996). However, there have been few studies that explore the space of possible subword unit representations to determine the complexity of the subword units needed to perform effective spoken document retrieval and to measure the behavior and sensitivity of different types of subword units to speech recognition errors.

Although there has been some work in trying to compensate for optical character recognition (OCR) errors introduced into automatically scanned text documents (Marukawa et al. 1997; Zhai et al. 1996), the area of robust methods for dealing with speech recognition errors in the context of spoken document retrieval is still relatively new. There has been some recent work in this area performed independently and in parallel to the work presented in this thesis. In (Ng and Zobel 1998), manual correction and string edit distances are used to try to compensate for phonetic recognition errors. In (Jourlin et al. 1999), a number of different query expansion techniques are used to try to compensate for word recognition errors. In (Singhal et al. 1998), noisy document representations are expanded to include clean words from similar documents obtained from a parallel clean text corpus. In (Wechsler et al. 1998), a keyword spotting technique that allows for phone mismatches is used to detect query terms in the errorful phonetic transcriptions of the spoken documents.

## 1.5 Goals and Contributions

The main goal of this research is to investigate the feasibility of using subword unit representations for spoken document retrieval as an alternative to words generated by either keyword spotting or continuous speech recognition. Four research issues are addressed:

1. What are suitable subword units and how well can they perform?

2. How can these units be reliably extracted from the speech signal?

3. What is the behavior of the subword units when there are speech recognition errors and how well do they perform?

4. How can the indexing and retrieval methods be modified to take into account the fact that the speech recognition output will be errorful?

We first explore a range of subword units of varying complexity derived from error-free phonetic transcriptions and measure their ability to effectively index and retrieve speech messages (Ng and Zue 1997a; Ng and Zue 1997b). Next, we develop a phonetic speech recognizer and process the spoken document collection to generate phonetic transcriptions. We then measure the ability of subword units derived from these transcriptions to perform retrieval and examine the effects of recognition errors on retrieval performance (Ng and Zue 1998). We then investigate a number of robust methods that take into account the characteristics of the recognition errors and try to compensate for them in an effort to improve retrieval performance when there are speech recognition errors; we study the methods individually and explore the effects of combining them (Ng 1998). We also propose a novel approach to SDR where the speech recognition and information retrieval components are more tightly integrated. This is accomplished by developing new recognizer and retrieval models where the interface between the two components is better matched and the goals of the two components are consistent with each other and with the overall goal of the combined system. The novel probabilistic retrieval model that we develop as part of this effort is evaluated separately on standard TREC text retrieval tasks and is able to achieve state-of-the-art performance (Ng 1999).

In this thesis, we make the following contributions to research in the area of spoken document retrieval:

- An empirical study of the ability of different subword units to perform spoken document retrieval and their behavior in the presence of speech recognition errors.
- The development of a number of robust indexing and retrieval methods that can improve retrieval performance when there are speech recognition errors.
- The development of a novel spoken document retrieval approach with a tighter coupling between the recognition and retrieval components that results in improved retrieval performance when there are speech recognition errors.
- The development of a novel probabilistic information retrieval model that achieves state-of-the-art performance on standardized text retrieval tasks.

## 1.6 Overview

The thesis is organized as follows. Chapter 2 contains background information for the experimental work presented in this thesis. This includes information about the various speech and test corpora used in the experiments and descriptions of the SUMMIT speech recognition system and the initial information retrieval model used. In Chapter 3, we explore a range of subword units of varying complexity derived from error-free phonetic transcriptions and measure their ability to effectively index and retrieve speech messages. Next, we train and tune a phonetic recognizer and use it to process the entire spoken document collection to generate phonetic transcriptions in Chapter 4. We then explore a range of subword unit indexing terms of varying complexity derived from these errorful phonetic transcriptions and measure their ability to support spoken document retrieval. In Chapter 5, we investigate a number of robust methods that take into account the characteristics of the recognition errors and try to compensate for them in an effort to improve spoken document retrieval performance when there are speech recognition errors. We study the methods individually and explore the effects of combining them. We take a brief digression from spoken document retrieval in Chapter 6 where we describe the development of a novel probabilistic retrieval

model and evaluate its performance using standard text retrieval tasks. This retrieval model is used in Chapter 7 to develop a novel approach to SDR where the speech recognition and information retrieval components are more tightly integrated. We develop new recognizer and retrieval models where the interface between the two components is better matched and the goals of the two components are consistent with each other and with the overall goal of the combined system. Finally, in Chapter 8, we summarize the work, draw some conclusions, and mention some possible directions for future work.

# Chapter 2

# Experimental Background

This chapter contains background information for the experimental work presented in this thesis. This includes information about the various speech and test corpora used in the experiments and descriptions of the SUMMIT speech recognition system and the initial information retrieval model used.

## 2.1 NPR Speech Corpus

Many of the spoken document retrieval experiments done in this thesis make use of an internally collected data set: the NPR speech corpus. This corpus consists of FM radio broadcasts of the National Public Radio (NPR) "Morning Edition" news program (Spina and Zue 1996). The data is recorded off the air onto digital audio tape, orthographically transcribed, and partitioned into separate news stories. Both the word transcription and story partitioning processes were done manually by professional transcribers. In addition, the transcribers created a short "topic description" for each story. The data is divided into two sets, one for training and tuning the speech recognizer and another for use as the spoken document collection for the information retrieval experiments. Because the speech data consists of different broadcasts of the same radio show, there will be some recurring speakers, such as the studio announcers, from one broadcast to another. As a result, the data should be considered, for speech recognition purposes, as multi-talker instead of speaker-independent.

| **No. of documents** | 384 | | |
|---|---|---|---|
| **No. of topics** | 50 | | |
| | **Min.** | **Mean** | **Max.** |
| **Document length (words)** | 22 | 325 | 663 |
| **Document length (seconds)** | 7.4 | 114.0 | 586.5 |
| **Topic length (words)** | 2 | 4.5 | 11 |
| **No. of relevant docs/topic** | 2 | 6.2 | 35 |

Table 2-1: Statistics for the NPR spoken document collection.

### 2.1.1 Speech recognition data sets

The speech recognition training set consists of 2.5 hours of "clean" (studio quality, without noise, etc.) speech from five one-hour shows with 156 unique speakers. In related work (Spina and Zue 1997), it was found that training on speech from all the different noise conditions in the data (background noise, background music, telephone channel, etc.) does not significantly improve recognition performance over training on only the clean speech. As a result, only clean speech is used for training the speech recognizer in our experiments.

The development set, used to tune and test the recognizer, consists of one hour of data from one show and contains speech from all acoustic conditions. There are 46 unique speakers in this data set, 12 of whom also occur in the training data.

### 2.1.2 Spoken document collection

The spoken document collection consists of over 12 hours of speech from 16 one hour shows partitioned into 384 separate news stories. Filler segments such as commercials and headline introductions which don't belong to a single news story are disregarded. Each news story averages two minutes in duration and typically contains speech from multiple acoustic conditions. There are 462 unique speakers in this data set, 35 of whom also occur in the training data. Statistics for the NPR spoken document collection are shown in Table 2-1.

A set of 50 natural language text queries and associated relevance judgments on the message collection (i.e., the set of relevant documents or "answers") were created to support retrieval experiments. The queries, which resemble story headlines, are relatively short, each averaging 4.5 words. Some example queries are "Whitewater controversy: hearings,

investigations, and trials," "IRA bomb explosion in England," and "Proposal to increase the minimum wage." The queries were created using the manually generated "topic descriptions" for each story. First, the topic descriptions for all stories were compiled and manually categorized into groups that are about the same topic. Then, the 50 topics with the most relevant documents were selected and, for each topic, a representative topic description was chosen to be the corresponding query. The topic descriptions are not part of the document (since they are not in the speech data) and are not indexed or used in any other way except to create the queries. The binary relevance assessments were done manually. For each query, every document in the collection was examined to determine if it was relevant to the query or not. Some documents turned out to be relevant to more than one query. Each query has on average 6.2 relevant documents in the collection. The complete set of queries and their relevance judgments are listed in Appendix A.

We note that there is a mismatch between the queries and the documents since the former is text and the latter speech. However, this is a realistic scenario for applications where the user is entering text queries on a computer to search for relevant speech messages from an archived collection. A natural extension, which is beyond the scope of this thesis, is to have spoken queries so both the documents and queries are speech. One issue with text queries is that they need to be translated into a representation that is consistent with that used for the speech documents. With subword unit representations, we need to map the words in the query to their corresponding subword units. Since our subword units are derived from phonetic transcriptions (Section 3.2), the primary task in translating the text queries is the mapping of the words to their corresponding phone sequences. To do this, we make use of standard pronunciation dictionaries developed for speech recognition (McLemore 1997) and text-to-phone mapping algorithms developed for speech synthesis (Sproat 1998).

Although 12 hours is a reasonable amount of speech data, the amount of corresponding transcribed text is relatively small in comparison to the size of experimental *text* retrieval collections (Harman 1997) such as the ones described in Section 2.2. One needs to keep this in mind when generalizing the performance and results from retrieval experiments using this data set.

| Data Set | Size (MB) | # Docs | Avg. # Words/Doc |
|---|---|---|---|
| *Financial Times* (FT) 1991-1994 | 564 | 210,158 | 412.7 |
| *Federal Register* (FR) 1994 | 395 | 55,630 | 644.7 |
| *Congressional Record* (CR) 1993 | 235 | 27,922 | 1373.5 |
| *Foreign Broadcast Information Service* (FBIS) | 470 | 130,471 | 543.6 |
| *L.A. Times* (LA) | 475 | 131,896 | 526.5 |
| **TREC-6** (all 5 sources) | 2139 | 556,077 | 541.9 |
| **TREC-7** (4 sources excluding CR) | 1904 | 528,155 | 497.9 |
| **TREC-8** (same as TREC-7) | 1904 | 528,155 | 497.9 |

Table 2-2: Statistics for the document collections used in the TREC-6, TREC-7, and TREC-8 ad hoc text retrieval tasks.

## 2.2 TREC Ad Hoc Retrieval Text Corpora

To evaluate the new probabilistic retrieval model that we develop in Chapter 6, we use the standardized document collections from the ad hoc retrieval tasks in the 1997, 1998, and 1999 Text REtreival Conferences (TREC-6, TREC-7, and TREC-8) sponsored by the National Institute of Standards and Technology (NIST) (Harman 1997; Harman 1998; Harman 1999). The ad hoc task involves searching a static set of documents using new queries and returning an ordered list of documents ranked according to their relevance to the query. The retrieved documents are then evaluated against relevance assessments created for each query. These document collections consists of text stories from various news and information sources. Details of the composition and size of the collections are given in Table 2-2. The documents in the TREC-7 task are a subset of those in the TREC-6 task (documents from the *Congressional Record* are excluded from the TREC-7 collection). The document collection used in the TREC-8 task is identical to that used in TREC-7. Each collection contains approximately 2 gigabytes of text from over half a million documents. A sample document from the *L.A. Times* is shown in Figure 2-1. The documents are SGML (Standard Generalized Markup Language) tagged to facilitate parsing.

There are 50 queries (also called "topics") for each of the TREC-6, TREC-7, and TREC-8 ad hoc retrieval tasks. Topic numbers 301-350 are used in the TREC-6 task, while 351-400 are used in the TREC-7 task, and 401-450 are used in the TREC-8 task. Each topic consists

```
<DOC>
<DOCNO> LA073090-0005 </DOCNO>
<DOCID> 254383 </DOCID>
<DATE>
<P>
July 30, 1990, Monday, Home Edition
</P>
</DATE>
<SECTION>
<P>
Metro; Part B; Page 6; Column 1; Letters Desk
</P>
</SECTION>
<LENGTH>
<P>
34 words
</P>
</LENGTH>
<HEADLINE>
<P>
LOYAL FOLLOWING
</P>
</HEADLINE>
<TEXT>
<P>
Supporters of the insurance initiative, Proposition 103, remind me of
George Bernard Shaw's comment: "Those who rob Peter to pay Paul will
always have the support of Paul."
</P>
<P>
GARY A. ROBB
</P>
<P>
Los Angeles
</P>
</TEXT>
<TYPE>
<P>
Letter to the Editor
</P>
</TYPE>
</DOC>
```

Figure 2-1: A sample document from the TREC-6, TREC-7, and TREC-8 ad hoc retrieval task document collection. This is document number `LA073090-0005` from the *L.A. Times*.

```
<num> Number: 332

<title> Income Tax Evasion

<desc> Description:
This query is looking for investigations that have targeted evaders of
U.S. income tax.

<narr> Narrative:
A relevant document would mention investigations either in the U.S. or
abroad of people suspected of evading U.S. income tax laws. Of particular
interest are investigations involving revenue from illegal activities,
as a strategy to bring known or suspected criminals to justice.
```

Figure 2-2: A sample topic (number 332) from the TREC-6 ad hoc task. Each topic consists of three sections: a title, a description, and a narrative.

of three sections: a title, a description, and a narrative. A sample topic, number 332 from the TREC-6 ad hoc task, is shown in Figure 2-2. Statistics regarding the size of the topics are shown in Table 2-3.

In order to evaluate the performance of a retrieval system, relevance assessments must be provided for each topic. In other words, for each topic in the test set, the set of the known relevant documents in the collection needs to be determined. Since there are too many documents for complete manual inspection, an approximate method, known as the "pooling method," is used to find the set of relevant documents (Harman 1998). For each topic, a pool of possible relevant documents is first created by taking the top 100 documents retrieved from the various participating systems. Next, each document in this pool is manually assessed to determine its relevance. Finally, those documents that are judged relevant become the "answers" for the topic and are used to conduct the performance evaluations. Summary statistics for the number of relevant documents for the topics in the TREC-6, TREC-7, and TREC-8 ad hoc tasks are shown in Table 2-4. We note that there is great variability. Some topics have many relevant documents while other topics have only a few relevant documents.

In our text retrieval experiments in Chapter 6, we use the TREC-6 task as the "development" data set for tuning and optimizing our retrieval model. Most of the contrasting

| **Data Set** (topic #'s) | **# of Words** | | |
|---|---|---|---|
| | **Min** | **Max** | **Avg.** |
| **TREC-6** (301-350) | 47 | 156 | 88.4 |
| title | 1 | 5 | 2.7 |
| description | 5 | 62 | 20.4 |
| narrative | 17 | 142 | 65.3 |
| **TREC-7** (351-400) | 31 | 114 | 57.6 |
| title | 1 | 3 | 2.5 |
| description | 5 | 34 | 14.3 |
| narrative | 14 | 92 | 40.8 |
| **TREC-8** (401-450) | 23 | 98 | 51.3 |
| title | 1 | 4 | 2.4 |
| description | 5 | 32 | 13.8 |
| narrative | 14 | 75 | 35.1 |

Table 2-3: Statistics for the test topics used in the TREC-6, TREC-7, and TREC-8 ad hoc text retrieval tasks. There are 50 topics in each retrieval task.

| **Data Set** (topic #'s) | **# of Relevant Docs** | | | |
|---|---|---|---|---|
| | **Min** | **Max** | **Avg.** | **Total** |
| **TREC-6** (301-350) | 3 | 474 | 92.2 | 4611 |
| **TREC-7** (351-400) | 7 | 361 | 93.5 | 4674 |
| **TREC-8** (401-450) | 6 | 347 | 94.6 | 4728 |

Table 2-4: Statistics for the number of relevant documents for the topics in the TREC-6, TREC-7, and TREC-8 ad hoc text retrieval tasks. There are 50 topics in each retrieval task.

experiments will be done on the TREC-6 task. We reserve the TREC-7 task for use as the "test" data to objectively test our final retrieval model. An official TREC "evaluation" run was done using the TREC-8 task. Following standard practices, we use the entire topic statement (consisting of the title, description, and narrative components) in our retrieval experiments, unless otherwise noted.

## 2.3 Speech Recognition System

The SUMMIT speech recognition system, developed by the MIT Laboratory for Computer Science's Spoken Language Systems Group (Glass et al. 1996), is used to perform recognition on the speech messages in the spoken document corpora. The system adopts a probabilistic segment-based approach that differs from conventional frame-based hidden Markov model (HMM) approaches (Rabiner 1989). In segment-based approaches, the basic speech units are variable in length and much longer in comparison to frame-based methods. Acoustic features extracted from these segmental units have the potential to capture more of the acoustic-phonetic information encoded in the speech signal, especially those that are correlated across time, than short duration frame units. To extract these acoustic measurements, explicit segmental start and end times are needed. The SUMMIT system uses an "acoustic segmentation" algorithm (Glass 1988) to produce the segmentation hypotheses. Segment boundaries are hypothesized at locations of large spectral change. The boundaries are then fully interconnected to form a network of possible segmentations on which the recognition search is performed. The size of this network is determined by thresholds on the acoustic distance metrics.

The recognizer uses context-independent segment and context-dependent boundary (segment transition) acoustic models. The feature vector used in the segment models has 40 measurements consisting of three sets of Mel-frequency cepstral coefficient (MFCC) averages computed over segment thirds, two sets of MFCC derivatives computed over a time window of 40 ms centered at the segment beginning and end, and log duration. The derivatives of the MFCCs are computed using linear least square error regression. The boundary model feature vector has 112 dimensions and is made up of 8 sets of MFCC averages computed

over time windows of 10, 20, and 40 ms at various offsets ($\pm 5$, $\pm 15$, and $\pm 35$ ms) around the segment boundary. Cepstral mean subtraction normalization (Acero and Stern 1990) and principal components analysis are performed on the acoustic feature vectors in order to "whiten" the space prior to modeling.

The distribution of the feature vectors is modeled using mixture distributions composed of multivariate Gaussian probability density functions (PDF). The covariance matrix in the Gaussian PDF is restricted to be diagonal. Compared with full covariance Gaussians, diagonal covariance Gaussians have many fewer parameters and allows the use of more mixture components given the same amount of training data. In addition, the computational requirements for training and testing are reduced with the simpler distributions. The number of mixture components varies for each model and is decided automatically based on the number of available training tokens. The model parameters are trained using a two step process. First, the $K$-means algorithm (Duda and Hart 1973) is used to produce an initial clustering of the data. Next, these clusters are used to initialize the Estimate-Maximize (EM) algorithm (Dempster et al. 1977; Duda and Hart 1973) which iteratively estimates the parameters of the mixture distribution to maximize the likelihood of the training data. Since the EM algorithm is only guaranteed to converge to a local maximum, the final model parameters are highly dependent on the initial conditions obtained from the $K$-means clustering. To improve the performance and robustness of the mixture models, we used a technique called aggregation (Hazen and Halberstadt 1998), which is described in Section 4.2.

A two pass search strategy is used during recognition. A forward Viterbi search (Viterbi 1967; Forney 1973) is first performed using a statistical bigram language model. This pass significantly prunes the possible search space and creates a segment graph. Next, a backwards $A^*$ search (Winston 1992) is performed on the resulting segment graph using higher order statistical $n$-gram language models. In addition to applying more complex models, the second pass search can also be used to generate the $N$-best recognition hypotheses. In Section 4.2, we describe the development and application of this speech recognizer for use in the spoken document retrieval task. In Section 7.2, we modify the recognizer to facilitate a tighter integration between the speech recognizer and the retrieval model.

Speech recognition performance is typically measured in terms of the *error rate* (in percent) resulting from the comparison of the recognition hypotheses with the reference transcriptions. The total error is the sum of three different types of errors: substitutions, insertions, and deletions. A substitution error occurs when one symbol is confused with another, an insertion error happens when the hypothesis contains an extra symbol that is not in the reference, and a deletion error occurs when the hypothesis is missing a symbol that is in the reference. In this thesis, all speech recognition performance is reported in terms of error rate.

## 2.4 Information Retrieval Model

For our initial retrieval experiments, we implemented an information retrieval engine based on the standard vector space information retrieval (IR) model (Salton and McGill 1983). We later develop (in Chapter 6) a probabilistic information retrieval model that outperforms this initial retrieval model, that achieves performance competitive with current state-of-the-art approaches on text retrieval tasks, and that can be used in the development of a new spoken document retrieval approach that more tightly integrates the speech recognition and information retrieval components (Chapter 7).

In the vector space model, the documents and queries are represented as vectors where each vector component is an indexing term. A term can be a word, word fragment, or, in our case, a subword unit. Each term has an associated weight based on the term's occurrence statistics both within and across documents; the weight reflects the relative discrimination capability of that term. The weight of term $i$ in the vector for document $d$ is:

$$d[i] = 1 + \log(f_d[i]) \tag{2.1}$$

and the weight of term $i$ in the vector for query $q$ is:

$$q[i] = (1 + \log(f_q[i])) \cdot \log\left(\frac{N_D}{N_{D_i}}\right) \tag{2.2}$$

where $f_d[i]$ is the frequency of term $i$ in document $d$, $f_q[i]$ is the frequency of term $i$ in

query $q$, $N_{D_i}$ is the number of documents containing term $i$, and $N_D$ is the total number of documents in the collection. The weight in (2.1) is typically called the term frequency (TF). The second term in (2.2) is the inverse document frequency (IDF) for term $i$. Terms that occur in a small number of documents have a higher IDF weight than terms that occur in many documents. For computational efficiency, the IDF factor is included in the query terms but not the document terms. This allows the documents to be indexed in a single pass. Otherwise, a two pass indexing strategy is needed: the first pass to index the documents and compute the collection statistics (which is needed to compute the IDF factor) and the second pass to adjust the document term weights to include the IDF factor. The IDF values are still computed for use in the query term weights. In both document and query vectors, the weights are computed only for terms that occur one or more times; terms that do not occur are actually not represented in the vector.

A similarity measure between document and query vectors is computed and used to score and rank the documents in order to perform retrieval. A simple but effective retrieval measure is the normalized inner dot product between the document and query vectors (cosine similarity function) (Salton and McGill 1983):

$$S_e(\mathbf{q}, \mathbf{d}) \;=\; \frac{\mathbf{q} \cdot \mathbf{d}}{||\mathbf{q}||\;||\mathbf{d}||} = \sum_{i \in \mathbf{q}} \frac{q[i]}{||\mathbf{q}||}\;\frac{d[i]}{||\mathbf{d}||} \tag{2.3}$$

Because the vector space IR model was originally developed for use on text document collections, there are some limitations of this model when applied to spoken document retrieval. For example, there is no explicit mechanism for the approximate matching of indexing terms. With text this has generally not been an issue because the words in the documents are assumed to be known with certainty. However, with speech there will likely be errors generated by the recognizer and the need for approximate matching will be more important. Some amount of approximate matching can be done within the existing framework. For example the set of indexing terms can be expanded to include close alternatives with appropriate term weights. Alternatively, a new retrieval scoring function may be derived to allow approximate matching of the terms. Both of these methods (and several others) are examined in Chapter 5 when we explore approaches to improve retrieval perfor-

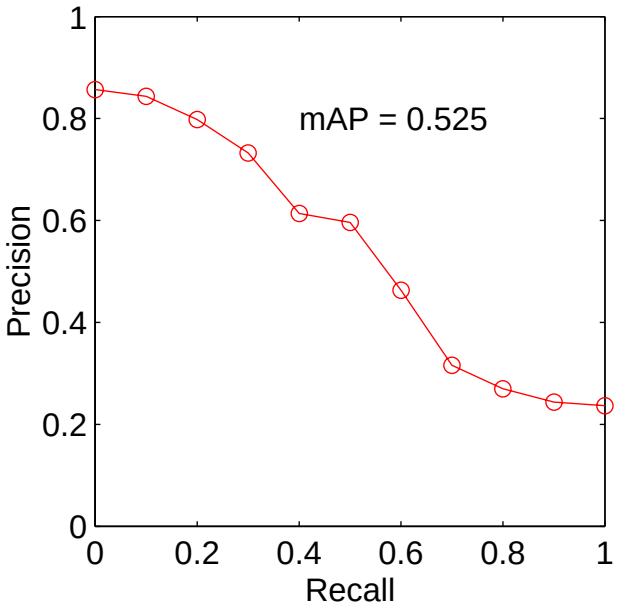


Figure 2-3: A precision-recall plot showing information retrieval performance. A single number performance measure, mean average precision (mAP), is computed by averaging the precision values at recall points of all relevant documents for each query and then averaging across all the queries.

mance in the presence of speech recognition errors. We note, however, that the development of more sophisticated robust indexing and retrieval methods that can make effective use of additional information available from the speech recognizer (e.g., multiple hypotheses, confidence scores, and recognition lattices) will require more significant changes to existing IR models. In our efforts to develop an approach to spoken document retrieval where the speech recognition and information retrieval components are more tightly integrated (Chapter 7), we find the need to develop a new probabilistic information retrieval model (Chapter 6).

Information retrieval performance is typically measured in terms of a tradeoff between *precision* and *recall* as illustrated in the graph in Figure 2-3. Precision is the number of relevant documents retrieved over the total number of documents retrieved. Recall is the number of relevant documents retrieved over the total number of relevant documents in the collection. If the retrieved documents are rank ordered according to a relevance score, as is typically the case, then the precision-recall curve can be generated by successively considering more of the retrieved documents by lowering the threshold on the score. If

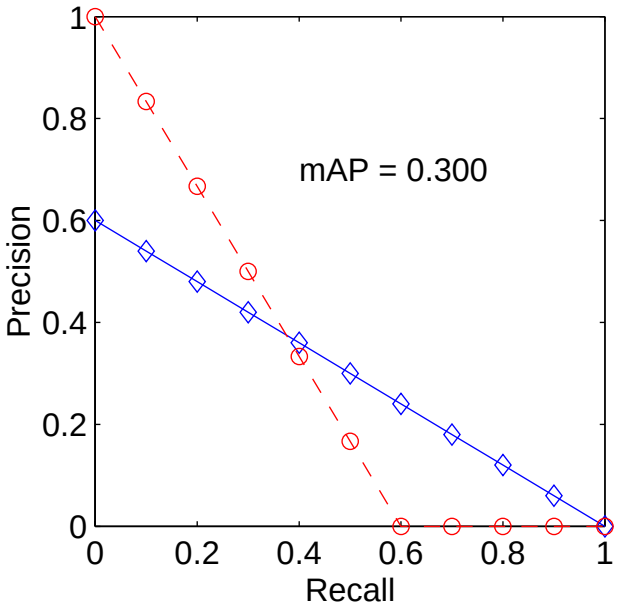


Figure 2-4: A precision-recall plot showing two different performance curves that have the same mean average precision (mAP) performance measure.

the retrieved documents are not scored, then a single operating point on the precision-recall graph is obtained. Because it is sometimes cumbersome to compare the performance of different retrieval systems using precision-recall curves, a single number performance measure called *mean average precision* (mAP) is commonly used (Harman 1997). It is computed by averaging the precision values at the recall points of all relevant documents for each query and then averaging those across all the queries in the test set. It can be interpreted as the area under the precision-recall curve. In this thesis, we report retrieval performance using this mean average precision metric. It is important to note that because the mAP measure is a single number, a lot of performance information is hidden. It does not allow analysis on different levels of precision at different levels of recall. For example, it is possible for two systems with very different precision-recall curves to have the same mAP measure as illustrated in Figure 2-4. The judgment of which is "better" depends on the particular application. If higher precision is more important, then the dashed curve (◯) is better. However, if higher recall is more important, then the solid curve (◇) is better.

# Chapter 3

# Feasibility of Subword Units for Information Retrieval

In this chapter, we address the issues of what subword units are suitable to use and how well they can perform in spoken document retrieval (Ng and Zue 1997a; Ng and Zue 1997b). First, we present some related work on subword representations. Next, we describe the set of subword unit indexing terms that we explored. Then, we establish a reference retrieval performance by using word units derived from error-free text transcriptions of the speech messages. Finally, using error-free phonetic transcriptions of the speech messages, we examine whether the various subword units have sufficient representational power to be used for indexing and retrieval. We find that many different subword units are able to capture enough information to perform effective retrieval and that it is possible, with the appropriate choice of subword units, to achieve retrieval performance approaching that of text-based word units if the underlying phonetic units are recognized correctly.

## 3.1 Related Work

Several subword based approaches to information retrieval have been proposed in the literature. For text retrieval, a method known as "character $n$-grams" has been developed which uses strings of consecutive text characters as the indexing terms instead of words (Cavnar 1994; Damashek 1995; Huffman 1995). For spoken documents, one approach makes

use of special syllable-like indexing units derived from text (Glavitsch and Schauble 1992; Schäuble and Glavitsch 1994) while others use phone sequences (phone $n$-grams) generated by post-processing the output of a phonetic speech recognizer (Ng and Zobel 1998; Wechsler and Schauble 1995). There are also methods that search for the query terms on phonetic transcriptions or phone lattice representations of the speech messages instead of creating subword indexing terms (Dharanipragada et al. 1998; James 1995; Jones et al. 1996; Wechsler et al. 1998).

In (Cavnar 1994; Huffman 1995), strings of $n$ consecutive characters generated from the text documents are used as the indexing terms. For example, the word `WEATHER` has the following character trigrams ($n$=3): `WEA`, `EAT`, `ATH`, `THE`, and `HER`. These character $n$-gram indexing terms are then used in a standard vector space IR model (similar to the one described in Section 2.4) to perform retrieval. Experiments on the TREC-3 (Cavnar 1994) and TREC-4 (Huffman 1995) ad hoc retrieval tasks show that performance using character $n$-grams ($n$=4) is reasonable but not as good as the best word-based retrieval systems (Harman 1994; Harman 1995). The method of character $n$-grams can be viewed as an alternate technique for word stemming that doesn't require linguistic knowledge. It maps words into shorter word fragments instead of semantically meaningful word stems. Since this method doesn't require prior knowledge about the contents of the documents or even its language, porting to a new domain or language is straightforward. However, the $n$-gram units have no semantic meaning and are therefore poor representations for concepts and their relationships. In addition to retrieval, character $n$-grams have also been successfully used to cluster text documents according to language and topic (Damashek 1995).

Syllable-like units derived from analyzing text documents are proposed in (Glavitsch and Schauble 1992; Schäuble and Glavitsch 1994) for use in spoken document retrieval. These subword units consist of a maximum sequence of consonants enclosed between two maximum sequences of vowels and are called "VCV-features." For example, the word `INFORMATION` has as its VCV-features: `INFO`, `ORMA`, and `ATIO`. These features are estimated from text transcriptions of the acoustic training data and then a subset is selected for use as the indexing terms. There are two criteria for selection: it must occur enough times to allow robust acoustic model training but not so often that its ability to discriminate between

different messages is poor. Word spotting is then performed on the speech messages to detect occurrences of the indexing terms. A standard vector space IR model is then used to perform retrieval. The indexing terms are weighted using TF×IDF weights (similar to (2.2)) and scoring is done using the cosine similarity function (2.3). In (Schäuble and Glavitsch 1994), retrieval experiments were performed using standard text document retrieval collections with simulated word spotting of the subword indexing features at various performance (detection and false alarm) levels. The main conclusion was that retrieval using these subword features is feasible even when the spotting performance is poor.

One major concern with the VCV-feature approach is that since only text transcriptions are used, acoustic confusability is not taken into account during the term selection process. A feature that discriminates well based on text may not work well on speech messages; maybe it cannot be reliably detected or it may have a high false alarm rate. A better approach may be to perform the feature selection on the acoustic data or on the speech recognition output. In this way, the recognizer characteristics are taken into account.

In (Ng and Zobel 1998; Wechsler and Schauble 1995), overlapping sequences of $n$ phones (phone $n$-grams) generated by post-processing the output of a phonetic speech recognizer are used as the indexing terms. For example, the word `weather`, represented phonetically as `w eh dh er`, has the phone trigrams ($n$=3): `w_eh_dh eh_dh_er`. Table 4-1 lists the set of phone labels used. The phone $n$-grams are then used in a standard vector space IR model (with TF×IDF term weights and a cosine similarity function) to perform retrieval. In (Wechsler and Schauble 1995), phone $n$-grams of length $n$=2,3,4 were examined and experiments indicated that the phone trigrams were optimal. With clean phonetic transcriptions, performance approached that of text word units and even with a high phonetic recognition error rate, performance was found to be reasonable.

Another set of "subword" approaches also makes use of phonetic representations of the spoken documents but not for generating indexing terms. Instead, these methods try to detect occurrences of the query terms by using word spotting techniques to search the phonetic representations for the phone sequences corresponding to the query terms. In (Wechsler et al. 1998), the query term search is done on a single phone sequence while in (James 1995; Jones et al. 1996) the search is done on a phone lattice which contains multiple phone

hypotheses. The approaches in (James 1995; Jones et al. 1996) use this "phone lattice scanning" procedure to complement a word based approach in an attempt to deal with new (out-of-vocabulary) words in the queries. A different multi-staged search algorithm is described in (Dharanipragada et al. 1998). A preprocessing stage first creates a phone level representation of the speech that can be quickly searched. Next, a coarse search, consisting of phone trigram matching, identifies regions of speech as putative query word occurrences. Finally a detailed acoustic match is done at these hypothesized locations to make a more accurate decision. Experiments in (James 1995; Jones et al. 1996; Wechsler et al. 1998) show that searching for phonetic representations of the query terms can be effective and that combining words and subwords in a hybrid approach performs better than just using words alone.

In the subword based approaches discussed above, generally only one type of subword unit or, at most, a small number of very similar types of subword units are explored and compared. We were unable to find any studies that explore the space of possible subword unit representations to measure their behavior and to determine the complexity of the subword units needed to perform effective spoken document retrieval. We believe that the experiments in this chapter are a step toward addressing this issue.

## 3.2 Subword Unit Representations

To explore the space of possible subword unit representations in order to determine the complexity of the subword units needed to perform effective spoken document retrieval, we examine a range of subword units of varying complexity derived from phonetic transcriptions. The basic underlying unit of representation is the phone; more and less complex subword units are derived by varying the complexity of these phonetic units in terms of their level of detail and sequence length. For level of detail, we look at labels ranging from specific phone classes to broad phonetic classes. For sequence length, we look at automatically derived fixed- and variable-length sequences ranging from one to five units long. In addition, sequences with and without overlapping units are also examined. Since it is difficult to obtain word and sentence boundary information from phonetic transcriptions, all

| Subword Unit | Indexing Terms |
|---|---|
| word | weather forecast |
| phone ($n$=1) | w eh dh er f ow r k ae s t |
| phone ($n$=2) | w_eh eh_dh dh_er er_f f_ow ow_r r_k k_ae ae_s s_t |
| phone ($n$=3) | w_eh_dh eh_dh_er dh_er_f er_f_ow f_ow_r ow_r_k r_k_ae k_ae_s ae_s_t |
| bclass ($c$=20, $n$=4) | liquid_frntvowel_voicefric_retroflex frntvowel_voicefric_retroflex_weakfric voicefric_retroflex_weakfric_$\cdots$ |
| mgram ($m$=4) | w_eh_dh_er f_ow_r k_ae_s_t |
| sylb | w_eh dh_er f_ow_r k_ae_s_t |

Table 3-1: Examples of indexing terms for different subword units. The reference word sequence, listed in the first row, is "weather forecast." The corresponding phonetic transcription is given in the second row labeled "phone ($n$=1)." Different subword units derived from the phonetic transcription are shown in the other rows: phone sequences of length $n$=2 and 3, broad class sequences of length $n$=4 with $c$=20 broad class categories, variable-length multigrams with a maximum length of $m$=4, and variable-length syllables.

subword units are generated by treating each message/query as a single long phone sequence with no word or sentence boundary information.

### 3.2.1 Phone Sequences

The most straightforward subword units that we examine are overlapping, fixed-length, phonetic sequences (phone) ranging from $n$=1 to $n$=5 phones long; a phone inventory of $c$=41 classes is used. These phonetic $n$-gram subword units are derived by successively concatenating together the appropriate number of phones from the phonetic transcriptions. Examples of $n$=1, 2, 3 phone sequence subword units for the phrase "weather forecast" are given in Table 3-1. Tables 4-1 and 4-4 lists the set of phone labels used. For large enough $n$, we see that cross-word constraints can be captured by these units (e.g., dh_er_f, er_f_ow).

For a given length $n$ and number of phonetic classes $c$, there is a fixed number of possible unique subword units: $c^n$. The number of units that are actually observed in real speech data is much less than the maximum number because many phone sequences are not possible in the language. For example, in the NPR spoken document set, the number

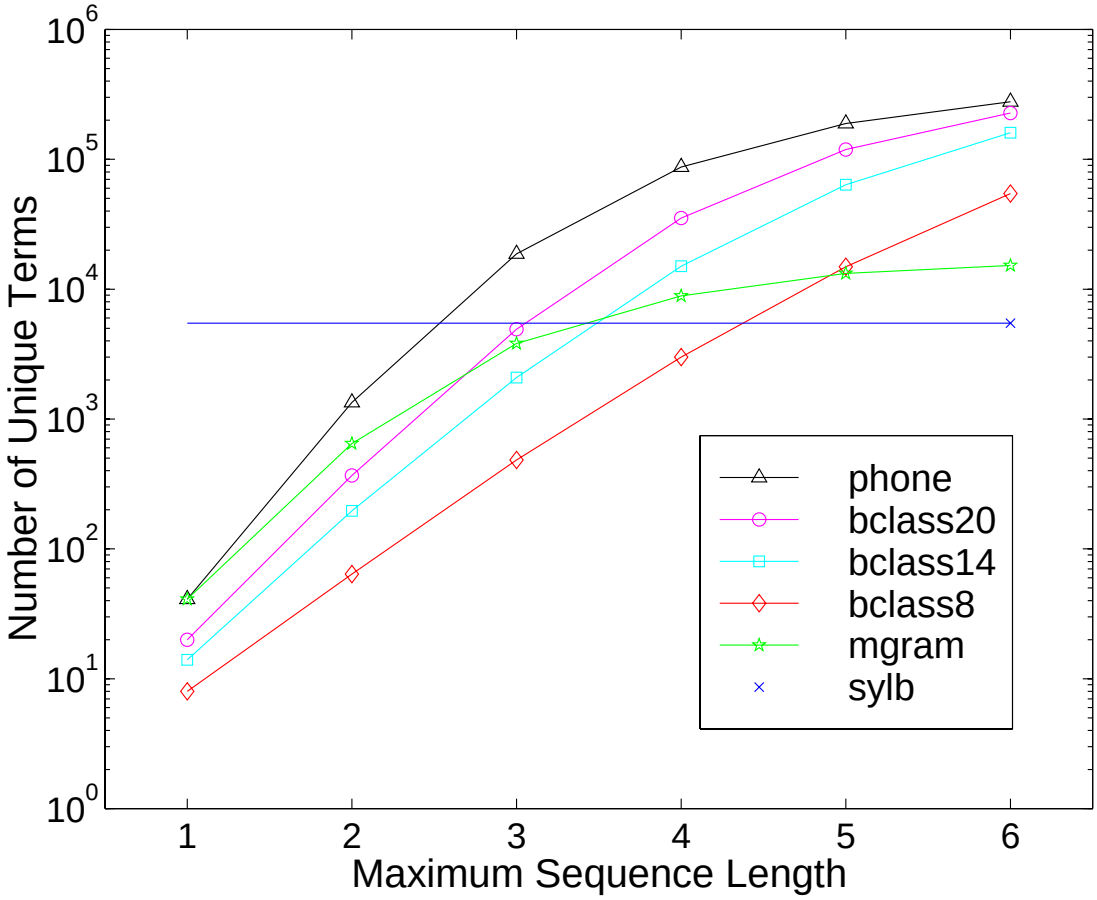


Figure 3-1: Number of unique terms for each type of subword unit. For each subword unit type (phone sequences, broad class sequences, multigrams, and syllables), the number of unique terms for varying sequence lengths ($n = 1, \ldots, 6$) is shown. The subwords are derived from clean phonetic transcriptions of the spoken documents from the NPR corpus.

of unique phonetic $n$-grams of length $n$=3 (derived from clean phonetic transcriptions of the speech) is 18705 out of a total of 68921 possibilities. The top 100 phonetic trigrams along with their frequency of occurrence on the NPR collection are tabulated in Table A-2 in Appendix A. Figure 3-1 plots the number of unique terms for phonetic sequences of varying lengths $n$=1, ..., 6. We see that the number of terms grows approximately exponentially with the length of the sequence.

### 3.2.2 Broad Phonetic Class Sequences

In addition to the original phone classes, we also explore more general groupings of the phones into broad phonetic classes (bclass) to investigate how the specificity of the phone labels (level of detail) impacts performance. Broad phonetic classes are interesting for a number of reasons. First, a lot of phonological constraints are captured with broad classes. For example, the permissible phone sequences in a language depend on the phonological characteristics of the sounds involved. The distributions of the phones (i.e., the phonotactics) are governed by constraints which refer not to individual sounds but to classes of sounds that can be identified by the distinctive features shared by members of that class. Distinctive features, such as **high**, **back**, and **round**, are minimal units in the phonological dimension that can be used to characterize speech sounds (Chomsky and Halle 1968). For instance, when three consonants appear at the beginning of a word in English, the first consonant must be an `s`, the second a **voiceless stop** (`p`, `t`, or `k`), and the third is constrained to be a **liquid** or **glide**: `l`, `r`, `w`, or `y`. A second reason is that phonetic classification and recognition experiments have shown that many of the errors occur between phones that are within the same broad phonetic class (Halberstadt 1998). In Section 5.2, we see evidence of this in the confusion matrix used to characterize the errors produced by our phonetic recognizer.

The broad classes are derived via hierarchical clustering of the 41 original phones using acoustic measurements derived from the TIMIT corpus (Garofolo et al. 1993). The feature vector has 61 measurements and consists of three sets of Mel-frequency cepstral coefficient (MFCC) averages computed over segment thirds, two sets of MFCC derivatives computed over a time window of 40 ms centered at the segment beginning and end, and log duration; each MFCC vector has 12 components. The final feature vector for each phone is computed by averaging the feature vectors from all occurrences of the phone in the TIMIT training set. The goal of the clustering is to group acoustically similar phones into the same class. We use standard hierarchical clustering (Hartigan 1975) which is a bottom-up procedure that starts with the individual data points. At each stage of the process the two "nearest" clusters are combined to form one bigger cluster. The process continues to aggregate groups together until there is just one big group. Depending on the distance metric and the clustering

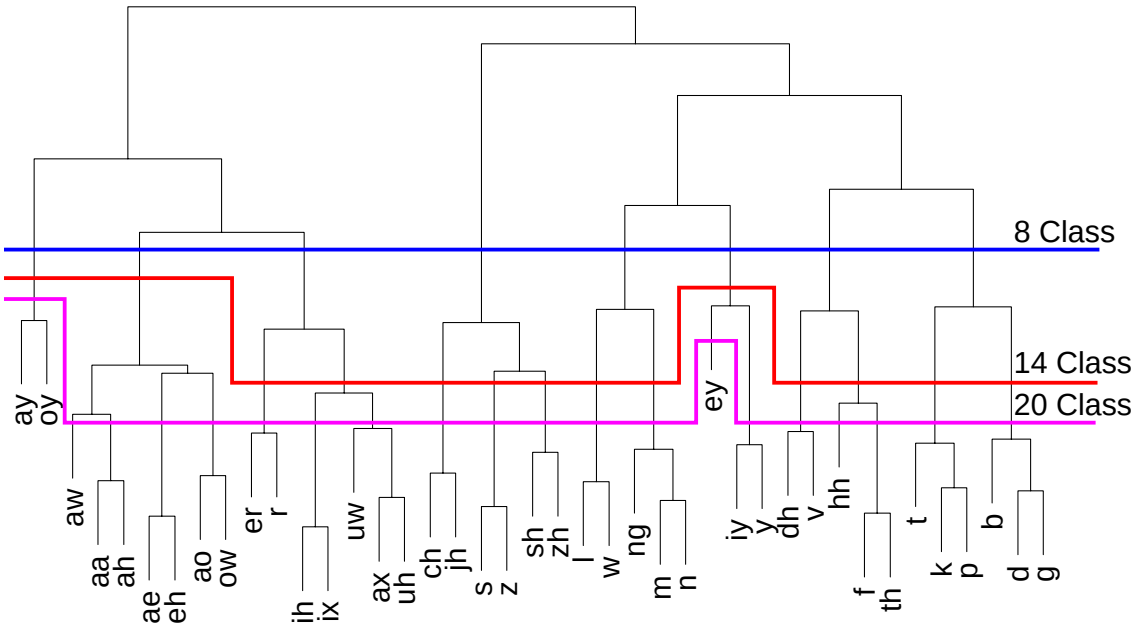


Figure 3-2: The hierarchical clustering tree used to generate phonetic broad classes. By cutting the tree at different heights, we obtain three different sets of broad classes with $c$=20, 14, and 8 distinct classes.

method used, several different cluster trees can be created from a single dataset. We use a simple Euclidean distance measure between the acoustic feature vectors of the phones and use the "complete linkage" clustering method in which the distance between two clusters is the largest distance between a point in one cluster and a point in the other cluster. This method gives rise to more "spherical" or compact clusters than alternative methods such as "average" (average distance) and "single linkage" (minimum distance).

Hierarchical clustering does not require the *a priori* specification of the number of desired clusters unlike other methods such as $K$-means clustering (Hartigan 1975). The decision regarding the number of clusters is made after the entire dendrogram tree is created. The number and membership of the clusters is determined by "cutting" the dendrogram at appropriate places to create subtrees. A known problem with hierarchical clustering is that deciding on the appropriate places to cut the dendrogram can be difficult. Figure 3-2 shows the resulting dendrogram and the "cuts" used to derive three different sets of broad classes with 20, 14, and 8 distinct classes. In deciding on the dendrogram cuts, we tried to maintain

one similarity threshold value, but had to change the value at some locations in order to obtain reasonable classes. For example, in creating the $c$=8 class set, it was possible to use a single threshold value (the cut is a straight line). However, with the $c$=14 and $c$=20 classes, several different threshold values had to be used. In particular, the threshold had to be increased in order to include the `ay` and `oy` phones in the $c$=20 class set. Examples of some broad class subword units (class $c$=20, length $n$=4) are given in Table 3-1. For the NPR spoken document set, the number of unique broad class subword units ($c$=20, $n$=4) derived from clean phonetic transcriptions of the speech is 35265 out of a total of $c^n = 20^4 = 160000$ possibilities. The top 100 units along with their frequency of occurrence on the NPR collection are tabulated in Table A-3 in Appendix A. Figure 3-1 plots the number of unique terms for broad class sequences with varying number of classes ($c$=20, 14, and 8) and varying sequence lengths ($n$=1, ..., 6).

Alternative methods can also be used to cluster the phones into broad classes. Instead of acoustic similarity, one can use a measure based on the number of distinctive features the phones have in common or use a set of rules to map the individual phones into established broad phonetic class categories such as back vowel, voiced fricative, nasal, etc. (Chomsky and Halle 1968). However, since we are interested in extracting the phonetic units from the speech signal, is seems reasonable to use a data driven approach with an acoustic similarity measure to group confusible phones together rather than using a measure of linguistic closeness. One difficulty with using a linguistic measure is that a decision needs to be made as to which linguistic feature is more discriminating when there is a conflict. With the data driven approach, this is automatically determined. For example, we see that the voiced/unvoiced distinction is more important than the place distinction when clustering the stops (`[p t k] [b d g]`) whereas place is more important for the strong fricatives (`[s z] [sh zh]`). It is reassuring that the resulting clusters are consistent with linguistically established broad phonetic classes.

### 3.2.3 Phone Multigrams

We also examine non-overlapping, variable-length, phonetic sequences (mgram) discovered automatically by applying an iterative unsupervised learning algorithm previously used in

developing "multigram" language models for speech recognition (Deligne and Bimbot 1995). The multigram model assumes that a phone sequence is composed of a concatenation of independent, non-overlapping, variable-length, phone subsequences (with some maximum length $m$). Given a segmentation of the sequence into subsequences, the likelihood of the sequence is the product of the likelihood of the individual variable-length subsequences. Without a segmentation, the likelihood of a sequence is the sum of the likelihoods over all possible segmentations.

In this model, the parameters are the set of subsequences and their associated probabilities. These parameters are trained with maximum likelihood (ML) estimation from incomplete data using the iterative expectation-maximization (EM) algorithm (Dempster et al. 1977): the observed data is the symbol sequence and the unknown data is the segmentation into the variable-length subsequences. At each training iteration, the likelihood of all possible segmentations is first computed using the current set of parameter values. Then the probability of each subsequence is re-estimated as the weighted average of the number of occurrences of that subsequence within each segmentation. Subsequence probabilities that fall below a certain threshold, $p_0 = 1 \times 10^{-6}$, are set to zero except those of length 1 which are assigned a minimum probability of $p_0$. After each iteration, the probabilities are renormalized so that they sum to 1. The model parameters are initialized with the relative frequencies of all phone sequences up to length $m$ that occur $c_0 = 2$ or more times in the training corpus. There is a dynamic programming (DP) algorithm similar to the hidden Markov model (HMM) forward-backward procedure that makes the EM algorithm efficient. Given a set of trained model parameters, a Viterbi-type search can then be used to generate a ML segmentation of an input phone sequence to give the most likely set of non-overlapping, variable-length, phone subsequences. The multigram model, with $m$=1,...,5, was trained on and then used to process the speech message collection. Examples of some multigram ($m$=4) subword units are given in Table 3-1.

For a given maximum length $m$ and number of phonetic classes $c$, there is a fixed number of possible unique multigram subword units: $\sum_{n=1}^{m} c^n$. Like the phonetic $n$-gram units, the number of multigram units that are actually observed in real speech data is much less because many phone sequences are not allowed. In addition, the multigram algorithm

selects only a subset of the possible units. For the NPR spoken document set, the number of unique multigrams ($m$=4) derived from clean phonetic transcriptions of the speech is 8875. The top 100 multigrams ($m$=4) along with their frequency of occurrence is shown in Table A-5 in Appendix A. Figure 3-1 plots the number of unique terms for multigrams with varying maximum lengths $m$=1, . . . , 6.

### 3.2.4 Syllable Units

We also consider linguistically motivated syllable units (sylb) composed of non-overlapping, variable-length, phone sequences generated automatically by rule. The rules take into account English syllable structure constraints (i.e., syllable-initial and syllable-final consonant clusters) and allow for ambisyllabicity (Fisher 1996; Kahn 1976). In Section 3.2.2, we mentioned that a lot of phonotactic constraint in the English language can be captured at the broad phonetic class level. In addition, the constraints on the combinations of phonemes within words can also be expressed by using structural units intermediate between phonemes and words, i.e., syllables (Chomsky and Halle 1968). Syllabic units were generated for the speech messages and queries using these rules, treating the message/query as one long phone sequence with no word boundary information. Examples of some syllabic subword units are given in Table 3-1. For the NPR spoken document set, the number of unique syllable units derived from clean phonetic transcriptions of the speech is 5475. This is plotted in Figure 3-1. The top 100 syllables along with their frequency of occurrence is shown in Table A-6 in Appendix A.

## 3.3 Text-Based Word Retrieval Reference

To provide a basis for comparison, we first establish a reference retrieval performance by using word units derived from error-free text transcriptions of the spoken documents and queries. This is equivalent to using a perfect word recognizer to transcribe the speech messages followed by a full-text retrieval system. Two standard IR text preprocessing techniques are applied (Salton and McGill 1983). The first is the removal of frequently occurring, non-content words using a list of 570 English "stop" words derived from the

stop-list used in the Cornell SMART system (Buckley 1985). The second is the collapsing of word variants using Porter's stemming algorithm (Porter 1980). Retrieval performance, measured in *mean average precision*, is mAP=0.87 (mAP=0.84 without stop word removal and word stemming). This number is high compared to text retrieval performance using very large document collections (Harman 1997) and indicates that this task is relatively straightforward. This is due, in part, to the relatively small number and concise nature of the speech messages. The text-based word performance (word) is plotted in the performance figures using a dotted horizontal line.

As an experimental control, we also evaluated the retrieval performance resulting from a random ordering of the documents in response to each query. Performance ranging from mAP=0.026 to mAP=0.031 from a number of different random trials are obtained. The small mAP numbers of the control experiments indicate that the retrieval algorithms are performing significantly better than chance.

## 3.4 Subwords From Error-Free Phonetic Transcriptions

Next, we study the feasibility of using subword units for indexing and retrieval. The goal here is to determine whether subword units have enough representational power to capture the information needed to perform effective information retrieval. For this experiment, the subword units are derived from error-free phonetic transcriptions of the speech messages and queries generated with the aid of a pronunciation dictionary. As a result, these experiments provide an *upper bound* on the performance of the different subword units since it assumes that the underlying phonetic recognition is perfect. It can also be used to eliminate poor subword units from further consideration.

Retrieval performance for the different subword units, measured in mean average precision, is shown in Figure 3-3. We can make several observations. First, as the length of the sequence is increased, performance improves, levels off, and then slowly declines. As the sequence becomes longer the units capture more lexical information and begin to approximate words and short phrases which are useful for discriminating between the different documents. After a certain length, however, the terms become too specific and can't match

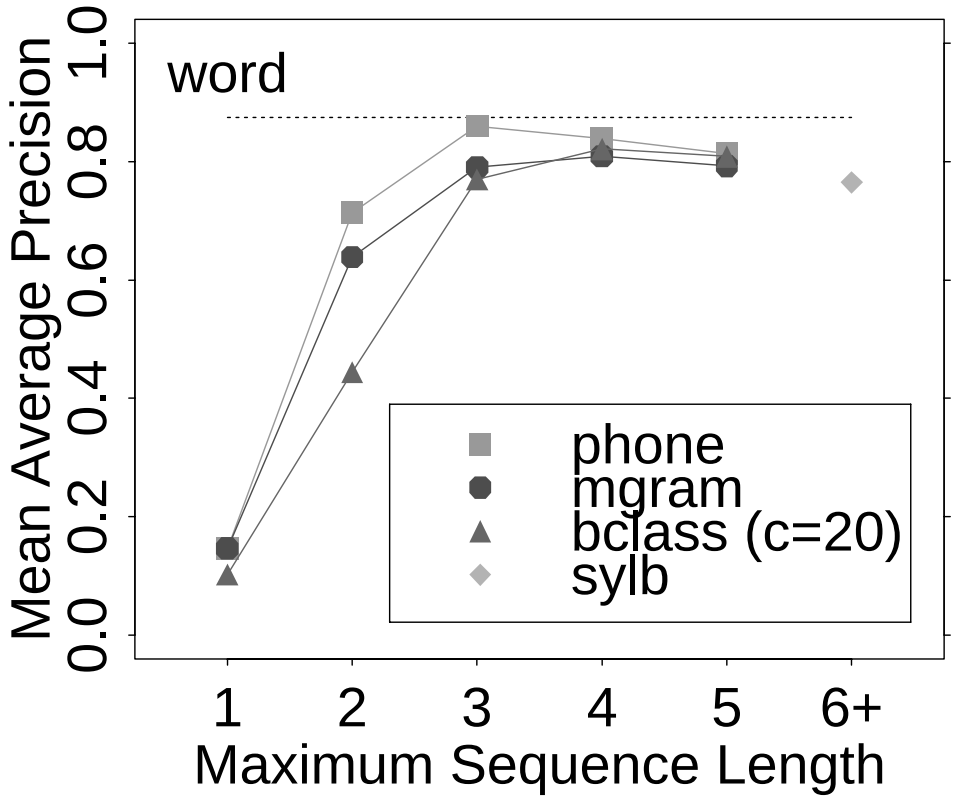


Figure 3-3: Retrieval performance of different subword units derived from error-free phonetic transcriptions. For each subword unit type (phone sequences, multigrams, broad class sequences ($c$=20), and syllables), performance for varying sequence lengths ($n = 1, \ldots, 6+$) is shown. Reference retrieval performance using word units derived from clean text is indicated by the dotted horizontal line.

other terms. Another way to interpret this is that as the subword unit gets longer, the inventory of possible subword units increases (as we saw in Section 3.2) thereby allowing a more expressive representation of the document content. Instead of being restricted to select repeated instances from a small inventory of subword units to represent the document, a smaller number of more unique units can be used to represent the same information. The longer units are more discriminating and can better differentiate between the different documents which leads to improved retrieval performance.

Second, overlapping subword units (phone, $n$=3, mAP=0.86) perform better than non-overlapping units (mgram, $m$=4, mAP=0.81). Units with overlap provide more chances for partial matches and, as a result, are more robust to variations in the phonetic realization of

the words. For example, the word "forecast" is represented by two multigram terms: `f_ow_r` and `k_ae_s_t` but five $n$=3 phonetic units: `f_ow_r, ow_r_k, r_k_ae, k_ae_s,` and `ae_s_t`. A pronunciation variation of the vowel `ow` would corrupt the first multigram term, leaving only the second term for matching purposes. With the $n$=3 phonetic unit representation, the first two terms would be corrupted, but the last three terms would still be available for matching. The increased number of subword units due to the overlapping nature of the generation process provide more chances for term matches. Thus, the impact of phonetic variations is reduced for overlapping subword units.

Third, between the two non-overlapping subword units (mgram and sylb), the automatically derived multigram units ($m$=4, mAP=0.81) perform better than the rule-based syllable units (mAP=0.76) when no word boundary information is used. As described in Section 3.2.3, the algorithm that generates the multigram units selects the most likely phone subsequences in the document collection and is designed to find consistent subword units. The rules that are used to generate the syllable units, on the other hand, only take into consideration phonotactic constraints to determine the units. There is no explicit guarantee of the consistency of the units in the generation process. Unit consistency comes about because of natural restrictions on the permissible phone sequences in the language. Since a syllable boundary always occurs at a word boundary, specification of the word boundaries simplifies the task of generating syllable units. The word boundaries break up the phone sequence into shorter subsequences in which to find the syllable units. More reliable syllables can be obtained this way. Without word boundaries, the task is more difficult because it is possible to hypothesize a syllable that spans across a word boundary. If the word boundaries are specified, then improved syllabic units are generated and retrieval performance improves from mAP=0.76 to 0.82 (not plotted). In this case, performance of the syllabic units reaches that of the multigram units.

Fourth, even after collapsing the number of phones down to 20 broad classes, enough information is preserved to perform effective retrieval (bclass, $c$=20, $n$=4, mAP=0.82). Figure 3-4 shows the retrieval performance of broad class subword units for a varying number of broad phonetic classes ($c$ = 41, 20, 14, 8). There is a clear tradeoff between the number of broad classes and the sequence length required to achieve good performance.

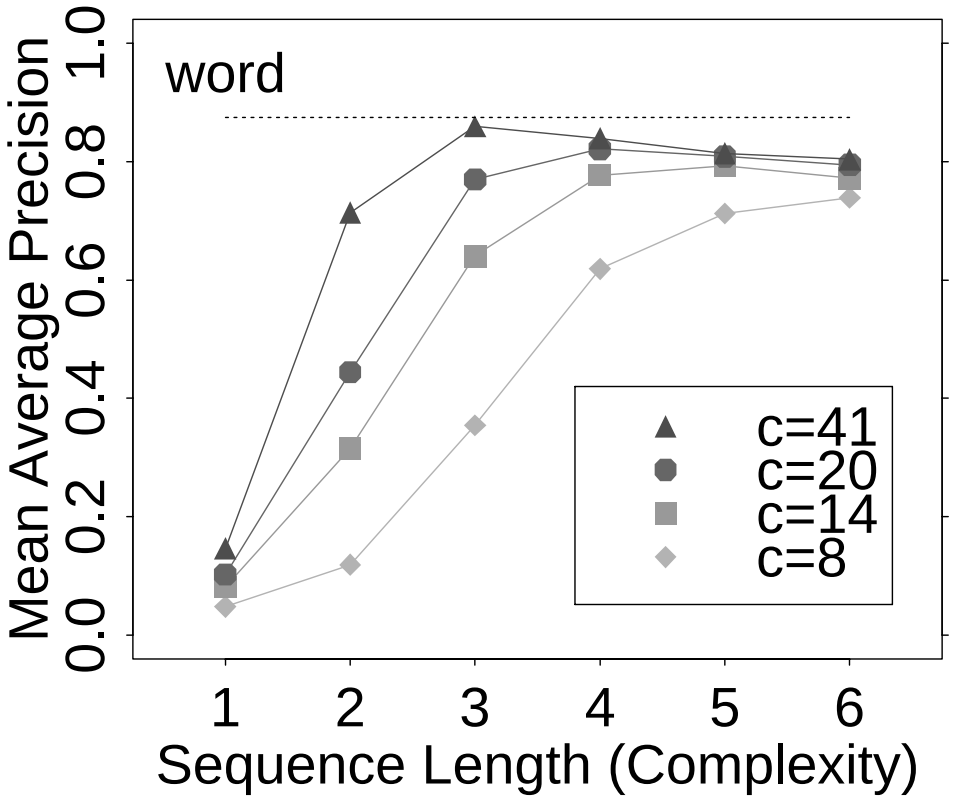


Figure 3-4: Retrieval performance of broad phonetic class subword units with varying number of broad phonetic classes ($c$=41,20,14,8) and sequence lengths ($n = 1, \ldots, 6$). Reference retrieval performance using word units derived from clean text is indicated by the dotted horizontal line.

As the number of classes is reduced, the length of the sequence needs to increase to retain performance. It is interesting to note that with 8 broad classes, only sequences of length 5 or 6 are needed to obtain reasonable retrieval performance. This indicates that there is a lot of phonotactic constraint in the English language that can be captured at the broad phonetic class level.

From these experiments, we find that many different subword units are able to capture enough information to perform effective retrieval. With the appropriate choice of subword units it is possible to achieve retrieval performance approaching that of text-based word units. For phone sequence subword units of length $n$=3, the retrieval performance is mAP=0.86 which is about the same as the performance using word units, mAP=0.87. Al-

| Subword Unit | Top N=5 Stop Terms |
|---|---|
| phone ($n$=3) | ae_n_d ax_n_t sh_ax_n dh_ae_t f_ao_r |
| mgram ($m$=4) | dh_ax ae_n_d ix_nx t_uw ax_n |
| sylb | dh_ax t_uw ax t_ax t_iy |

Table 3-2: Examples of automatically derived stop terms for different subword units: phone sequences of length $n$=3, multigrams with a maximum length of $m$=4, and syllables. The stop terms correspond to short function words and common prefixes and suffixes.

though beyond the scope of this thesis, it would be interesting to compare the performance of character $n$-grams with phone $n$-grams since both perform similar conflation of related words. In the next chapter, Chapter 4, we examine the performance of all of these subword units again, but this time the units are derived from errorful phonetic transcriptions generated by processing the speech messages with a phonetic speech recognizer.

## 3.5 Removal of Subword "Stop" Units

A standard IR technique that has been shown to improve performance is to remove frequently occurring, non-content words ("stop words") from the set of indexing terms (Salton and McGill 1983). We briefly explored several methods to automatically discover and remove "stop terms" from the different subword unit representations to try to improve performance. We used both term and inverse document frequencies to rank order and select the top $N = 5, \ldots, 200$ indexing terms as the "stop terms" to remove. A list of the top five terms for the phone ($n$=3), multigram ($m$=4), and syllable subword units is shown in Table 3-2. We see that they mainly consist of short function words and common prefixes and suffixes. We found that although removing these terms does improve retrieval performance, the gain was very small (less than 1% in mean average precision). It appears that the term weights are effective in suppressing the effects of these subword "stop" terms. However, we should note that the performance gain in using stop-lists can be collection dependent; the behavior we observe here may be different with another collection. In our subword unit experiments, "stop term" removal was not used.

## 3.6 Summary

In this chapter, we explored a range of subword units of varying complexity derived from error-free phonetic transcriptions and measured their ability to effectively index and retrieve speech messages. These experiments provide an *upper bound* on the performance of the different subword units since they assume that the underlying phonetic recognition is error-free. In particular, we examined overlapping, fixed-length phone sequences and broad phonetic class sequences, and non-overlapping, variable-length, phone sequences derived automatically (multigrams) and by rule (syllables). We found that many different subword units are able to capture enough information to perform effective retrieval. We saw that overlapping subword units perform better than non-overlapping units. There is also a trade-off between the number of phonetic class labels and the sequence length required to achieve good performance. With the appropriate choice of subword units it is possible to achieve retrieval performance approaching that of text-based word units if the underlying phonetic units are recognized correctly. Although we were able to automatically derive a meaningful set of subword "stop" terms, experiments using the stop-list did not result in significant improvements in retrieval performance.

# Chapter 4

# Extracting Subword Units from Spoken Documents

We now turn to the issues of extracting the subword units from the speech signal and evaluating the ability of the resulting subword units to perform spoken document retrieval. A two step procedure is used to generate the subword unit representations used for indexing and retrieval. First, a speech recognizer is used to create phonetic transcriptions of the speech messages. Then, the recognized phone units are processed to produce the subword unit indexing terms.

Aside from the top one hypothesis, additional recognizer outputs can also be used. This includes the top $N$ ($N$-best) hypotheses, outputs with associated confidence scores, and recognition lattices, which, like the $N$-best output, captures multiple recognition hypotheses. These outputs provide more recognition information which can be used during the indexing and retrieval process. As we will see in Chapter 5, some of this additional information can be useful in the development of approximate term matching and other robust indexing and retrieval methods. Also, as part of our effort to develop a more integrated SDR approach in Chapter 7, we modify the speech recognizer to output occurrence probabilities of the subword unit indexing terms instead of the most likely phone sequence.

In the following sections, we describe the development and application of a phonetic recognizer for use in the spoken document retrieval task. First, we look at some related work

on approaches for extracting subword units from the speech signal. Next, we investigate the use of different acoustic and language models in the speech recognizer in an effort to improve phonetic recognition performance. We then examine subword unit indexing terms derived from the errorful phonetic recognition output and measure their ability to perform effective spoken document retrieval. We also look at the relationship between phonetic recognition performance and spoken document retrieval performance. We find that in the presence of phonetic recognition errors, retrieval performance degrades compared to using error-free phonetic transcriptions or word-level text units. We also observe that phonetic recognition and retrieval performance are strongly correlated with better recognition leading to improved retrieval.

## 4.1 Related Work

In Section 3.1, we described several subword based approaches to information retrieval that have been proposed in the literature. In this section, we examine how these approaches generate their subword representations from the speech signal.

The phone $n$-gram units (overlapping sequences of $n$ phones) used in (Ng and Zobel 1998; Wechsler et al. 1998), are generated by post-processing the output of a phonetic recognizer. A speaker-independent HMM recognizer is built using the HTK Toolkit (Young et al. 1997). The acoustic models consist of a set of 40 context-independent phone models trained on the TIMIT corpus and a larger set of context-dependent biphone acoustic models trained on the TREC-6 SDR training set (50 hours of speech from television broadcast news) (Garofolo et al. 1997). A phonetic bigram statistical language model is used to constrain the search during recognition. The output consists of the single most likely phone sequence hypothesized by the recognizer. An additional post-processing step of clustering the 40 phones into a set of 30 broader classes is done to group acoustically similar phones together. Retrieval experiments on the TREC-6 SDR task (Garofolo et al. 1997) show that using phone $n$-grams generated from the errorful phone transcriptions is significantly worse than using units generated from error-free phonetic transcriptions. Performance is also worse than using word units generated by large vocabulary speech recognition. It is

important to note, however, that there is no out-of-vocabulary (OOV) problem in this task since the recognition vocabulary contains all the words in the test queries. The two stage approach used here to create the phone $n$-grams is similar to the method that we use to generate our subword unit indexing terms from the speech messages.

In (Dharanipragada et al. 1998; James 1995; Jones et al. 1996; Wechsler et al. 1998), a number of different "subword" representations are generated from the speech messages and then searched using word spotting techniques in an effort to detect occurrences of the query words. This is done by looking for the phone sequences corresponding to the query words in the subword representations. In (Wechsler et al. 1998), the query word search is done on a single (the most likely) phone sequence hypothesis generated by an HMM phone recognizer (the system described above). Retrieval experiments on the TREC-6 SDR task (Garofolo et al. 1997) show that the query word search technique can be effective although performance only reaches 57% of the reference (using error-free text word transcriptions). Experiments on a different document collection consisting of recorded Swiss radio news show that the query word search technique can perform better than using phonetic trigram indexing units (Wechsler and Schauble 1995). In (James 1995) and (Jones et al. 1996) the search is done on a "phone lattice" generated by a modified HMM phone recognizer. The lattice is a connected, directed, acyclical graph where each node represents a point in time and each edge is labelled with a phone hypothesis and a score representing its likelihood. The lattice structure allows more flexibility for matching query words than the single "best sequence" of phones normally generated by a standard recognizer. The lattice is a much more compact representation of the speech information. It is computationally much less expensive to perform searches on the lattice than directly on the original speech waveform. In (James 1995), experiments are done on a corpus of Swiss radio broadcast news and in (Jones et al. 1996), a corpus of video mail recordings is used. Both sets of experiments show that using just the phone lattice scanning technique can give reasonable retrieval performance. However, when it is used in combination with a word based approach to search for query words that are out of the recognition vocabulary, performance is much improved and is better than using either method individually. In the multi-staged search algorithm described in (Dharanipragada et al. 1998), a table of triphones (phone trigrams)

is used to rapidly search and identify regions of speech as potential query word occurrences. The triphone table is created by running an HMM phone recognizer on the speech message and outputting the identities of the top triphone models at regular time intervals during the recognition search along with their times of occurrence and acoustic scores. Like the phone lattice, the triphone table allows more flexibility for matching query words than the single most likely phone sequence hypothesis. This approach was proposed to complement a word based retrieval approach to deal with out of vocabulary words, but has not yet been evaluated in a retrieval task.

In recent spoken document retrieval experiments (TREC-7 SDR) (Garofolo et al. 1998) using large vocabulary speech recognition approaches, retrieval performance was found to be correlated with speech recognition performance. As expected, decreases in the recognition word error rate result in increases in the retrieval mean average precision. Similar behavior was observed in word spotting based approaches where improvements in spotting performance result in better retrieval performance (Jones et al. 1995b). Interestingly, the interaction between phonetic recognition error rate and retrieval performance using subword units derived from the errorful phonetic transcriptions have not been examined. We explore this relationship in Section 4.4.

## 4.2 Phonetic Recognition Experiments

In this section, we perform a series of experiments that explore the effects of using different acoustic and language models to try to improve phonetic recognition performance on the NPR speech data (Ng and Zue 1998).

### 4.2.1 Segment Acoustic Models

The baseline phonetic recognition system, which is based on the SUMMIT recognizer described in Section 2.3, uses 61 context-independent segment acoustic models corresponding to the TIMIT phone labels (Garofolo et al. 1993) and a phonetic bigram statistical language model. A list of the 61 phones in the TIMIT corpus with their IPA (international phonetic alphabet) symbols, TIMIT labels, and example occurrences is shown in Table 4-1.

The acoustic models are trained using 2.5 hours of "clean" speech from the speech recognition training set of the NPR corpus (Section 2.1). The language model is trained from the phonetic transcriptions of the entire training set (not just the "clean" utterances) which consists of approximately 230,000 phone occurrences. Performance, in terms of phonetic recognition error rate, is measured on a collapsed set of 39 classes typically used in reporting phonetic recognition results (Chang and Glass 1997; Halberstadt 1998; Lee 1989; Spina and Zue 1997). The mapping between the 61 phones and the 39 classes is shown in Table 4-2. Results on the development set for speech from all acoustic conditions (all) and from only the clean condition (clean) are shown in Table 4-3 (seg). We see that the acoustically diverse speech is considerably more difficult to recognize (43.5% error) than the clean speech (35.0% error).

### 4.2.2 Boundary Acoustic Models

Boundary models are context-dependent acoustic models that try to model the transitions between two adjacent segments. They are used in conjunction with the segment models and provide more information to the recognizer. Boundary models are trained for all segment transitions that occur more than once in the training data. For the 2.5 hours acoustic training set of the NPR corpus (Section 2.1), there is a total of 1900 such boundary models. As shown in Table 4-3 (+bnd), the addition of boundary acoustic models to the recognizer significantly improves phonetic recognition performance from 35.0% to 29.1% on clean speech and 43.5% to 37.7% on all conditions.

### 4.2.3 Aggregate Acoustic Models

Since the EM algorithm used to train the acoustic models makes use of random initializations of the parameter values and only guarantees convergence to a local optimum, different sets of models can result from different training runs using the same training data. An interesting question, then, is how to select the "best" set of models resulting from multiple training runs. It has been shown that aggregating or combining the different models into a single larger model results in better performance than either using just the set of models that yield the best performance on a development set or using methods such as cross validation (Hazen

| IPA | TIMIT | Example | IPA | TIMIT | Example |
|---|---|---|---|---|---|
| [ɑ] | aa | b*o*b | [ɨ] | ix | deb*i*t |
| [æ] | ae | b*a*t | [iʸ] | iy | b*ee*t |
| [ʌ] | ah | b*u*t | [ǰ] | jh | *j*oke |
| [ɔ] | ao | b*ough*t | [k] | k | *k*ey |
| [ɑʷ] | aw | b*ou*t | [k˺] | kcl | k closure |
| [ə] | ax | *a*bout | [l] | l | *l*ay |
| [əʰ] | ax-h | p*o*tato | [m] | m | *mom* |
| [ɚ] | axr | butt*er* | [n] | n | *noon* |
| [ɑʸ] | ay | b*i*te | [ŋ] | ng | si*ng* |
| [b] | b | *b*ee | [ɾ̃] | nx | wi*nn*er |
| [b˺] | bcl | b closure | [oʷ] | ow | b*oa*t |
| [č] | ch | *ch*oke | [oʸ] | oy | b*oy* |
| [d] | d | *d*ay | [p] | p | *p*ea |
| [d˺] | dcl | d closure | [□] | pau | pause |
| [ð] | dh | *th*en | [p˺] | pcl | p closure |
| [ɾ] | dx | mu*dd*y | [ʔ] | q | glottal stop |
| [ɛ] | eh | b*e*t | [r] | r | *r*ay |
| [l̩] | el | bott*le* | [s] | s | *s*ea |
| [m̩] | em | bott*om* | [š] | sh | *sh*e |
| [n̩] | en | butt*on* | [t] | t | *t*ea |
| [ŋ̩] | eng | Wash*ing*ton | [t˺] | tcl | t closure |
| [▯] | epi | epenthetic silence | [θ] | th | *th*in |
| [ɝ] | er | b*ir*d | [ʊ] | uh | b*oo*k |
| [eʸ] | ey | b*ai*t | [uʷ] | uw | b*oo*t |
| [f] | f | *f*in | [ü] | ux | t*oo*t |
| [g] | g | *g*ay | [v] | v | *v*an |
| [g˺] | gcl | g closure | [w] | w | *w*ay |
| [h] | hh | *h*ay | [y] | y | *y*acht |
| [ɦ] | hv | a*h*ead | [z] | z | *z*one |
| [ɪ] | ih | b*i*t | [ž] | zh | a*z*ure |
| - | h# | utterance initial and final silence | | | |

Table 4-1: List of the 61 phones in the TIMIT corpus. The IPA symbol, TIMIT label, and an example occurrence is shown for each phone.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **aa** | aa ao | **ae** | ae | **ah** | ah ax ax-h | **aw** | aw |
| **ay** | ay | **b** | b | **ch** | ch | **d** | d |
| **dx** | dx | **dh** | dh | **eh** | eh | **er** | er axr |
| **ey** | ey | **f** | f | **g** | g | **hh** | hh hv |
| **ih** | ih ix | **iy** | iy | **jh** | jh | **k** | k |
| **l** | l el | **m** | m em | **n** | n en nx | **ng** | ng eng |
| **ow** | ow | **oy** | oy | **p** | p | **r** | r |
| **s** | s | **sh** | sh zh | **t** | t | **th** | th |
| **uh** | uh | **uw** | uw ux | **v** | v | **w** | w |
| **y** | y | **z** | z | **CL** | bcl pcl dcl tcl gcl kcl epi pau h# | | |

Table 4-2: Mapping between the 61 TIMIT phones and the 39 phone classes typically used in measuring phonetic recognition performance. The glottal stop "q" is ignored.

and Halberstadt 1998). We adopt this aggregation approach and combine five separate acoustic models trained using different random initializations. The different models are combined using a simple linear combination with equal weights for each model. We observe a moderate performance improvement (from 29.1% to 27.9% on clean speech and 37.7% to 36.9% on all conditions) as shown in Table 4-3 (+agg).

### 4.2.4 Language Models

All of the above recognizers use a statistical bigram language model, trained on the phonetic transcriptions of the training data, to constrain the forward Viterbi search during decoding. More detailed knowledge sources, such as higher order $n$-gram language models, can be applied by running a second pass, backwards $A^*$, search. The higher order statistical language models provide more context and constraint for the recognition search. Using the same data used to training the bigram language model, we train and use $n$-grams of order $n$=3, 4, and 5 for the second pass and observe that recognition performance improves as $n$ increases. This can be seen in the last three columns ($n$=3, $n$=4, $n$=5) of Table 4-3.

The final phone error rate is 26.2% on the clean speech and 35.0% on the entire development set. State of the art phonetic recognition performance on the standardized TIMIT corpus, which is composed of clean speech, is around 25% (Chang and Glass 1997). We caution that these two sets of results cannot be compared directly. There are several com-

| | Acoustic Model | | | Language Model | | |
|---|---|---|---|---|---|---|
| **Model** | **seg** | **+bnd** | **+agg** | ***n*=3** | ***n*=4** | ***n*=5** |
| **Dev (clean)** | 35.0 | 29.1 | 27.9 | 27.3 | 26.7 | 26.2 |
| **Dev (all)** | 43.5 | 37.7 | 36.9 | 36.2 | 35.5 | 35.0 |

Table 4-3: Phonetic recognition error rate (%) on the entire development set (all) and on only the clean portions (clean) using various acoustic and language models. Segment (seg), boundary (+bnd), and aggregate (+agg) acoustic models and higher order statistical $n$-gram language models with $n$=3,4, and 5 are examined.

plicating factors. First, the characteristics of the speech data is different between the two corpora. The TIMIT corpus is composed of clean, studio-quality, phonetically balanced read speech. The NPR corpus, in contrast, contains read and spontaneous speech from a variety of acoustic and channel conditions. Second, the TIMIT task is designed to be speaker-independent whereas the NPR data is multi-speaker in nature. As a result, we can only take the performance numbers to be suggestive. To determine if the performance on the development set is indicative of the performance on the actual speech document collection, three hours of the speech messages (out of the 12+ hours total) are processed with the phonetic recognizer and evaluated against the corresponding manually obtained phonetic transcriptions. A phone error rate of 36.5% is obtained on this subset of the spoken document collection. This result confirms that the speech in the development set is consistent with the speech in the spoken document collection.

We note that additional work can be done in improving the performance of the phonetic recognizer, including the use of more training data to improve model robustness and the use of more complex models to capture more information from the speech signal.

## 4.3 Subwords From Errorful Phonetic Transcriptions

Next, we examine the retrieval performance of subword unit indexing terms derived from errorful phonetic transcriptions. These transcriptions are created by running the phonetic recognizer on the entire spoken document collection and taking the single best recognition output (later, in Chapter 5, we look at using additional recognizer outputs such as $N$-best

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **aa** | aa | **ae** | ae | **ah** | ah | **ao** | ao |
| **aw** | aw | **ax** | ax ax-h | **ay** | ay | **b** | b bcl+b |
| **ch** | ch | **d** | d dcl+d | **dh** | dh | **eh** | eh |
| **er** | er axr | **ey** | ey | **f** | f | **g** | g gcl+g |
| **hh** | hh hv | **ih** | ih | **ix** | ix | **iy** | iy |
| **jh** | jh | **k** | k kcl+k | **l** | l el | **m** | m em |
| **n** | n en nx | **ng** | ng eng | **ow** | ow | **oy** | oy |
| **p** | p pcl+p | **r** | r | **s** | s | **sh** | sh |
| **t** | t tcl+t dx | **th** | th | **uh** | uh | **uw** | uw ux |
| **v** | v | **w** | w | **y** | y | **z** | z |
| **zh** | zh | - | epi h# pau q | | | | |

Table 4-4: Mapping between the 41 phones used to create the subword unit indexing terms and the 61 TIMIT phones generated by the recognizer. Sequences of a stop closure followed by the stop are replaced by just the stop. In addition, the silence phones are ignored.

hypotheses). Errors are introduced only into the message collection; the text queries are not corrupted. The phonetic recognition error rate on the message collection, as mentioned in the previous section, is about 36.5%. The 61 phone labels are collapsed to a set of 41 labels using the mapping shown in Table 4-4. This mapping is slightly different from the one used to collapse the 61 phones to 39 classes for measuring phonetic recognition performance (Table 4-2), the primary differences being the treatment of stop closures/releases and in not collapsing a few phones together. Specifically, sequences of a stop closure followed by the stop, for example `kcl k`, are replaced by just the stop phone, `k`. The phone groups {`ah`, `ax`}, {`ih`, `ix`}, {`aa`, `ao`}, and {`sh`, `zh`} are not collapsed. In addition, the silence phones are ignored. After the phones are collapsed to this set of 41, the resulting phonetic transcriptions are then processed using the procedures described in Section 3.2 to generate the different subword unit indexing terms.

Figure 4-1 shows the retrieval performance, measured in mean average precision, of the (A) phone, (B) broad class, and (C) multigram subword units derived from error-free (text) and errorful (rec) phonetic transcriptions. We can make several observations. First, retrieval performance degrades significantly when errorful phonetic recognition output is used. The phonetic errors lead to corrupted subword units which then result in indexing term mismatches. Second, as the sequences get longer, performance falls off faster in the

errorful case than in the clean case. This is because more errors are being included in the longer terms which leads to more term mismatches. Finally, in the errorful case, broad class units are slightly better than phone units when the sequence is long ($n$=5). Because the broad class units collapse multiple similar phonetic units into the same class and treat them as equivalent, these units are more robust to phonetic recognition errors. In Section 5.2, we will see that most confusions occur between phones that are within the same broad phonetic class. The collapsed number of classes results in fewer recognition errors: the broad class ($c$=20) error rate is 29.0% versus 36.5% for the original set of classes. In addition to recognition error rate, sequential constraints also play a role. Because of the improved retrieval performance, it must be the case that enough phonotactic constraint is being preserved at the broad phonetic class level to maintain the necessary discrimination capability between indexing terms to differential the documents.

Retrieval performance for the best performing version of the different subword units (phone, $n$=3; bclass, $c$=20, $n$=4; mgram, $m$=4; and sylb) derived from error-free (text) and errorful (rec) phonetic transcriptions is shown in Figure 4-2. We note that the phonetic and broad class units perform better than the multigram and syllable units when there are speech recognition errors. There are several contributing factors. One is that the phonetic and broad class units are overlapping and fixed-length while the multigram and syllable units are non-overlapping and variable-length. Due to the two stage process we use to create the subword units from phonetic sequences, overlapping fixed-length units are more robust to variations and errors in the phone stream than variable-length non-overlapping units. The overlapping fixed-length units provide more opportunities for partial matching. More terms are created and the effect of a phonetic recognition error is more localized when using fixed-length overlapping units. We saw an example of this before in Section 3.4.

Another factor is that the multigram and syllable generation algorithms, which automatically discover their units from the phone stream, are able to find fewer consistent units when there are phonetic errors. This is evidenced by the larger number of multigram and syllable units generated when errorful phonetic transcriptions are used: 13708 multigram ($m$=4) units and 8335 syllable units. This is compared to 8875 multigrams ($m$=4) units and 5475 syllable units when clean phonetic transcriptions are used (see Sections 3.2.3

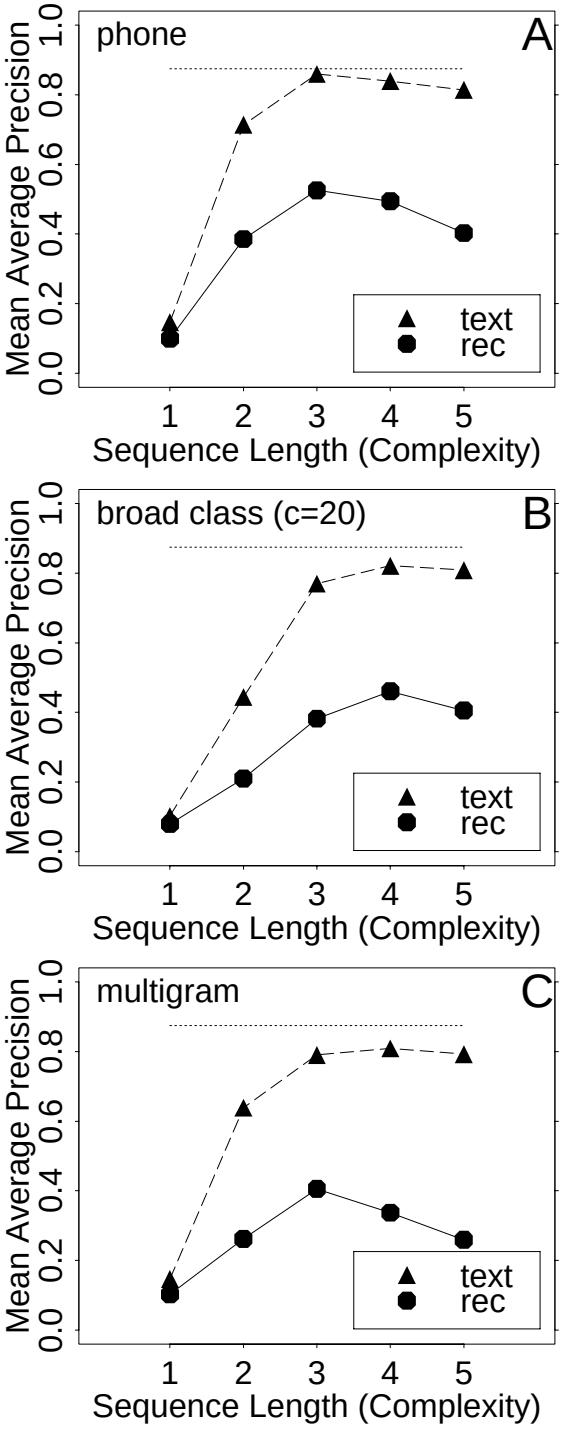


Figure 4-1: Performance of (A) phonetic, (B) broad class (with $c$=20 classes), and (C) variable-length multigram subword units of varying length ($n = 1, \ldots, 6$) derived from error-free (text) and errorful (rec) phonetic transcriptions. Reference retrieval performance using word units derived from clean text is indicated by the dotted horizontal line.

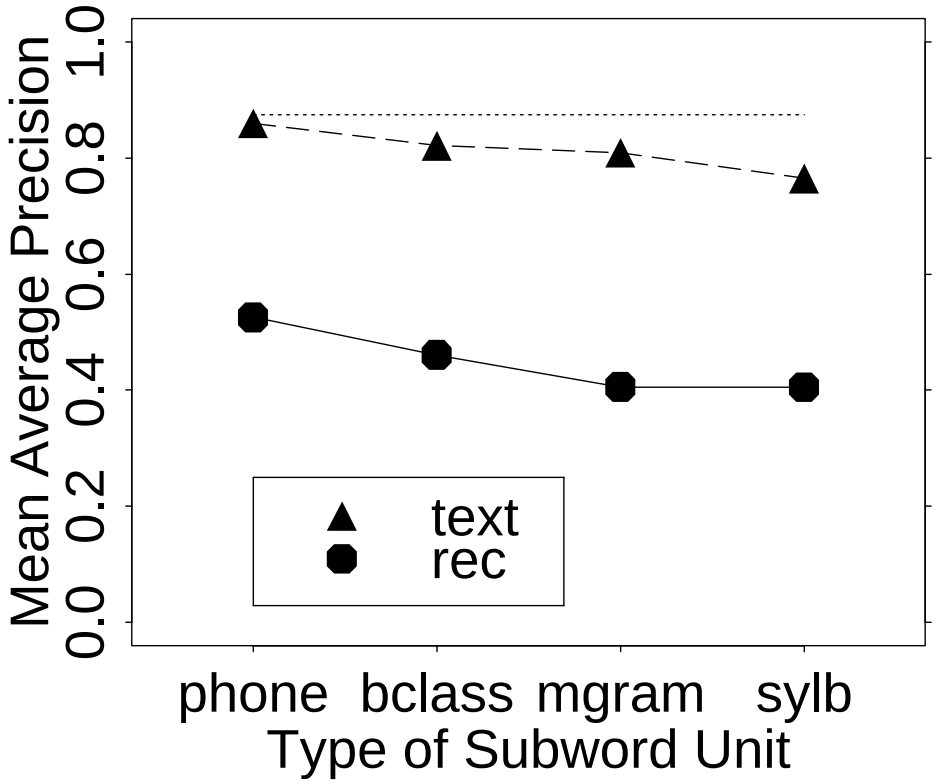


Figure 4-2: Retrieval performance for selected subword units (phone sequences, $n$=3; broad class sequences, $c$=20, $n$=4; multigrams, $m$=4; and syllables) derived from error-free (text) and errorful (rec) phonetic transcriptions. Reference retrieval performance using word units derived from clean text is indicated by the dotted horizontal line.

and 3.2.4). Fewer consistent units leads to poorer document representations and worse retrieval performance. One possible approach to try to make these units more robust to phonetic recognition errors is to build the multigrams and syllable units based on broad classes instead of detailed phonetic units. However, as we saw in Section 3.4, there exists a tradeoff between the number of broad classes and the sequence length required to achieve good retrieval performance. As the number of classes is reduced, the length of the sequence needs to increase in order to maintain performance. With multigram units, it is straightforward to increase the maximum length of the units generated. With syllable units, however, it is not clear how the length of the units can be increased in a meaningful way while preserving the notion and meaning of a syllable.

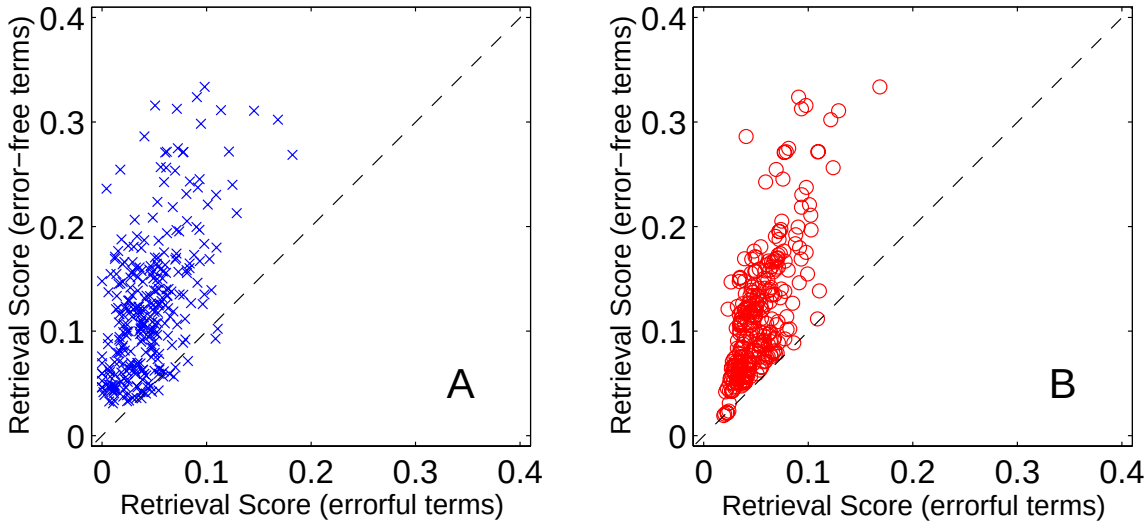


Figure 4-3: (A) Scatter plot of the retrieval scores of the relevant documents for all queries in the NPR data set using phonetic subword units of length $n$=3 derived from error-free and errorful phonetic transcriptions. (B) Scatter plot of the retrieval scores of the top 10 retrieved documents for all queries in the NPR data set using phonetic subword units of length $n$=3 derived from error-free and errorful phonetic transcriptions.

An analysis of the effects of the corrupted document terms caused by the speech recognition errors indicate that the decrease in retrieval performance is caused by false negative rather than false positive term matches. In other words, the reduced retrieval performance is the result of fewer query term matches in the relevant documents rather than more query term matches in the non-relevant documents. This behavior can be observed in Figures 4-3A and B. Figure 4-3A shows a scatter plot of the retrieval scores of the relevant documents for all queries using phonetic subword units of length $n$=3 derived from error-free (text) and errorful (rec) phonetic transcriptions. We see that the retrieval scores are lower when the subword units contain errors than when the subword units are error-free. The magnitude of the score is dependent on the number of query term matches. This means that the relevant documents have fewer query term matches when the document terms are corrupted than when they are error-free. Figure 4-3B shows a similar scatter plot, but this time the scores of the top 10 retrieved documents for each query are displayed. Both relevant and non-relevant documents are included in the top retrieved documents. We see that even the scores of the top documents are lower when the document terms are noisy than when

they are clean. This means that fewer query term matches are happening not only for the relevant documents but also for the non-relevant documents. Thus, the reduced retrieval performance with noisy document terms is due to relevant documents scoring lower rather than non-relevant documents scoring higher.

## 4.4 Recognition vs. Retrieval Performance

Because the subword unit indexing terms are derived from the phonetic recognizer output, speech recognition performance will have an impact on retrieval performance. To quantify this effect, we perform a series of retrieval experiments using one type of subword unit (phone sequences of length $n$=3) derived from phonetic transcriptions with different phonetic recognition error rates. The different outputs are generated by the different recognizers described in Section 4.2 when we explored the use of different acoustic and language models to improve phonetic recognition performance.

Figure 4-4 plots the relationship between spoken document retrieval performance, measured in mean average precision, and phonetic recognition performance, measured in error rate. As expected, we see that there exists a strong correlation: better phonetic recognition performance leads to better retrieval performance. It is interesting to note that relatively small changes in the phonetic recognition error rate result in much larger changes in the retrieval performance. For example, a decrease in the recognition error from 38% to 35% has a corresponding improvement in the retrieval performance from 0.435 to 0.525 in mean average precision. It will be interesting to see how much better retrieval performance can be with better phonetic recognizers. We should note, however, that accurate phonetic recognition is a very difficult task. The best reported speaker-independent phonetic recognition error rates (on clean speech) are only in the mid 20's (Chang and Glass 1997; Halberstadt 1998); with noisy speech data, the error rates will be much higher. The experiments in this section show that improving the performance of the speech recognizer remains an important intermediate goal in the development of better SDR systems.

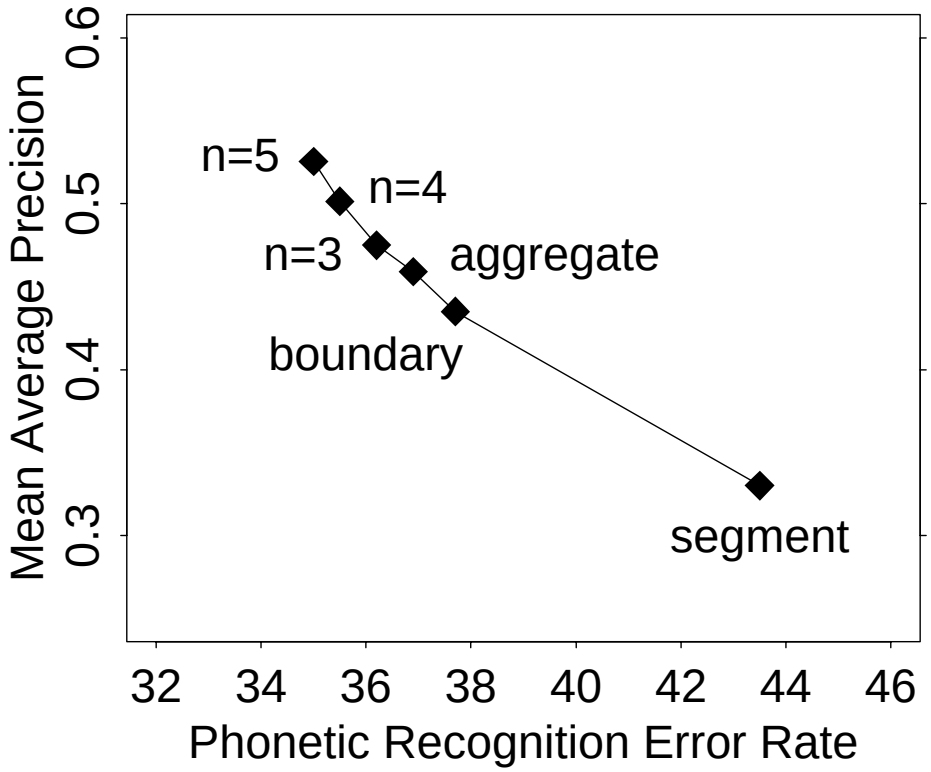


Figure 4-4: Relationship between spoken document retrieval performance (mean average precision) and phonetic recognition performance (error rate). The performance of the phonetic recognizer changes as different acoustic and language models are used.

## 4.5 Summary

In this chapter, we trained and tuned a phonetic recognizer to operate on the radio broadcast news domain and used it to process the entire spoken document collection to generate phonetic transcriptions. We then explored a range of subword unit indexing terms of varying complexity derived from these errorful phonetic transcriptions and measured their ability to perform spoken document retrieval. We found that in the presence of phonetic recognition errors, retrieval performance degrades, as expected, compared to using error-free phonetic transcriptions or word-level text units: performance falls to 60% of the clean reference performance. However, many subword unit indexing terms can still give reasonable performance even without the use of any error compensation techniques. We also observed that there is a strong correlation between recognition and retrieval performance: better phonetic

recognition performance leads to improved retrieval performance. The experiments in this chapter establish a *lower bound* on the retrieval performance of the different subword units since no error compensation techniques are used. We know that there are speech recognition errors, but we are not doing anything about them. Hopefully improving the performance of the recognizer and developing robust indexing and retrieval methods to deal with the recognition errors (which is the subject of the next chapter) will help improve retrieval performance.

# Chapter 5

# Robust Indexing and Retrieval Methods

In this chapter, we address the issue of modifying the indexing and retrieval methods to take into account the fact that the speech recognition outputs are errorful. We investigate robust indexing and retrieval methods in an effort to improve retrieval performance when there are speech recognition errors (Ng 1998). We examine a number of methods that take into account the characteristics of the recognition errors and try to compensate for them. In the first approach, the original query representation is modified to include similar or confusible terms that could match erroneously recognized speech; these terms are determined using information from the phonetic recognizer's error confusion matrix. The second approach is a generalization of the first method and involves developing a new document-query retrieval measure using approximate term matching designed to be less sensitive to speech recognition errors. In the third method, the document representation is expanded to include multiple recognition candidates (e.g., $N$-best) to increase the chance of capturing the correct hypothesis. The fourth method modifies the original query using automatic relevance feedback (Salton and McGill 1983) to include new terms as well as approximate match terms found in the top ranked documents from an initial retrieval pass. The last method involves the "fusion" or combination of information from multiple subword unit representations. We study the different methods individually and then explore the effects

of combining them. We find that each method is able to improve retrieval performance and that the gains are additive when the methods are combined. Overall, the robust methods improve retrieval performance using subword units generated from errorful phonetic recognition transcriptions by 23%.

In the experiments presented in this chapter, we use one of the better performing sets of subword units: overlapping, fixed-length, phone sequences ranging from $n$=2 to $n$=6 in length with a phone inventory of 41 classes. As described in Section 3.2.1, these phonetic $n$-gram subword units are derived by successively concatenating the appropriate number of phones from the phonetic transcriptions.

## 5.1 Related Work

In this section, we review some related work on approaches for dealing with errorful input in the context of information retrieval. For text documents, there has been work in trying to compensate for optical character recognition (OCR) errors introduced into automatically scanned text documents (Marukawa et al. 1997; Zhai et al. 1996). In (Marukawa et al. 1997), two methods are proposed to deal with character recognition errors for Japanese text documents. One method uses a character error confusion matrix to generate "equivalent" query strings to try to match erroneously recognized text. The other searches a "non-deterministic text" (represented as a finite state automaton) that contains multiple candidates for ambiguous recognition results. Experiments using simulated errors show that both methods can be effective. In (Zhai et al. 1996), two other OCR compensation methods are presented. One method is a query expansion technique that adds similar word variants as measured by an edit distance, i.e., the number of character insertions, deletions, and substitutions that are needed to transform one word to the other (Sankoff and Kruskal 1983). The second method is a word error correction technique that changes an unknown word (i.e., one not in a standard dictionary) to the most likely known word as predicted by a statistical word bigram model. Experiments on OCR corrupted English texts show that expanding the query to include similar words performs better than trying to correct word errors in the document.

The area of robust methods for dealing with speech recognition errors in the context of spoken document retrieval is still relatively new. There has been some recent work in this area performed independently and in parallel to the work presented in this chapter. One set of approaches expands the query to include terms that may have a better chance of matching the errorful terms in the document representation. Another expands the noisy document representation to include additional clean terms. A third method searches the errorful document representations for the query terms using approximate matching to allow for recognition errors. In (Ng and Zobel 1998), terms similar to the original query terms based on a standard edit distance or determined manually are added to the query. Retrieval experiments on the TREC-6 SDR task (Garofolo et al. 1997), using phonetic $n$-gram units of length $n$=3 and 4, did not show performance improvements using these methods. A possible explanation may be that the terms added to the queries were not weighted by their similarity to the original terms and no information about the characteristics of the speech recognizer errors was used.

In (Jourlin et al. 1999), a number of different query expansion techniques are used to try to compensate for word recognition errors. One method expands each geographic location word in the query with a partially ordered set of geographic location information. A second method adds hyponyms (members of a class) obtained from WordNet (Fellbaum 1998) for query words that are unambiguous nouns (i.e., nouns with only one possible word sense). Another method performs automatic (blind) relevance feedback (see Section 5.5) to expand the original query by including new terms from top ranked documents retrieved from a first pass. Two variants are explored; one performs the first pass retrieval on the spoken document collection generated by the speech recognizer and the other uses a parallel corpus comprised of clean text documents. Retrieval experiments on the TREC-7 SDR task (Garofolo et al. 1998) show that each method can improve retrieval performance and when used together results in a gain of over 14%.

In (Singhal et al. 1998), noisy document representations are expanded to include clean terms from similar documents obtained from a clean parallel text corpus. For each document in the spoken document collection, the top recognition word sequence is used to search a parallel clean text document collection for similar documents by treating the noisy

document as a query. A limited number of new terms from the top ranked retrieved clean documents are then added to the noisy speech document. Experiments on the TREC-7 SDR task (Garofolo et al. 1998) show that approach can improve retrieval performance.

In (Wechsler et al. 1998), a word spotting technique that allows for phone mismatches is used to detect query terms in the errorful phonetic transcriptions of the spoken documents. The top recognition hypothesis generated by an HMM phone recognizer is searched. Characteristics of the speech recognition errors are used in weighting phone sequences that approximately match the query term. As mentioned in Section 4.1, retrieval experiments on the TREC-6 SDR task (Garofolo et al. 1997) show that the query word search technique can be effective. Experiments on a spoken document collection consisting of recorded Swiss radio news show that the query word search approach can perform better than using phonetic trigram indexing units (Wechsler and Schauble 1995).

## 5.2 Expanding the Query Representation

Phonetic recognition errors in the spoken messages result in corrupted indexing terms in the document representation. One way to address this is to modify the query representation to include errorful variants of the original terms to improve the chance of matching the corrupted document terms. These "approximate match" terms are determined using information from the phonetic recognition error confusion matrix (Figure 5-1) obtained by running the recognizer on the development data set (See Section 2.1). Each confusion matrix entry, $C(r, h)$, corresponds to a recognition error confusing reference phone $r$ with hypothesis phone $h$. The bubble radius shown is linearly proportional to the error. The first row ($r = \emptyset$) and column ($h = \emptyset$) correspond to insertion and deletion errors, respectively. We note that many of the confusion errors occur between phones that are within the same broad phonetic class (Halberstadt 1998) and that many of the insertion and deletion errors happen with short phones.

By thresholding the error on the confusion matrix, we obtain a set of phone confusion pairs which can then be used to generate approximate match terms via substitution into each original query term. The threshold controls the number of near-miss terms added to

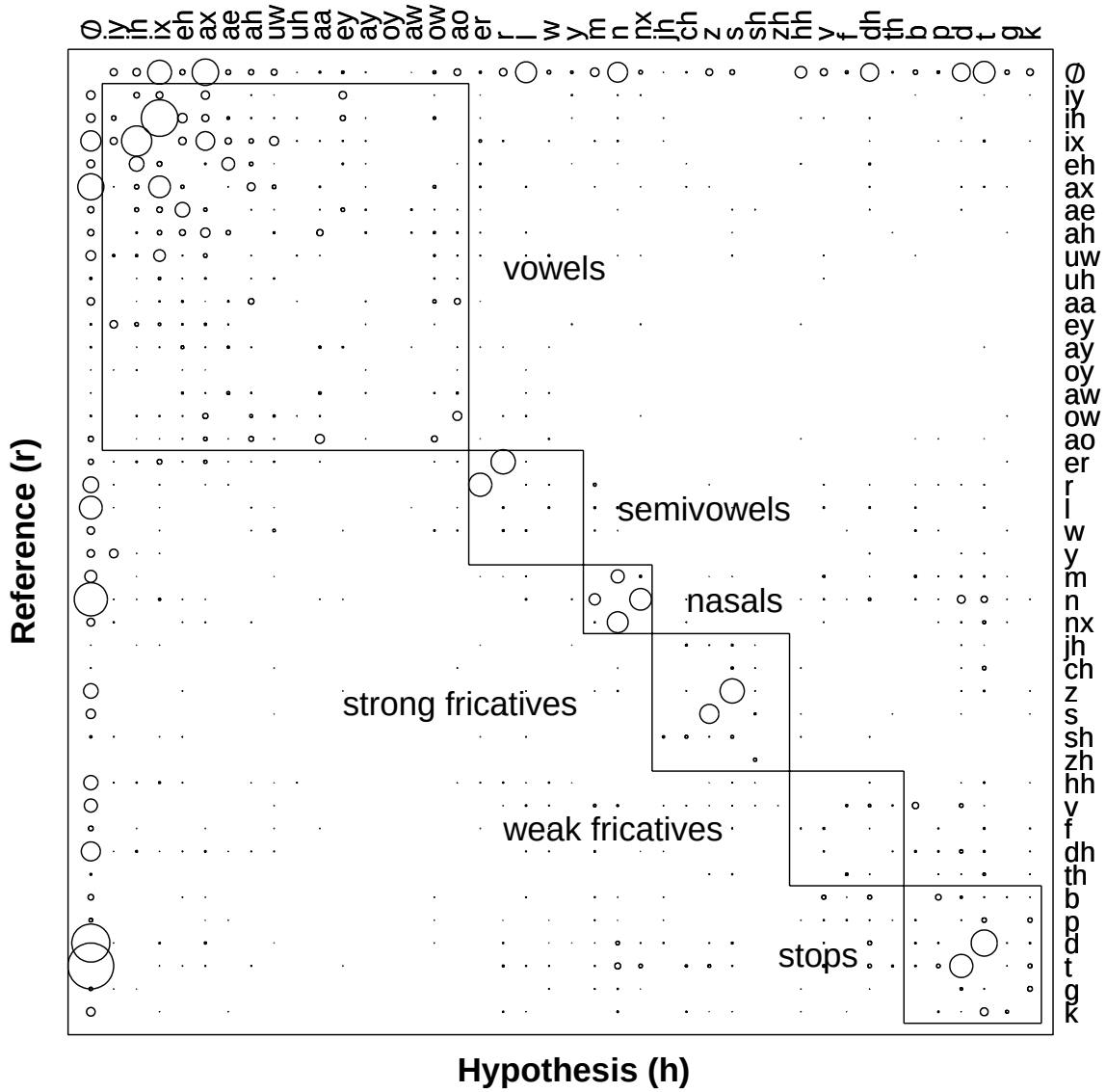


Figure 5-1: Phonetic recognition error confusion matrix $C$. The radius of the "bubbles" in each entry $C(r, h)$ are linearly proportional to the error so entries with large bubbles indicate more likely phone confusion pairs. The blocks along the diagonal group together phones that belong to the same broad phonetic category. Aside from insertion and deletion errors, which happen mainly with the short duration phones, most confusions occur between phones that are within the same broad phonetic class.

| **reference term** $i$ | **near-miss term** $j$ | **similarity** $s(i,j)$ |
|---|---|---|
| eh_dh_er | eh_dh_er | 1.0000 |
| eh_dh_er | ih_dh_er | 0.8892 |
| eh_dh_er | ae_dh_er | 0.8873 |
| eh_dh_er | eh_dh_r | 0.8733 |
| eh_dh_er | ih_dh_r | 0.7625 |
| eh_dh_er | ae_dh_r | 0.7605 |

Table 5-1: A list of some automatically generated near miss terms $j$ along with their similarity scores $s(i,j)$ for a reference term $i$=eh_dh_er.

the query. The frequency weight of a new term $j$ is a scaled version of the frequency weight of the original term $i$ where the scaling factor is a heuristically derived similarity measure between terms $i$ and $j$:

$$f_q[j] = s(i,j)\ f_q[i] = \frac{\sum_{m=1}^{n} C(i[m], j[m])}{\sum_{m=1}^{n} C(i[m], i[m])}\ f_q[i] \tag{5.1}$$

where $i[m]$ is the $m^{\text{th}}$ phone in subword unit term $i$ with length $n$. The measure is normalized so that exact term matches will have a weight of one. In this approach, we are using the confusion matrix $C$ as a *similarity* matrix with the error values as indicators of phone similarity. Since the confusion matrix entries, $C(r,h)$, are counts of the number of confusions between phones $r$ and $h$, the similarity measure is a normalized sum of the number of total confusions between the phones in terms $i$ and $j$. Table 5-1 shows an example of some near-miss terms that are generated along with their similarity scores for the reference term eh_dh_er. Terms acoustically close to the original term have a similarity score close to one.

We note that this method of creating near miss terms can only allow for substitution errors; insertion and deletion errors are not modeled directly. We need to rely on the partial matching of the terms and the overlapping nature of the subword units to allow for more matching opportunities to try compensate for this. A more direct approach would be to develop a more complex model that can explicitly allow for these other types of errors (Livescu 1999). In our approach to approximate matching (Section 5.3), we develop a more general (allowing for insertion and deletion errors) and better motivated probabilistic measure of the similarity between two terms $i$ and $j$.

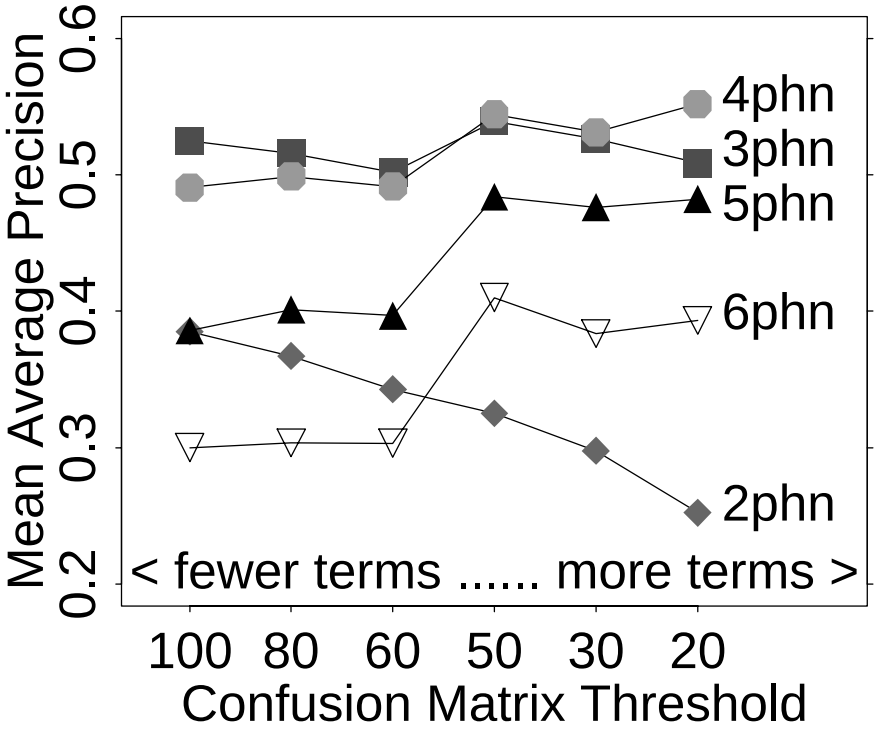


Figure 5-2: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units as the query expansion threshold is varied. More terms are added to the query as the threshold value is lowered.

Figure 5-2 shows retrieval performance, measured in mean average precision, for the different phonetic subword units ($n$=2,3,4,5,6) using this query expansion method as the threshold is lowered to include more approximate match terms. We first note that subword units of intermediate length ($n$=3,4) perform better than short ($n$=2) or long ($n$=5,6) units; this is due to a better tradeoff of being too short and matching too many terms versus being too long and not matching enough terms. As the threshold is lowered and more terms are added, performance of the short subword unit ($n$=2) becomes worse. This is due to an increase in the number of spurious matches caused by the additional terms. Because of the short length of these subword units and the small number of possible terms ($41^2 = 1681$), the added terms are likely to match terms that occur in many of the documents. The performance of the longer subword units ($n$=4,5,6), however, are much improved with expanded queries. In this case, the additional query terms are matching corrupted document terms

| | Threshold Value | | | | | | |
|---|---|---|---|---|---|---|---|
| **Subword Unit** | **100** | **80** | **60** | **50** | **30** | **20** | **10** |
| **2phn** | 22.0 | 24.1 | 34.6 | 50.4 | 62.2 | 100.2 | 213.9 |
| **3phn** | 21.0 | 24.0 | 41.2 | 72.7 | 99.7 | 202.4 | 625.6 |
| **4phn** | 20.0 | 23.9 | 48.9 | 104.1 | 159.8 | 410.5 | 1830.1 |
| **5phn** | 19.0 | 23.8 | 58.3 | 150.6 | 257.3 | 839.6 | 5428.5 |
| **6phn** | 18.0 | 23.6 | 69.0 | 217.0 | 411.5 | 1704.7 | 15768.3 |

Table 5-2: The growth in the average number of terms in the query as the query expansion threshold is varied. We start with the original query at a threshold value of 100. More terms are added to the query as the threshold value is lowered.

but the longer subword unit sequence length now makes it more difficult to get spurious matches. The net effect is positive resulting in performance improvements. Performance for the length $n$=3 units stays about the same. In this case, the combined effect of corrupted term matches and spurious matches result in no net gain.

Table 5-2 shows the growth in the size of the query (in number of terms averaged over the 50 queries) as the query expansion threshold is varied. At a threshold value of 100, the query is the original one with no added terms. As the threshold value is lowered, more phone confusions are considered and more near-miss terms are added to the query. We observe that the queries for the longer subword units grow faster as the threshold is lowered. At a threshold of 20, the query growth is almost exponential. In the last column of the table, we also include the average number of query terms for a threshold value of 10. It is clear that the number of possible near-miss terms resulting from considering more and more phone confusions can grow very large.

## 5.3 Approximate Match Retrieval Measure

Instead of explicitly adding query terms to improve the chance of matching corrupted document terms, we can implicitly consider *all* possible matches between the "clean" query terms and the "noisy" document terms by generalizing the document-query retrieval metric to make use of approximate term matching.

In the retrieval measure specified in (2.3), the document vector $\mathbf{d}$ contains a weight,

$d[i]$, for each term $i$ that occurs in that document. The original retrieval model assumes that these weights are derived from error-free text transcriptions of the documents. With speech messages, however, the document representation contains errors introduced by the speech recognizer. The retrieval metric should, therefore, be modified to take these errors into account. One approach is to estimate the weight of term $i$ in a noisy document as a weighted sum of the weights of all the observed terms in that document:

$$d[i] = \sum_{j \in \mathbf{d}} p(i \,|\, j)\; d[j] \tag{5.2}$$

where the mixing weight, $p(i \,|\, j)$, is the conditional probability that the term is really $i$ given that we observe term $j$ in the noisy document. This conditional probability models the error characteristics of the speech recognizer since it estimates the probability of reference term $i$ given that we observe hypothesis term $j$. Using this estimate of term weights from noisy documents, we can formulate a new retrieval measure that allows for approximate matches between a clean query term $i$ and a noisy document term $j$ as follows:

$$\begin{aligned} S_a(\mathbf{q}, \mathbf{d}) &= \sum_{i \in \mathbf{q}} \frac{q[i]}{||\mathbf{q}||} \left\{ \sum_{j \in \mathbf{d}} p(i \,|\, j)\; \frac{d[j]}{||\mathbf{d}^*||} \right\} \\ &= \sum_{i \in \mathbf{q}} \sum_{j \in \mathbf{d}} p(i \,|\, j)\; \frac{q[i]}{||\mathbf{q}||}\; \frac{d[j]}{||\mathbf{d}^*||} \end{aligned} \tag{5.3}$$

where $||\mathbf{d}^*||$ is the magnitude of the new document vector described in Equation 5.2. We observe that the new metric (5.3) reduces to the original metric (2.3) when only exact matches between terms $i$ and $j$ are allowed:

$$p(i \,|\, j) = \begin{cases} 1, & \text{if } \; j = i \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

If we make the simplifying assumption that the phones comprising each subword unit term are independent, then we can estimate this conditional probability using a dynamic programming (DP) procedure:

$$p(i \,|\, j) = A(l_i, l_j) \tag{5.5}$$

where $l_i$ and $l_j$ are the lengths of terms $i$ and $j$, respectively, and $A$ is the $l_i \times l_j$ DP matrix which can be computed recursively:

$$A(m,n) = \begin{cases} 1, & m{=}0,\ n{=}0 \\ A(0,n{-}1) \cdot \tilde{C}(\emptyset, j[n{-}1]), & m{=}0,\ n{>}0 \\ A(m{-}1,0) \cdot \tilde{C}(i[m{-}1], \emptyset), & m{>}0,\ n{=}0 \\ \max \begin{cases} A(m{-}1,n) \cdot \tilde{C}(i[m{-}1], \emptyset) \\ A(m{-}1,n{-}1) \cdot \tilde{C}(i[m{-}1], j[n{-}1]) \\ A(m,n{-}1) \cdot \tilde{C}(\emptyset, j[n{-}1]) \end{cases}, & m{>}0,\ n{>}0 \end{cases} \tag{5.6}$$

where $\tilde{C}(r,h)$ is the probability of reference phone $r$ given that we observe hypothesis phone $h$ and is obtained by normalizing the error confusion matrix:

$$\tilde{C}(r,h) = \frac{C(r,h)}{\sum_{k \in \{h\}} C(r,k)} \tag{5.7}$$

Thresholds can be placed on $p(i \mid j)$ to limit the number of approximate term matches that have to be considered when computing the retrieval score in (5.3). We note that other probabilistic models such as hidden Markov Models (HMMs) can also be used to estimate this conditional probability.

It is important to note that the use of this approximate match score can significantly increase the amount of computation needed to perform retrieval because we have to now consider all possible matches between the terms in the query and the terms in the documents. In the original model, only the query terms that occur in the documents needed to be examined. If there are $n_q$ terms in the query and $n_d$ terms in the document, we have to, in principle, consider $n_q \times n_d$ terms for approximate matching whereas only $n_q$ terms are needed for exact matching. Fortunately, the most computationally expensive part of the processing, the computation of the quantity $p(i \mid j)$ for all $i$ and $j$, can be done once (for a given speech recognizer) off-line and stored in a table for future use during retrieval.

Figure 5-3 shows retrieval performance for the different phonetic subword units using the new document-query retrieval metric in (5.3) as the threshold on $p(i \mid j)$ is lowered to consider more approximate matches. The performance behavior is very similar to that observed in Figure 5-2 with improvements for the longer subword units and losses for the

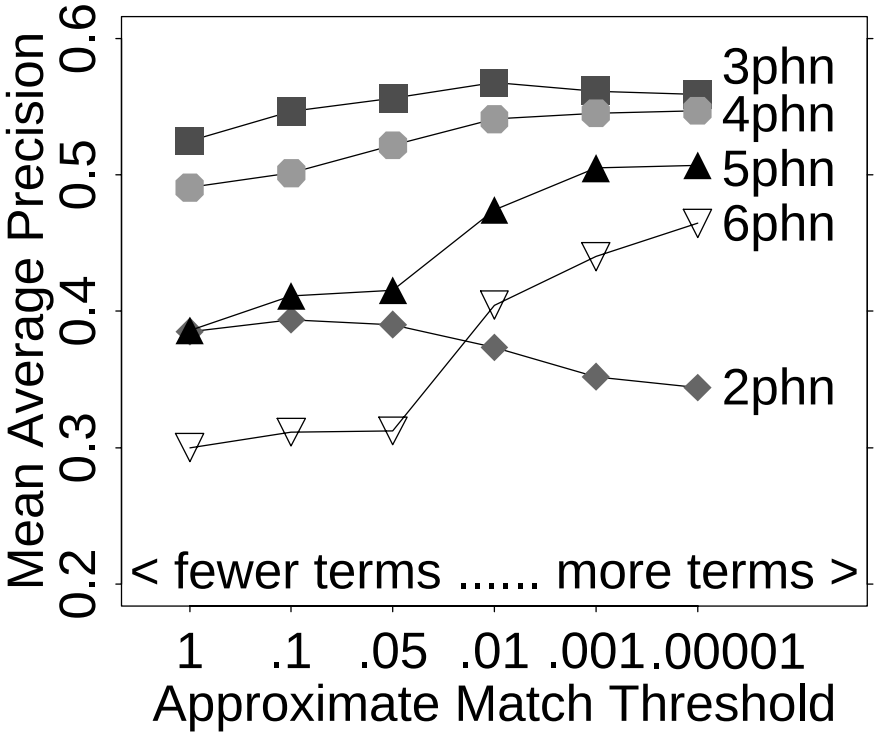


Figure 5-3: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units as the approximate match threshold is varied. More term matches are considered in the scoring process as the threshold value is lowered.

short ones as the threshold is lowered and more approximate match terms are considered. Again, the short subword units are more susceptible to spurious matches than the longer subword units.

The overall performance gains are better with approximate matching than with query expansion by adding "near-miss" terms. For example, using the new document-query metric, performance of the $n$=3 subword unit improves from mAP=0.52 to mAP=0.57. With the query expansion approach, however, performance only reaches mAP=0.54. There are several contributing factors. One is that the approximate match procedure considers all possible matches between the "clean" query terms and the "noisy" document terms whereas the "near-miss" terms generated using the query expansion approach generally only cover a subset of the noisy document terms. Not all terms that occur in the document may be included in the expanded query because only phone confusions that occur enough times

(above some threshold) are considered. Lowering the threshold addresses this problem, but then the number of terms in the query grows exponentially as we saw in Table 5-2. Generalizing the document-query retrieval metric to include approximate term matching allows us to consider all possible matches between the query and document terms without having to explicitly enumerate them *a priori* or to modify the query. Another factor is that the similarity measure used in the approximate match approach, $p(i \mid j)$ (Equation 5.5), is better than the similarity measure used in the query expansion approach, $s(i, j)$ (Equation 5.1). The quantity $p(i \mid j)$ is more general (allowing for insertion and deletion errors) and is probabilistically motivated whereas $s(i, j)$ is heuristically derived. Overall, implementing approximate match using the new document-query metric is superior to adding "near-miss" terms to the query.

Another way to view query expansion, approximate matching, and document expansion is the following. Let $\mathbf{q}$ be the original query vector, $\mathbf{d}$ a document vector, and $\mathbf{A}$ the matrix containing similarity measures between terms $i$ and $j$. To get the expanded query, we are essentially computing $\mathbf{qA}$. And to perform retrieval, we are basically computing $(\mathbf{qA})\,\mathbf{d}$ to get the similarity score to the document $\mathbf{d}$. Approximate matching can be interpreted as performing the same basic operations, $\mathbf{qAd}$. The main difference between query expansion and approximate matching is that the matrix $\mathbf{A}$ is different in the two cases. For query expansion, the values in matrix $\mathbf{A}$ are computed using $s(i, j)$ while $p(i \mid j)$ is used for approximate matching. Second, the set of non-zero entries in matrix $\mathbf{A}$, which affects which term pairs $i$ and $j$ actually contribute to the final retrieval score, is different. With query expansion, the term pairs are determined by the threshold on the confusion matrix entries. With approximate matching, only the cross product of the query and document terms are considered. Although these two sets have some overlap, as we discussed above, they are not usually the same. We can also view a document expansion approach which adds near miss-terms to the document representation (analogous to query expansion) as computing the value $\mathbf{q}\,(\mathbf{Ad})$. In this respect, query and document expansion are essentially equivalent provided $\mathbf{A}$ is the same. We point out that our document expansion approach, described in the next section, is different. It doesn't add near-miss terms to the document but rather includes terms that are found in high scoring recognition alternatives.

## 5.4 Expanding the Document Representation

A different approach is to modify the speech document representation by including high scoring recognition alternatives to increase the chance of capturing the correct hypothesis. This can be done by using the $N$-best recognition hypotheses, instead of just the single best one. If a term appears in many of the top $N$ hypotheses, it is more likely to have actually occurred than if it appears in only a few. As a result, a simple estimate of the frequency of term $i$ in document $d$, $f_d[i]$, can be obtained by considering the number of times, $n_i$, it appears in the top $N$ hypotheses:

$$f_d[i] = \frac{n_i}{N} \, . \tag{5.8}$$

We note that other information from the recognizer, such as likelihood and confidence scores, can also be used to weight our belief in the accuracy of different hypotheses. Associated with each recognition hypothesis is a likelihood score (i.e., the log probability of the term sequence given the acoustic signal). Each term in the hypothesis contributes a certain amount of likelihood to the overall score of the utterance. This term likelihood can be extracted and used, after appropriate normalization, as the weight of the term to reflect the recognizer's "belief" in that term hypothesis. Another estimate of the belief in the correctness of a term hypothesis can be obtained by computing associated confidence measures (Chase 1997; Siu et al. 1997).

Retrieval performance for the different subword units as the document representation is expanded to include the $N$-best recognition hypotheses is shown in Figure 5-4. Performance improves slightly for all the subword units as $N$ increases and then levels off after $N$=5 or 10. It appears that most of the useful term variants occur within the first few hypotheses; the addition of more recognition hypotheses after this does not help. One danger of using too many recognition hypotheses is that errorful terms from low scoring hypotheses may be included in the document representation. This can lead to spurious matches with query terms resulting in a decrease in retrieval precision. A mitigating factor is that these terms are likely to have low weights because of their small number of occurrences so their effect is minimized.

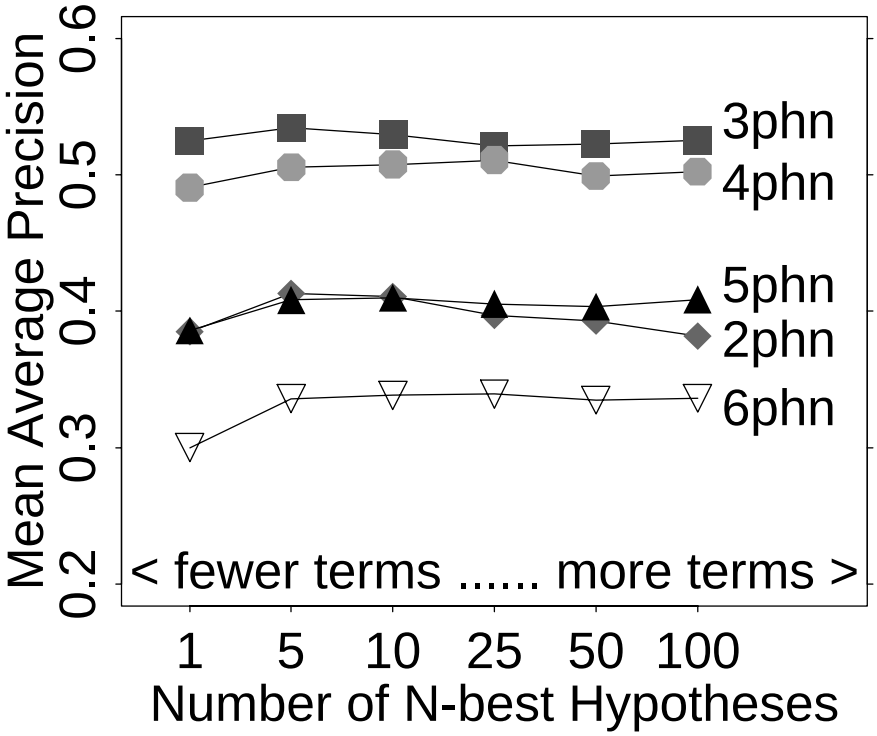


Figure 5-4: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units as the number of $N$-best recognition hypotheses used for document expansion is increased from N=1 to 100.

## 5.5 Query Modification via Automatic Relevance Feedback

The goal in relevance feedback is to iteratively refine a query by modifying it based on the results from a prior retrieval run. A commonly used query reformulation strategy, the Rocchio algorithm (Salton and McGill 1983), starts with the original query, $\mathbf{q}$, adds terms found in the retrieved relevant documents, and removes terms found in the retrieved non-relevant documents to come up with a new query, $\mathbf{q}'$:

$$\mathbf{q}' = \alpha\,\mathbf{q} + \beta\left(\frac{1}{N_r}\sum_{i\in D_r}\mathbf{d}_i\right) - \gamma\left(\frac{1}{N_n}\sum_{i\in D_n}\mathbf{d}_i\right) \tag{5.9}$$

where $D_r$ is the set of $N_r$ relevant documents, $D_n$ is the set of $N_n$ non-relevant documents, and $\alpha$, $\beta$, and $\gamma$ are tunable parameters controlling the relative contribution of the original, added, and removed terms, respectively. In addition to modifying the set of terms, the

above method also modifies their weights. The original term weights are scaled by $\alpha$; the added terms have a weight that is proportional to their average weight across the set of $N_r$ relevant documents; and the subtracted terms have a weight that is proportional to their average weight across the set of $N_n$ non-relevant documents. A threshold can also be placed on the number of new terms, $N_t$, that are added to the original query. Since there is no human user in the loop to label the initially retrieved documents as relevant and non-relevant, an automatic variation of the above strategy can be implemented by simply assuming that the top $N_r$ retrieved documents are relevant and the bottom $N_n$ documents are not relevant. Modifying the query in this way adds new terms from the top scoring documents from the first retrieval pass. These terms are ones that co-occur in the same documents with the original query terms. In addition, the query modification can potentially add approximate match terms that occur in the top ranked documents as well. We note that other query reformulation methods, such as probabilistic relevance feedback (Robertson and Jones 1976), can also be used.

Retrieval performance with ($\square$) and without ($\triangle$) the use of automatic relevance feedback is shown in Figure 5-5 for the different subword units. The following empirically derived relevance feedback parameters are used: $N_r$=1, $N_n$=10, $\alpha$=$\beta$=$\gamma$=1, and $N_t$=50. Performance is significantly improved for subword units of length $n$=3,4,5 but remains about the same for units of length $n$=2,6. This illustrates again the tradeoff advantages of intermediate length units.

Since automatic feedback helps improve performance in the case of noisy subword units, we also tried using automatic feedback with "clean" subword units derived from error free phonetic transcriptions and also word units derived from error-free text transcriptions. On clean phonetic subword units of length $n$=3, retrieval performance improves from mAP=0.86 to mAP=0.88 with the addition of automatic relevance feedback. With word-based units derived from error-free text, performance improves from mAP=0.87 (Section 3.3) to mAP=0.89 with the use of automatic feedback. Although automatic feedback improves performance in both cases, the gains are not as large as the ones obtained with noisy subword units.

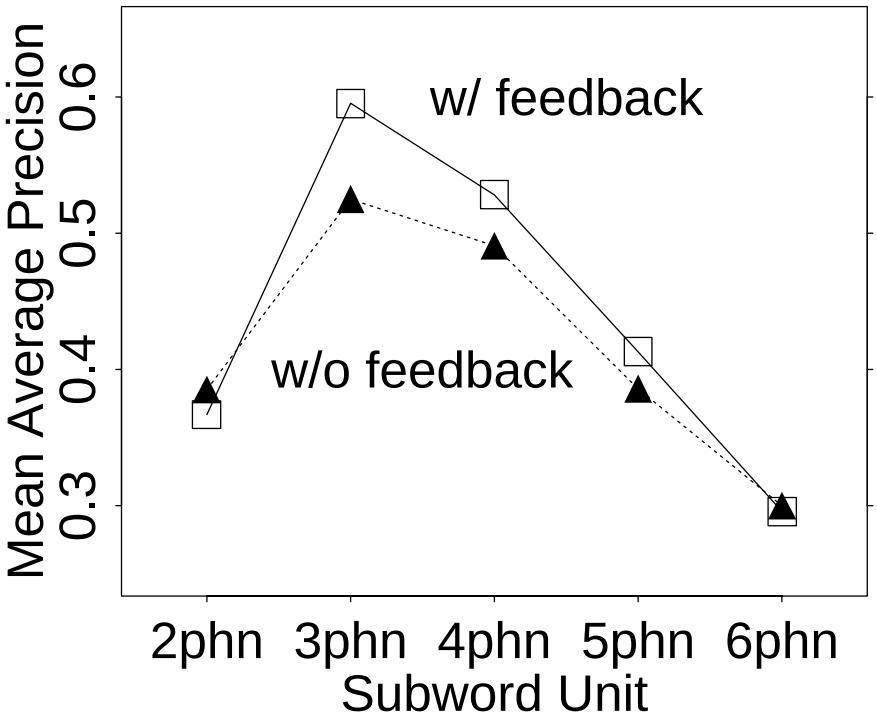


Figure 5-5: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units with and without using automatic relevance feedback.

## 5.6 Fusion of Multiple Subword Representations

Different subword unit representations can capture different types of information. For example, longer subword units can capture word or phrase information while shorter units can model word fragments. The tradeoff is that the shorter units are more robust to errors and word variants than the longer units but the longer units capture more discrimination information and are less susceptible to false matches. One simple way to try to combine the different information is to form a new document-query retrieval score by linearly combining the individual retrieval scores obtained from the separate subword units:

$$S_f(\mathbf{q}, \mathbf{d}) = \sum_n w_n \, S^n(\mathbf{q}, \mathbf{d}) \tag{5.10}$$

where $S^n(\mathbf{q}, \mathbf{d})$ is the document-query score (2.3) obtained using subword representation $n$ and $w_n$ is a tunable weight parameter. An alternate "fusion" method is to create a

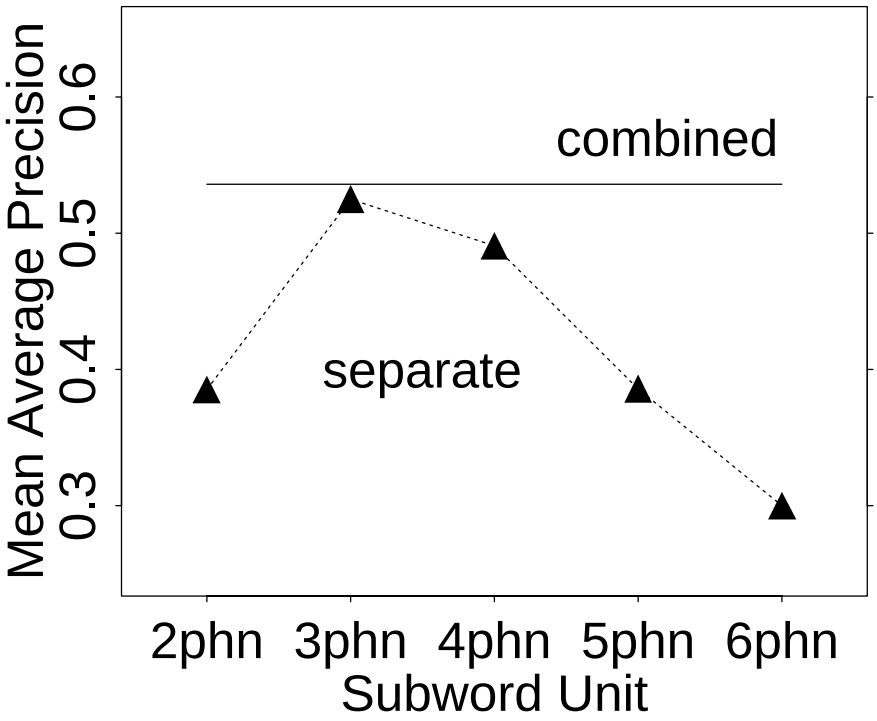


Figure 5-6: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units with and without using combination/fusion of different subword units.

heterogeneous set of indexing terms by pooling together the different subword units and performing a single retrieval run.

Subword unit combination should help if the separate subword units behave differently from each other. In particular, if the different subword units make different "errors" (i.e., different relevant documents score poorly for different subword units), then combining the results provides a chance to improve the results since some units are performing well. However, if all the subword units make the same "errors" (i.e., the same relevant documents score poorly for all representations), then combination would not expected to be useful. In addition, combination methods avoid the need to commit *a priori* to a single subword unit representation. However, they are more computationally expensive because multiple retrieval runs need to be performed and/or larger sets of terms need to be considered.

Linearly combining the separate retrieval scores with equal weights ($w_n$=0.2, $n$=2,3,4,5,6) results in performance that is slightly worse than just using the best performing subword

unit ($n$=3). Changing the weights to favor the better performing units ($w_3$=0.5, $w_4$=0.2, $w_{2,5,6}$=0.1) is slightly better than the $n$=3 subword unit as shown by the solid horizontal line in Figure 5-6 (mAP=0.536 vs. mAP=0.524). We also tried performing combination by simply pooling together the different subword units to create a heterogeneous set of indexing terms. In this case, the indexing terms consist of the union of phonetic subword units of lengths $n$=2, 3, 4, and 5. No preferential weighting of the different subword units is done. This method of combination results in only a very small change in the retrieval performance (mAP=0.526 vs. mAP=0.524). There may be less variability in the errors than desired for combination to be effective. In other words, the same relevant documents may be scoring poorly for many representations, in which case the benefit of subword unit combination is reduced. The use of more sophisticated non-linear combination methods such as bagging (Breiman 1994), boosting (Freund and Schapire 1996), or stacking (Wolpert 1992) might lead to better performance. However, with a more complex method with more parameters, the danger of over-fitting the training data and having a combination model that does not generalize well to new previously unseen data becomes more of an issue.

## 5.7 Combined use of the Robust Methods

Starting with the baseline retrieval performance of the different subword units, we cumulatively combine the various robust methods to see how performance improves. As shown in Figure 5-7, adding automatic relevance feedback (+fdbk) improves performance for the $n$=3,4,5 subword units. Using the approximate match retrieval metric (+approx) further improves performance for all subword units except for $n$=2. Expanding the documents using the top $N$=10 recognition hypotheses (+nbest) improves performance for the longer subword units. Finally, combining the scores of the different subword units (+combo) gives performance similar to that of the best performing subword unit ($n$=3). Since the approximate matching method is a generalization of the query expansion method and has better performance as we saw in Section 5.3, we only consider the approximate matching method in this cumulative combination. Table 5-3 lists the performance improvements for the $n$=3 subword unit using the various various robust methods individually and in combination.

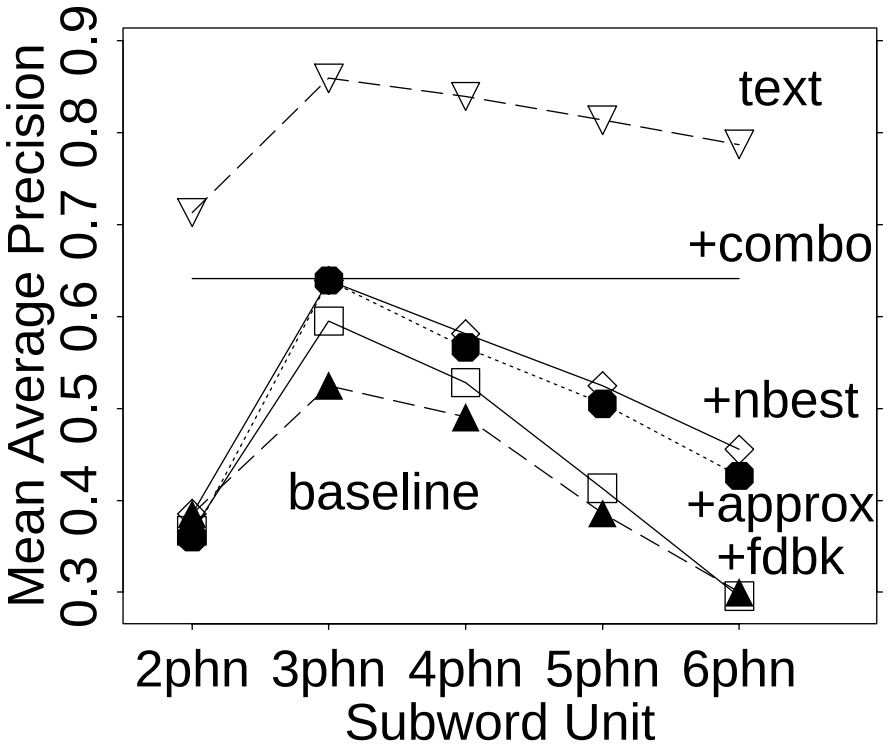


Figure 5-7: Retrieval performance for different length ($n = 2, \ldots, 6$) phonetic subword units as the different robust methods are combined. Performance improves from the baseline as relevance feedback (+fdbk), approximate matching (+approx), N-best document expansion (+nbest), and combination/fusion of different subword units (+combo) is applied. Performance using subwords generated from clean phonetic transcriptions (text) is still better.

Each of the methods used individually is able to improve performance above the baseline. When combined, many of the performance gains are additive with the automatic relevance feedback (+fdbk) and the approximate match retrieval metric (+approx) contributing the most. The final result is that information retrieval performance, measured in mean average precision, improves from mAP=0.52 (for the initial $n$=3 subword unit) to mAP=0.64, a gain of about 23%. There remains, however, a large performance gap when compared to subword units derived from error-free phonetic transcriptions (text). Performance of the clean subword units with automatic relevance feedback processing is also shown (text+fdbk).

| | Mean Average Precision | |
|---|---|---|
| **Condition** | **Individually** | **Combined** |
| **baseline** | 0.524 | 0.524 |
| **+fdbk** | 0.595 | 0.595 |
| **+approx** | 0.568 | 0.639 |
| **+nbest** | 0.535 | 0.640 |
| **+combo** | 0.536 | 0.641 |
| **text** | 0.859 | 0.859 |
| **text+fdbk** | 0.878 | 0.878 |

Table 5-3: Information retrieval performance (measured in mean average precision) for the $n$=3 phonetic subword unit for the different robust methods used individually and in combination. Performance improves from the baseline as relevance feedback (+fdbk), approximate matching (+approx), N-best document expansion (+nbest), and combination/fusion of different subword units (+combo) is applied. Using subwords generated from clean phonetic transcriptions with (text+fdbk) and without (text) feedback is still better.

## 5.8 Summary

In this chapter, we investigated a number of robust methods in an effort to improve spoken document retrieval performance when there are speech recognition errors. In the first approach, the original query is modified to include near-miss terms that could match erroneously recognized speech. The second approach involves developing a new document-query retrieval measure using approximate term matching designed to be less sensitive to speech recognition errors. In the third method, the document is expanded to include multiple recognition candidates to increase the chance of capturing the correct hypothesis. The fourth method modifies the original query using automatic relevance feedback to include new terms as well as approximate match terms. The last method involves combining information from multiple subword unit representations. We studied the different methods individually and then explored the effects of combining them. We found that using a new approximate match retrieval metric, modifying the queries via automatic relevance feedback, and expanding the documents with $N$-best recognition hypotheses improved performance; subword unit fusion, however, resulted in only marginal gains. Combining the approaches resulted in additive performance improvements. Using these robust methods improved retrieval performance using subword units generated from errorful phonetic recognition transcriptions by 23%.

# Chapter 6

# A Maximum Likelihood Ratio Information Retrieval Model

In this chapter, we take a brief digression from spoken document retrieval and describe a novel probabilistic retrieval model that we developed as part of our work (Ng 1999). We also move from spoken document retrieval to traditional text retrieval in order to benchmark the performance of our probabilistic retrieval model on standard text retrieval tasks and to allow a comparative evaluation of our model to other retrieval approaches. We return to speech retrieval in the next chapter, Chapter 7, when we use this probabilistic retrieval model to implement an approach to spoken document retrieval where the speech recognition and information retrieval components are more tightly integrated.

Probabilistic modeling for information retrieval (IR) has a long history (Crestani et al. 1998). Many of these approaches try to evaluate the probability of a document being relevant ($R$) to a given query $Q$ by estimating $\mathrm{p}(R \,|\, Q, D_i)$ for every document $D_i$ in the collection. These relevance probabilities are then used to rank order the retrieved documents. However, due to the imprecise definition of the concept of relevance and the lack of available relevance training data, reliably estimating these probabilities has been a difficult task. Because of the the nature of the IR task, training data in the form of document-query pairs labeled with their corresponding relevance judgments is not generally available *a priori*. Previously seen queries, for which relevance information can be created, can be used for

training but their applicability to new queries is not clear. Some relevance information can be obtained in a multi-pass retrieval strategy by using relevance feedback. However, only a small number of relevance judgments is typically generated. Many of these probabilistic methods are better suited for related applications, such as information filtering, where more relevance training data is available (Harman 1997; Harman 1998).

Instead of the imprecisely defined notion of relevance, we consider the better defined measure of likelihood. In particular, we examine the relative change in the likelihood of a document before and after a query is specified, and use that as the metric for scoring and ranking the documents. The idea is that documents that become more likely after the query is specified are probably more useful to the user and should score better and be ranked ahead of those documents whose likelihoods either stay the same or decrease. The document likelihoods are computed using statistical language modeling techniques and the model parameters are estimated automatically and dynamically for each query to optimize well-specified objective functions.

In the following sections, we first discuss some related modeling approaches. Next, we derive the basic retrieval model, describe the details of the model, and present some extensions to the model including a method to perform automatic feedback. Then, we evaluate the performance of the retrieval model and present experimental results on the TREC-6 and TREC-7 ad hoc text retrieval tasks. Official evaluation results on the 1999 TREC-8 ad hoc retrieval task are also reported. Experimental results indicate that the model is able to achieve performance that is competitive with current state-of-the-art retrieval approaches.

## 6.1 Related Work

In our retrieval model, we use the relative change in the likelihood of a document $D_i$ before and after the user query $Q$ is specified, expressed as the likelihood ratio of the conditional and the prior probabilities, $\frac{\mathrm{p}(D_i|Q)}{\mathrm{p}(D_i)}$, as the metric for scoring and ranking the documents. A document that becomes more likely after the query is specified is probably more useful to the user than one that either remains the same or becomes less likely. This score can be equivalently rewritten as $\frac{\mathrm{p}(Q|D_i)}{\mathrm{p}(Q)}$. Since we need to estimate $\mathrm{p}(Q|D_i)$, the probability of

query $Q$ given document $D_i$, our model is related to several recently proposed IR approaches which also make use of this probabilistic quantity (Hiemstra and Kraaij 1998; Miller et al. 1998; Ponte and Croft 1998).

In (Ponte and Croft 1998) and (Hiemstra and Kraaij 1998), a language modeling argument is used to directly posit that $\mathrm{p}(Q|D_i)$ is an appropriate quantity for scoring document $D_i$ in response to query $Q$. Mixture models are then used to compute this quantity. In (Miller et al. 1998), the probability that document $D_i$ is relevant given query $Q$, $\mathrm{p}(D_i \,\mathrm{is}\, R|Q)$, is used to score the documents. This quantity can be rewritten, using Bayes rule, as $\frac{\mathrm{p}(Q|D_i \,\mathrm{is}\, R)\,\mathrm{p}(D_i \,\mathrm{is}\, R)}{\mathrm{p}(Q)}$. A generative hidden Markov model (HMM) is then used to compute the quantity $\mathrm{p}(Q|D_i \,\mathrm{is}\, R)$.

Although our retrieval model shares this commonality with these other approaches, there are some important differences. First, as described above, our model is derived starting from a different theoretical justification. Second, different modeling assumptions and estimation techniques are used to determine the underlying probabilistic quantities. Although we use the standard technique of mixture models to estimate $\mathrm{p}(Q|D_i)$ (See Section 6.2.1), the underlying probabilistic components in our mixture model are different from those used in (Ponte and Croft 1998) and (Hiemstra and Kraaij 1998). We back-off to the term's probability of occurrence in the entire document collection. In (Hiemstra and Kraaij 1998), the back-off is to the term's document frequency while in (Ponte and Croft 1998) the back-off is a scaled version of the term's mean probability of occurrence in documents that contain the term. We also automatically estimate the mixture model parameters dynamically (for each query $Q$) to maximize the likelihood of the query given a set of top scoring documents $\{D_i\}$ from the current document collection. This is done by using an iterative process starting with initial estimates of the mixture model parameters. This approach is in contrast to the standard approach of determining static, query-independent, mixture model parameter values by empirically tuning on an old development set. In addition, we attempt to deal with unobserved query terms in a more principled way by using Good-Turing techniques to smooth the underlying probability models. Finally, we develop a new automatic relevance feedback strategy that is specific to our probabilistic model (See Section 6.2.2). The procedure automatically creates a new query (based on the

original query and a set of top-ranked documents from a preliminary retrieval pass) that optimizes a well-specified objective function. In particular, the term selection and the term weight estimation procedures are designed to maximize the likelihood ratio scores of the set of documents presumed to be relevant to the query. Hopefully, improving these scores will lead to improved retrieval performance.

## 6.2 Information Retrieval Model

Given a collection of $n$ documents, $\{D_i\}_{i=1}^{n}$, each document $D_i$ has a prior likelihood given by $\mathrm{p}(D_i)$. After a query $Q$ is specified by a user, the likelihood of each document changes and becomes that given by the conditional probability: $\mathrm{p}(D_i|Q)$. Some documents will become more likely after the query is specified while others will either remain the same or become less likely. The documents that become more likely are probably more useful to the user and should score better and be ranked ahead of those that either stay the same or become less likely. As a result, we propose to use the relative change in the document likelihoods, expressed as the likelihood ratio of the conditional and the prior probabilities, as the metric for scoring and ranking the documents in response to query $Q$:

$$\mathrm{S}(D_i, Q) = \frac{\mathrm{p}(D_i|Q)}{\mathrm{p}(D_i)} \tag{6.1}$$

We illustrate the idea with a simple example. Suppose we have two documents in our collection: $D_1$ ={`blue house`} and $D_2$ ={`magenta house`}. If we use a simple unigram language model for general English, $\Lambda$, to compute the document likelihoods, document $D_1$ will be more likely than $D_2$, $\mathrm{p}(D_1 \,|\, \Lambda) > \mathrm{p}(D_2 \,|\, \Lambda)$, because the word "`blue`" occurs more frequently than "`magenta`": $\mathrm{p}(\texttt{blue} \,|\, \Lambda) > \mathrm{p}(\texttt{magenta} \,|\, \Lambda)$. After a query is specified, the language model is modified to reflect the contents of the query and becomes $\Lambda'$. For example if the specified query is $Q$ ={`magenta`}, the general English model will be modified to make the word "`magenta`" more likely: $\mathrm{p}(\texttt{magenta} \,|\, \Lambda') > \mathrm{p}(\texttt{magenta} \,|\, \Lambda)$. In the new language model, the probability of the word "`blue`" will either remain the same or decrease depending on how $\Lambda$ is transformed to $\Lambda'$: $\mathrm{p}(\texttt{blue} \,|\, \Lambda') \leq \mathrm{p}(\texttt{blue} \,|\, \Lambda)$. As a result, $\mathrm{p}(D_2 \,|\, \Lambda') > \mathrm{p}(D_2 \,|\, \Lambda)$ and $\mathrm{p}(D_1 \,|\, \Lambda') \leq \mathrm{p}(D_1 \,|\, \Lambda)$ which means $D_2$ will have a higher score than $D_1$ as a result of

specifying the query: $\mathrm{S}(D_2, Q) > \mathrm{S}(D_1, Q)$.

We can decompose the likelihood ratio score in Equation 6.1 into more easily estimated components using Bayes' rule, rewriting it as follows:

$$\mathrm{S}(D_i, Q) = \frac{\mathrm{p}(Q|D_i)\,\mathrm{p}(D_i)/\mathrm{p}(Q)}{\mathrm{p}(D_i)} = \frac{\mathrm{p}(Q|D_i)}{\mathrm{p}(Q)} \tag{6.2}$$

where $\mathrm{p}(Q|D_i)$ is the probability of query $Q$ given document $D_i$ and $\mathrm{p}(Q)$ is the prior probability of query $Q$. Each document $D_i$ specifies a different language model $\Lambda_i$. We can view $\mathrm{p}(Q|D_i)$ as the probability that query $Q$ is generated by $\Lambda_i$, the language model associated with document $D_i$. This means that our goal during the retrieval process is to find those documents in the collection that maximize the likelihood of the query. These documents should be the ones that are most useful to the user who specified query $Q$.

The $\mathrm{p}(Q)$ term represents the probability that query $Q$ is generated from a document independent (general) language model $\Lambda$, and serves as a normalization factor. Since $\mathrm{p}(Q)$ is constant for all documents $D_i$ given a specific query $Q$, it does not affect the ranking of the documents and can be safely removed from the scoring function. However, this $\mathrm{p}(Q)$ normalization factor is useful if we want a meaningful interpretation of the scores (as a relative change in the likelihood) and if we want to be able to compare scores across different queries. In Section 6.3.2, we illustrate the usefulness of $\mathrm{p}(Q)$ for these purposes. In addition, the $\mathrm{p}(Q)$ normalization factor is an important part of the automatic feedback extension to the basic model as we will see in Section 6.2.2. For these reasons, we will keep the $\mathrm{p}(Q)$ term in the scoring function in (6.2).

### 6.2.1 Retrieval Model Details

In order to compute the score in (6.2), we need to be able to estimate the quantities $\mathrm{p}(Q|D_i)$ and $\mathrm{p}(Q)$. To do this, we make the assumption that the query $Q$ is drawn from a multinomial distribution over the set of possible terms in the corpus and document $D_i$ specifies the parameters of the multinomial model. This gives us the following estimates for

$\mathrm{p}(Q|D_i)$ and $\mathrm{p}(Q)$:

$$\mathrm{p}(Q|D_i) = \frac{n!}{\prod_{t=1}^{k} c_t!} \prod_{t=1}^{k} \mathrm{p}(t|D_i)^{c_t} \tag{6.3}$$

$$\mathrm{p}(Q) = \frac{n!}{\prod_{t=1}^{k} c_t!} \prod_{t=1}^{k} \mathrm{p}(t)^{c_t} \tag{6.4}$$

where $c_t$ is the number of times term $t$ occurs in query $Q$, $k$ is the number of distinct terms in the corpus, $n = \sum_{t=1}^{k} c_t$ is the total number of terms in query $Q$, $\mathrm{p}(t|D_i)$ is the probability of query term $t$ occurring in document $D_i$ with the constraint $\sum_{t=1}^{k} \mathrm{p}(t|D_i) = 1$, and $\mathrm{p}(t)$ is the probability of query term $t$ occurring in the document collection with the constraint $\sum_{t=1}^{k} \mathrm{p}(t) = 1$. Substituting (6.3) and (6.4) into (6.2) and simplifying (noting that $c_t! = 1$ for $c_t = 0$), we have:

$$\mathrm{S}(D_i, Q) = \prod_{t=1}^{k} \left( \frac{\mathrm{p}(t|D_i)}{\mathrm{p}(t)} \right)^{c_t} \tag{6.5}$$

Since $x^0 = 1$ for all $x$, the product over all $k$ terms can be replaced by a product over only the terms that occur in the query:

$$\mathrm{S}(D_i, Q) = \prod_{t \in Q} \left( \frac{\mathrm{p}(t|D_i)}{\mathrm{p}(t)} \right)^{c_t} \tag{6.6}$$

To simplify computation and to prevent numerical underflows, we perform the score computation in the log domain:

$$\mathrm{S}_l(D_i, Q) = \log \mathrm{S}(D_i, Q) = \sum_{t \in Q} c_t \log \left( \frac{\mathrm{p}(t|D_i)}{\mathrm{p}(t)} \right) \tag{6.7}$$

We note that since the logarithm is a monotonic transformation, the rank ordering of the documents using the log score remains the same as that using the original score.

In the original multinomial model, $c_t$ is the number of times term $t$ occurs in query $Q$ and can only take on integral values: $c_t = 0, 1, \ldots, n$. We would like to generalize $c_t$ so that it can take on non-negative real values. This will allow more flexible weighting of the query terms including the use of fractional counts which will be useful in our automatic relevance feedback extension (Section 6.2.2) and query section weighting (Section 6.3.5). To indicate

this generalization in the scoring function, we replace $c_t$ in (6.7) with $q(t)$, which can be interpreted as the weight of term $t$ in query $Q$:

$$\mathrm{S}_l(D_i, Q) = \sum_{t \in Q} q(t)\ \log\left(\frac{\mathrm{p}(t|D_i)}{\mathrm{p}(t)}\right) \tag{6.8}$$

This generalization does not affect the ranking of the documents since it is equivalent to adding a query-dependent constant multiplicative factor, $1/n$, to the score in (6.7) to convert the $c_t$ counts to the $q(t)$ numbers. In fact, we can interpret $q(t)$ as $\mathrm{p}(t|Q)$, the probability of term $t$ occurring in query $Q$, if $q(t) = c_t/n$ where $n = \sum_t c_t$.

We note that the scoring function in (6.8) can be related to the Kullback-Leibler distance (Cover and Thomas 1991), which is an information theoretic measure of the divergence of two probability distributions $\mathrm{p}_1(x)$ and $\mathrm{p}_2(x)$ given by:

$$KL(\mathrm{p}_1(x), \mathrm{p}_2(x)) = -\sum_x \mathrm{p}_2(x) \log\left(\frac{\mathrm{p}_1(x)}{\mathrm{p}_2(x)}\right) \tag{6.9}$$

To show this relationship, we start by rewriting (6.8) as follows:

$$\mathrm{S}_l(D_i, Q) = \sum_{t \in Q} q(t)\ \log \mathrm{p}(t|D_i) - \sum_{t \in Q} q(t)\ \log \mathrm{p}(t) \tag{6.10}$$

Next, we add in and subtract out $\sum_{t \in Q} q(t)\ \log q(t)$ and rearrange terms to get:

$$\mathrm{S}_l(D_i, Q) = \tag{6.11}$$
$$\left(\sum_{t \in Q} q(t)\ \log \mathrm{p}(t|D_i) - \sum_{t \in Q} q(t)\ \log q(t)\right) - \left(\sum_{t \in Q} q(t)\ \log \mathrm{p}(t) - \sum_{t \in Q} q(t)\ \log q(t)\right)$$

Finally, we collapse the terms in the parentheses to obtain:

$$\mathrm{S}_l(D_i, Q) = \underbrace{\sum_{t \in Q} q(t)\ \log\left(\frac{\mathrm{p}(t|D_i)}{q(t)}\right)}_{-KL(q(t), \mathrm{p}(t|D_i))} \underbrace{- \sum_{t \in Q} q(t)\ \log\left(\frac{\mathrm{p}(t)}{q(t)}\right)}_{+KL(q(t), \mathrm{p}(t))} \tag{6.12}$$

Recall that $q(t)$ can be interpreted as $\mathrm{p}(t|Q)$, the probability of term $t$ in query $Q$, $\mathrm{p}(t|D_i)$ is the probability of term $t$ in document $D_i$, and $\mathrm{p}(t)$ is the probability of term $t$ in the

general language (i.e., using a document-independent model). The first term in (6.12) is the (negative) KL divergence between the term distribution of query $Q$ and document $D_i$. If the two term distributions are identical, then the divergence will be zero. As the difference between the query and document distributions becomes greater, the divergence increases, and the score decreases (because of the negative sign on the term). The second term is the KL divergence between the term distribution of query $Q$ and a general document-independent model. Since this term doesn't depend on the document, it has no effect on the rankings of the retrieved documents; it only serves as a bias or normalization factor. It is query-dependent and only comes into play if we compare scores across different queries.

We also note that the scoring function in (6.8) has the form of the standard vector space model. It consists of the sum over all terms $t$ in the query of the product of a query dependent factor, $q(t)$, and a document dependent factor, $\log\left(\frac{\mathrm{p}(t|D_i)}{\mathrm{p}(t)}\right)$. It turns out that many probabilistic models can be expressed in the standard vector space model format (Crestani et al. 1998; Hiemstra and Kraaij 1998; Salton and McGill 1983). The models differ in what the query and document factors are and how they are estimated.

Next, we need to estimate the probabilities $\mathrm{p}(t|D_i)$ and $\mathrm{p}(t)$. We start by considering their maximum likelihood (ML) estimates which are given by:

$$\mathrm{p}_{\mathrm{ml}}(t|D_i) = \frac{d_i(t)}{\sum_{t=1}^{k} d_i(t)} \tag{6.13}$$

$$\mathrm{p}_{\mathrm{ml}}(t) = \frac{\sum_{i=1}^{n} d_i(t)}{\sum_{i=1}^{n}\sum_{t=1}^{k} d_i(t)} \tag{6.14}$$

where $d_i(t)$ is the number of occurrences of term $t$ in document $D_i$, $k$ is the number of distinct terms in the corpus, and $n$ is the number of documents in the collection.

With a large document collection, there is enough data for $\mathrm{p}_{\mathrm{ml}}(t)$ to be robustly estimated. However, this ML estimate will assign a probability of zero to terms that do not occur in the document collection. To avoid this undesirable property, we can use Good-Turing (GT) methods to estimate $\mathrm{p}(t)$ (Jelinek 1999). GT methods provide probability estimates for both observed and unobserved terms with the constraint that the total probability of all terms must sum to one. For unobserved terms, GT methods provide an estimate of the *total* probability of these terms. This total probability can then be divided among the

possible unobserved terms to provide per term probability estimates. For observed terms, GT methods provide probability estimates for these terms that are consistent with estimating non-zero probabilities for the unobserved terms. This is done by reducing the total probability of the observed terms to be less than one such that the sum of the probabilities assigned to all terms, both observed and unobserved, equals one.

Good-Turing methods work as follows. If a certain term $t$ occurs $r$ times in the document collection, the ML estimate of $\mathrm{p}(t)$ is given by:

$$\mathrm{p}_{\mathrm{ml}}(t) = \frac{r}{N} \tag{6.15}$$

where $N$ is the total number of terms observed in the document collection. With GT estimation, the count $r$ is replaced by a modified count $r^*$ which is calculated as:

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \tag{6.16}$$

where $N_r$ is the number of terms that occurs exactly $r$ times in the document collection. As a result, the GT estimate of $\mathrm{p}(t)$ for observed terms is given by:

$$\mathrm{p}_{\mathrm{gt}}(t) = \mathrm{p}_r = \frac{r^*}{N} \tag{6.17}$$

where $N = \sum_r rN_r$ is the total number of terms observed in the document collection. The GT estimate for the *total* probability of unobserved terms is given by:

$$\mathrm{p}_0 = \frac{N_1}{N} \tag{6.18}$$

This total probability is then divided equally among the possible unobserved terms to provide per term probability estimates. Using the observed $N_r$ values to calculate $r^*$ in (6.16) can become problematic if $N_r = 0$ for some $r$. As a result, it is necessary to pre-smooth $N_r$ so that it never equals zero. There are many different possible smoothing methods and each gives rise to a slightly different GT approach. We use the Simple Good-Turing (SGT) approach described in (Gale and Sampson 1995). Basically $N_r$ is linearly smoothed (in the log domain) and a decision rule is used to decide when to switch from

using the observed $N_r$ values to the smoothed values. Details of the SGT method can be found in (Gale and Sampson 1995).

Unlike the estimate for $\mathrm{p}(t)$, the quantity $\mathrm{p_{ml}}(t|D_i)$ is likely to be poorly estimated regardless of the size of the document collection because of the limited size of the individual documents. Many of the terms in the model will have zero probability. There are many different ways to compensate for this sparse data problem. One approach is to model the term distributions using parametric distributions such as Beta and Dirichlet distributions. A standard statistical language modeling approach, and the one we adopt, is to linearly interpolate the more detailed $\mathrm{p_{ml}}(t|D_i)$ model with a better estimated, but more general model, for example, $\mathrm{p_{gt}}(t)$ (Jelinek 1999):

$$\mathrm{p}(t|D_i) = \alpha\, \mathrm{p_{ml}}(t|D_i) + (1-\alpha)\, \mathrm{p_{gt}}(t) \tag{6.19}$$

where $\alpha$ is the mixture weight. The estimate-maximize (EM) algorithm (Dempster et al. 1977) is used to estimate $\alpha$ to maximize the (log) likelihood of query $Q$ given document $D_i$:

$$\begin{aligned} \alpha^* &= \arg\max_{\alpha} \log\left(\mathrm{p}(Q|D_i)\right) && (6.20) \\ &= \arg\max_{\alpha} \sum_{t\in Q} q(t)\, \log\left(\alpha\, \mathrm{p_{ml}}(t|D_i) + (1-\alpha)\, \mathrm{p_{gt}}(t)\right) && (6.21) \end{aligned}$$

In the above formulation, there is a different $\alpha$ for each document $D_i$. To simplify the model and to provide more data for parameter estimation, we can "tie" the $\alpha$ weight across the documents so that there is only a single, document-independent, $\alpha$ for each query $Q$. The following iterative procedure can then be used to estimate $\alpha$:

1. Initialize $\alpha$ to a random estimate between 0 and 1.

2. Update $\alpha$:

$$\alpha' = \frac{1}{\sum_{t\in Q}\sum_{i\in\mathcal{I}_Q} q(t)} \sum_{t\in Q}\sum_{i\in\mathcal{I}_Q} q(t)\, \frac{\alpha\, \mathrm{p_{ml}}(t|D_i)}{\alpha\, \mathrm{p_{ml}}(t|D_i) + (1-\alpha)\, \mathrm{p_{gt}}(t)}$$

3. If $\alpha$ has converged (i.e., $|\alpha' - \alpha| < \delta$ for some small threshold $\delta$) then stop. Otherwise, set $\alpha = \alpha'$ and goto step 2.

In this procedure, $\mathcal{I}_Q$ contains the indices of the set of documents used to estimate $\alpha$ for query $Q$. We need to decide which documents should be in this set. If we use *all* the documents in the collection (i.e., $\mathcal{I}_Q = \{1, \ldots, n\}$), the query terms will occur so seldomly in the entire collection that $\alpha$ will almost always be set to zero. That would not be very useful. What we want is a reasonable estimate of $\alpha$ for those documents that are likely to be relevant to the query since they are the ones that we are interested in. Ideally, we want the set of documents to be those that *are* relevant to query $Q$. However, since this information is not available, we need to use an approximation. One approach is to borrow the technique used in automatic relevance feedback (Salton and McGill 1983) (see Section 6.2.2). Basically, we perform a preliminary retrieval run using an initial guess for $\alpha$ (e.g., $\alpha = 0.5$) and assume that the top $M$ retrieved documents are relevant to the query. These $M$ top-scoring documents then become the set we use to estimate the $\alpha$ weight for query $Q$. A typical value for the number of documents that we use is $M = 5$.

Using the approach described above, a separate $\alpha$ is estimated for each query $Q$. If desired, one can pool the query terms across all the queries and estimate a single query-independent $\alpha$. It is important to note that the above procedure estimates the mixture parameters dynamically using the current query and the current document collection. This is in contrast to the standard approach of determining static, query-independent, model parameter values by empirically tuning on an old development set which typically consists of a different set of queries and potentially a different collection of documents. In Section 6.3.3, we explore the effect of different estimated $\alpha$ values on retrieval performance and examine query-specific and query-independent $\alpha$'s.

In summary, the final metric used for scoring document $D_i$ in response to query $Q$ is obtained by substituting the estimates for $\mathrm{p}(t)$ and $\mathrm{p}(t|D_i)$ (Equations 6.17 and 6.19, respectively) into (6.8):

$$\mathrm{S}_l(D_i, Q) = \sum_{t \in Q} q(t) \, \log \left( \frac{\alpha \, \mathrm{p}_{\mathrm{ml}}(t|D_i) + (1 - \alpha) \, \mathrm{p}_{\mathrm{gt}}(t)}{\mathrm{p}_{\mathrm{gt}}(t)} \right) \tag{6.22}$$

### 6.2.2 Automatic Relevance Feedback

Automatic relevance feedback is a proven method for improving information retrieval performance (Harman 1997). As we discussed before, the process works in three steps. First, the original query is used to perform a preliminary retrieval run. Second, information from these retrieved documents are used to automatically construct a new query. Third, the new query is used to perform a second retrieval run to generate the final results. A commonly used query reformulation strategy, the Rocchio algorithm (Salton and McGill 1983), starts with the original query, $Q$, then adds terms found in the top $N_t$ retrieved documents and subtracts terms found in the bottom $N_b$ retrieved documents to come up with a new query, $Q'$. Modifying the query in this way adds new terms that occur in documents that are likely to be relevant to the query and eliminates terms that occur in documents that are probably non-relevant. The goal is to improve the ability of the query to discriminate between relevant and non-relevant documents.

We extend our basic retrieval model to include an automatic relevance feedback processing stage by developing a new query reformulation algorithm that is specific to our probabilistic model. Recall that in our retrieval model, we score document $D_i$ in response to query $Q$ using the likelihood ratio score (6.2):

$$\mathrm{S}(D_i, Q) = \frac{\mathrm{p}(Q|D_i)}{\mathrm{p}(Q)} \tag{6.23}$$

Since the documents are ranked based on descending values of this score, we can view the goal of the automatic feedback procedure as trying to create a new query $Q'$ (based on the original query $Q$ and the documents retrieved from the preliminary retrieval pass) such that the score using the new query is better than the score using the original query for those documents $D_i$ that are relevant to the query:

$$\frac{\mathrm{p}(Q'|D_i)}{\mathrm{p}(Q')} \geq \frac{\mathrm{p}(Q|D_i)}{\mathrm{p}(Q)} \qquad \text{for } i \in \mathcal{I}_Q \tag{6.24}$$

Because $\mathcal{I}_Q$, the set of relevant documents for query $Q$, is not known, we use an approximation and assume that the top scoring documents from a preliminary retrieval run using

the original query are relevant. There are many different ways to decide which of the top scoring documents to select. One approach is to simply select a fixed number, $M$, of the top scoring documents. One concern with this approach is that the selected documents can have very disparate scores. There can be a big score difference between the first and the $M^{\text{th}}$ document. Another approach is to use an absolute score threshold, $\theta$, so only documents with scores above $\theta$ are selected. With this approach, it is possible to not have any documents that score above the threshold. A different approach, and the one we adopt, is to use a relative score threshold, $\gamma \leq 1$, so documents that score within a factor of $\gamma$ of the top scoring document are selected:

$$\text{select } D_i \text{ if } \quad \frac{\mathrm{S}(D_i, Q)}{\max\limits_{D_i} \mathrm{S}(D_i, Q)} \leq \gamma \tag{6.25}$$

This results in a variable number of documents for each query, but the selected documents are guaranteed to have similar scores. A typical threshold value is $\gamma = 0.75$.

Since we want to improve the score for all the documents in the set $\mathcal{I}_Q$ simultaneously, we need to deal with the set of documents jointly. One way to do this is to create a new joint document $D'$ by pooling together all the documents in the set $\mathcal{I}_Q$ so the number of occurrences of term $t$ in the joint document $D'$ is given by:

$$d'(t) = \sum_{i \in \mathcal{I}_Q} d_i(t) \tag{6.26}$$

Another variation is to weight the contribution of each document, $D_i$, by its preliminary retrieval score, $\mathrm{S}(D_i, Q)$, so documents that score better have more impact:

$$d'(t) = \sum_{i \in \mathcal{I}_Q} \mathrm{S}(D_i, Q)\, d_i(t) \tag{6.27}$$

Using this new joint document, $D'$, the inequality in (6.24) becomes:

$$\frac{\mathrm{p}(Q'|D')}{\mathrm{p}(Q')} \geq \frac{\mathrm{p}(Q|D')}{\mathrm{p}(Q)} \tag{6.28}$$

Substituting our models for the conditional and prior probabilities and working in the log

domain (Equation 6.8), we have:

$$\sum_{t \in Q'} q'(t) \log \left( \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} \right) \geq \sum_{t \in Q} q(t) \log \left( \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} \right) \tag{6.29}$$

Let us consider the creation of the new query $Q'$ in two steps. First, let us examine which terms should be *removed* from the original query $Q$ in order to improve the score. Second, we can then examine which terms from the joint document $D'$ should be *added* to the query to further improve the score.

Starting with the original query $Q$, we consider each query term $t$ and determine whether it should be included or excluded from the new query $Q'$. Since the query term weights $q(t)$ are constrained to be greater than zero, the only way that a query term $t$ can decrease the score is if $\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} < 1$. Therefore, if we exclude such terms from the new query $Q'$ (while keeping the term weights the same, i.e., $q'(t) = q(t)$), we can be assured that the inequality in (6.29) is satisfied. This selection criteria makes intuitive sense since it basically states that query terms that occur more frequently in the general collection than in the pooled document $D'$ (which is created from assumed relevant documents) should not be used.

Next, we consider which terms from the joint document $D'$ should be included to the query $Q'$ in order to further improve the score. Following the same arguments as those used above, and noting that $q'(t) > 0$, we see that only terms $t$ for which $\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} > 1$ can increase the score. As a result, we will only add those terms from $D'$ that satisfy this property. Using this term selection criteria, we maintain the inequality in (6.29) with each newly included term. Substituting the estimates for $\mathrm{p}(t)$ and $\mathrm{p}(t|D_i)$ (Equations 6.17 and 6.19, respectively), the term selection criteria becomes:

$$\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} > 1 \tag{6.30}$$

$$\frac{\alpha\, \mathrm{p}_{\mathrm{ml}}(t|D') + (1-\alpha)\, \mathrm{p}_{\mathrm{gt}}(t)}{\mathrm{p}_{\mathrm{gt}}(t)} > 1$$

$$\alpha \frac{\mathrm{p}_{\mathrm{ml}}(t|D')}{\mathrm{p}_{\mathrm{gt}}(t)} + (1-\alpha) > 1$$

$$\frac{\mathrm{p}_{\mathrm{ml}}(t|D')}{\mathrm{p}_{\mathrm{gt}}(t)} > 1 \tag{6.31}$$

Therefore, we can equivalently use $\frac{\mathrm{p_{ml}}(t|D')}{\mathrm{p_{gt}}(t)} > 1$ or $\log\left(\frac{\mathrm{p_{ml}}(t|D')}{\mathrm{p_{gt}}(t)}\right) > 0$ to perform the term selection.

The only issue that remains is the estimation of appropriate values for the weights $q'(t)$ of the newly included query terms. Since the value of the score can be increased arbitrarily by using increasingly larger values of $q'(t)$, we need to constrain the aggregate value of the weights. One reasonable constraint is that the magnitude of the query weights be unity:

$$||Q'|| = \sqrt{\sum_{t \in Q'} q'(t)^2} = 1 \tag{6.32}$$

Adopting this constraint, we can use the technique of Lagrange multipliers (Bertsekas 1982) to find the set of query term weights, $\{q'(t)\}$, that maximizes the score:

$$\sum_{t \in Q'} q'(t) \log\left(\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)}\right) \tag{6.33}$$

The corresponding Lagrangian function is given by:

$$Ł(Q', \lambda) = \sum_{t \in Q'} q'(t) \log\left(\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)}\right) + \lambda\left(\sqrt{\sum_{t \in Q'} q'(t)^2} - 1\right) \tag{6.34}$$

Taking the partial derivative of (6.34) with respect to $\lambda$ and setting it to zero, we get back the constraint equation:

$$\frac{\partial}{\partial \lambda} Ł(Q', \lambda) = 0 \tag{6.35}$$

$$\sqrt{\sum_{t \in Q'} q'(t)^2} = 1 \tag{6.36}$$

Taking the partial derivative of (6.34) with respect to the query term weight $q'(t)$ and setting it to zero, we get

$$\frac{\partial}{\partial q'(t)} Ł(Q', \lambda) = 0 \tag{6.37}$$

$$\log\left(\frac{\mathrm{p}(t|D')}{\mathrm{p}(t)}\right) + \lambda \frac{q'(t)}{\sqrt{\sum_{t \in Q'} q'(t)^2}} = 0 \tag{6.38}$$

Taking the second derivative, we get

$$\frac{\partial^2}{\partial q'(t)^2}\mathrm{L}(Q', \lambda) = \lambda \left(1 - q'(t)^2\right) \tag{6.39}$$

For the score to be maximized, we need this second derivative to be less than zero. Since $0 < q'(t) < 1$, we must have $\lambda < 0$ in order for (6.39) to be negative.

Combining equations (6.36) and (6.38) and solving for $q'(t)$, we get

$$q'(t) = -\frac{1}{\lambda} \log \left( \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} \right) \tag{6.40}$$

Since we require $\lambda < 0$, we see that the appropriate query weights simply have to be proportional to their score contribution:

$$q'(t) \propto \log \left( \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} \right) \tag{6.41}$$

This weighting scheme makes intuitive sense since we want to emphasize terms that contribute more to the score. If desired, we can determine the exact value of the proportionality factor by substituting (6.40) back into (6.36) and solving for $\lambda$. Doing this, we find that:

$$\lambda = -\sqrt{\sum_{t \in Q'} \left( \log \left( \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} \right) \right)^2} \tag{6.42}$$

Our description of the automatic relevance feedback procedure is now complete. We have a procedure that automatically creates a new query $Q'$ based on the original query $Q$ and a set of top-ranked documents retrieved from a preliminary retrieval pass. The goal of the procedure is to increase the likelihood ratio scores of the top-ranked documents by removing certain terms from the original query and adding new terms from the top-ranked documents with appropriate term weights. Hopefully, improving the scores will lead to improved information retrieval performance.

In comparing our query reformulation process to the standard Rocchio algorithm, we note the following similarities and differences. First, both methods add to the query new terms that occur in the top scoring documents from the initial retrieval pass; the Rocchio

algorithm adds all terms while in our approach, only terms that contribute positively to the final score are added. Second, both approaches deemphasize certain terms in the original query; in our approach, we remove terms that contribute negatively to the final score while the Rocchio algorithm subtracts terms that occur in the poor scoring retrieved documents. Third, both methods modify the weights of the terms in the new query; the Rocchio algorithm weights the added and subtracted terms by their average weight in the documents while in our approach, the terms in the new query are weighted by their likelihood ratio scores.

We note that our automatic feedback procedure can significantly increase the number of terms in the query since many of the terms in the joint document $D'$ will satisfy the selection criteria (6.30). We can limit the number of additional terms by modifying this term selection criteria so only terms with scores greater than some threshold $\phi \geq 1$ will be included:

$$\text{add term } t \text{ if } \quad \frac{\mathrm{p}(t|D')}{\mathrm{p}(t)} > \phi \tag{6.43}$$

The use of the threshold to restrict the number of terms with small score contributions is similar in spirit to robust estimation techniques used in statistics that limit the effect of outliers (Huber 1981). In Section 6.3.4, we examine the ability of the automatic relevance feedback procedure to improve retrieval performance and explore the effects of limiting the number of new query terms by increasing the value of $\phi$ in (6.43).

## 6.3 Information Retrieval Experiments

Our information retrieval model is evaluated on the TREC-6 and TREC-7 ad hoc text retrieval tasks (Harman 1997; Harman 1998). Official evaluation results on the 1999 TREC-8 ad hoc text retrieval task are also reported. The ad hoc task involves searching a static set of documents using new queries and returning an ordered list of documents ranked according to their relevance to the query. The retrieved documents are then evaluated against relevance assessments created for each query.

The data sets for these three text retrieval tasks are described in Section 2.2. We note that this data is different from the data used in previous chapters. Specifically, the

documents are now text instead of speech and the size of the document collection is much larger. Also, word-based indexing units are used in the query and document representations instead of subword-based units. As we previously mentioned, one of the goals in this chapter is to benchmark our probabilistic retrieval model on standard text retrieval tasks and to perform a comparative evaluation of our model to other retrieval approaches. We also note that the large size of the document collection makes the retrieval task in this chapter more difficult than the one we have been using for spoken document retrieval. This is reflected in the lower mean average precision (mAP) retrieval score.

In the following sections, we briefly mention the text preprocessing that was done, and then present several retrieval experiments. In these experiments, we explore the usefulness of the $p(Q)$ normalization in the scoring, the effect of using different mixture weights in the probability model, the use of the automatic relevance feedback processing, and section-based weighting of the query terms.

### 6.3.1 Text Preprocessing

Before a document is indexed, it undergoes a relatively standard set of text preprocessing steps. First, the text is normalized to remove non-alphanumeric characters like punctuation and to collapse case. Next, sequences of individual characters are automatically grouped to create single terms in an "automatic acronym aggregation" stage. For example, the text string "`U. S. A.`" would be converted to "`u s a`" after normalization and then to "`usa`" after acronym aggregation. Stop words, derived from a list of 600 words, are then removed from the document. In addition to standard English function words, certain words frequently used in past TREC topics such as "document," "relevant," and "irrelevant" are also included in the list. Finally, the remaining words are conflated to collapse word variants using an implementation of Porter's stemming algorithm (Porter 1980). To maintain consistency, each topic description also undergoes the exact same text preprocessing steps before it is indexed and used to retrieve documents from the collection.

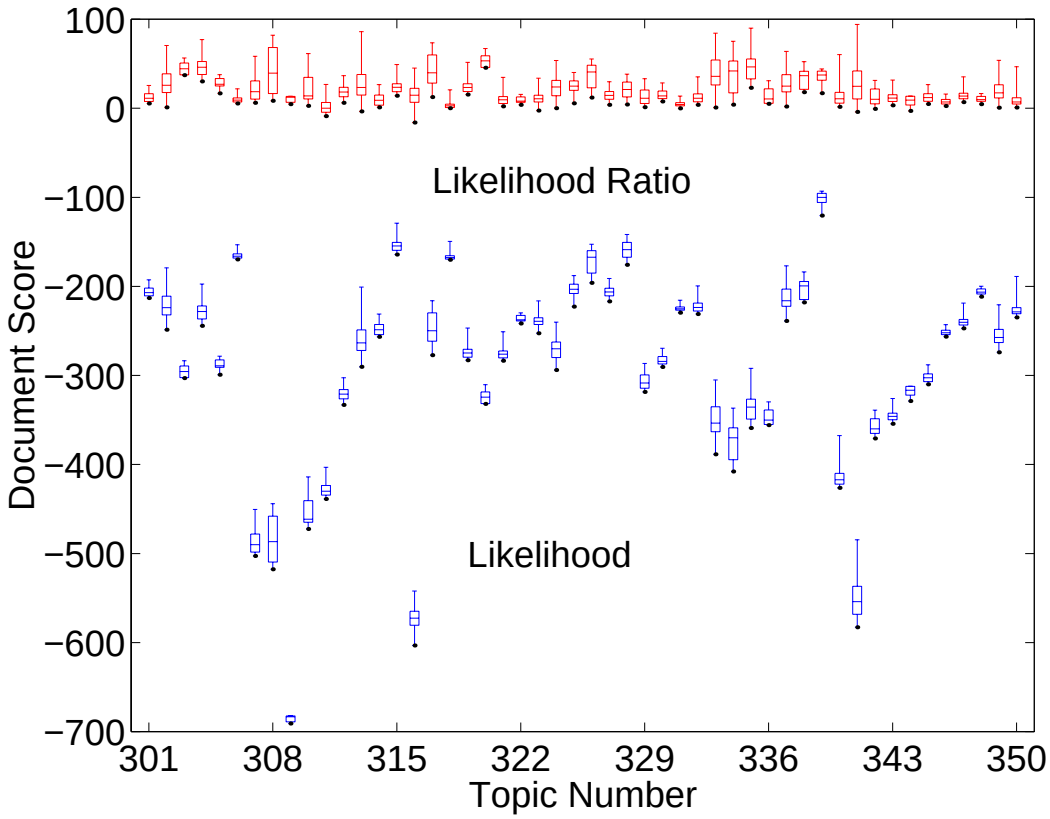


Figure 6-1: Distribution of scores for the relevant documents for topics 301-350 in the TREC-6 task. The likelihood scores have a very wide distribution across queries while the likelihood ratio scores are more tightly clustered.

### 6.3.2 p($Q$) Normalization

As discussed in Section 6.2.1, the p($Q$) normalization factor in the scoring function (6.2) does not affect the ranking of the documents because it is constant for all documents $D_i$ given a specific topic $Q$. However, we choose to keep this factor because it helps to provide a meaningful interpretation of the scores as a relative change in the likelihood and allows the document scores to be more comparable across different topics. In addition, as we've seen in Section 6.2.2, the p($Q$) normalization factor plays an important role in the term selection and weighting stages of the automatic relevance feedback procedure (Equation 6.30).

To illustrate the difference between the (unnormalized) likelihood score (p($Q|D_i$)) and the (normalized) likelihood ratio score ($\frac{\text{p}(Q|D_i)}{\text{p}(Q)}$), Figure 6-1 plots the distribution of these

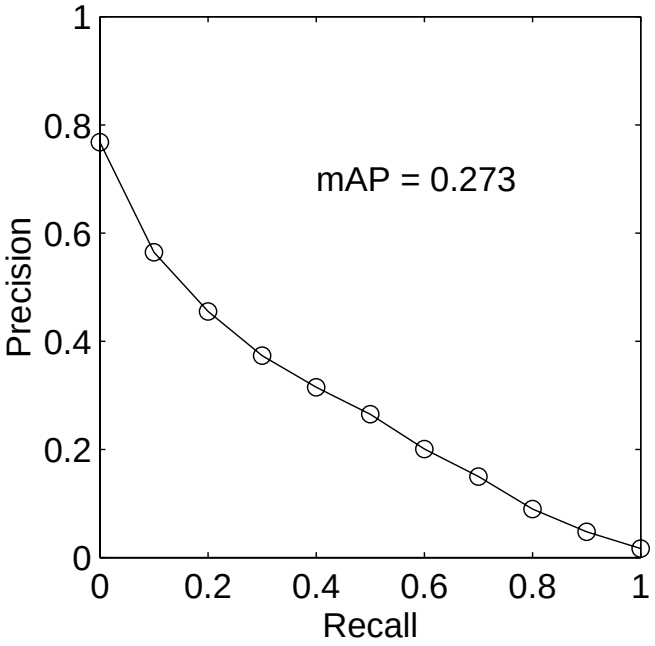


Figure 6-2: Precision-Recall curve and mean average precision (mAP) score on the TREC-6 ad hoc task using a mixture weight of $\alpha = 0.5$. Both likelihood and likelihood ratio scoring will give identical performance results since the document scores are not compared across the different topics.

two scores for the subset of relevant documents for the 50 topics (topics 301-350) in the TREC-6 task. The likelihood scores have a very wide distribution (relative to variance) across queries while the likelihood ratio scores are more tightly clustered. Box plots are used to indicate the score distributions. The center line in the box indicates the mean value while the lower and upper edges of the box indicate, respectively, the lower and upper quartiles. The vertical lines extending below and above the box show the entire range of the scores. From Figure 6-1, we observe that the document likelihood scores can differ drastically depending on the topic. Across topics, the scores for relevant documents may not even overlap. The best score for some topics (e.g., 309 and 316) can be worse than the lowest scores for other topics (e.g., 315 and 339). Scoring the documents using the likelihood ratio, however, puts the scores for the different topics on a much more comparable range; there is much more overlap across different topics. In addition, these scores can be interpreted as how much more likely the document has become after the topic is specified than before.

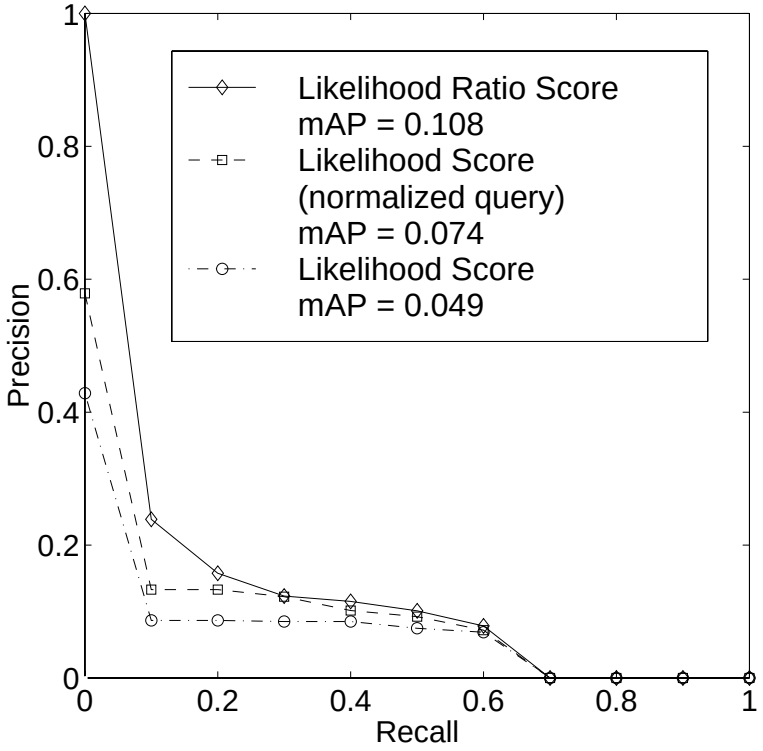


Figure 6-3: Precision-Recall curves resulting from using a single threshold across all topics on the TREC-6 data. This evaluation technique measures the ability of the different scoring methods to handle across topic comparisons.

In the computation of the standard information retrieval measures of recall, precision, and mean average precision (mAP), each topic is treated independently. Precision-recall curves are generated for each topic separately using individual thresholds. These separate curves are then combined to create an aggregate precision-recall curve and the single number mAP measure. Since document scores are not compared across the different topics in the computation of these standard information retrieval measures, they will be identical for both the likelihood and likelihood ratio scores. In Figure 6-2, we plot the resulting aggregate precision-recall curve and mean average precision (mAP) measure on the TREC-6 ad hoc task for the 50 topics (301-350). This is the baseline performance of our retrieval model using the preliminary retrieval run and a fixed topic-independent mixture weight of $\alpha = 0.5$. A performance of mAP=0.273 is achieved.

There are certain related applications, however, such as document clustering and topic

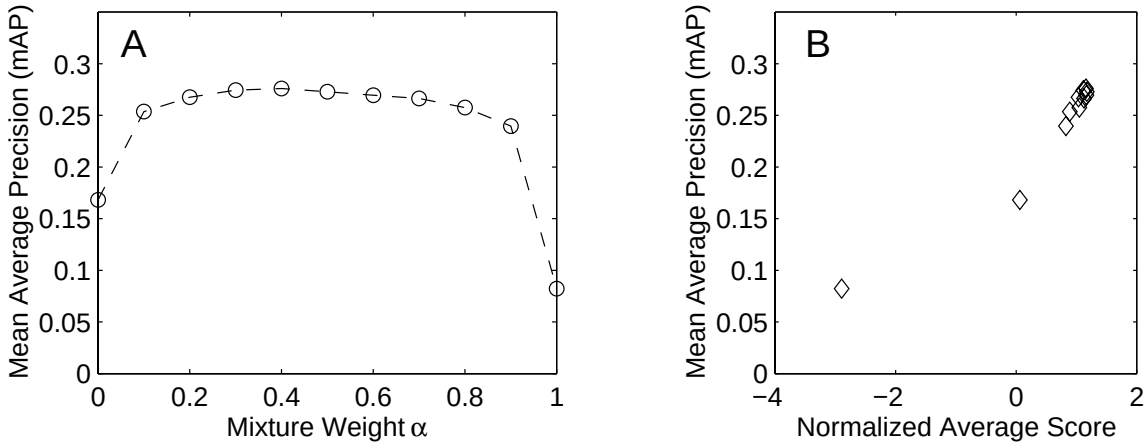


Figure 6-4: (A) Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task as a function of the value of the mixture weight $\alpha$. (B) Scatter plot of mAP versus the normalized average score of the top documents for each of the different $\alpha$ weights.

detection, where it is important to be able to compare document scores across different "topics." To quantify how much the likelihood ratio score can help in these situations, we can generate a precision-recall curve that results from using a *single threshold* across all the different topics. In this way, we can measure the ability of the different scoring methods to handle across topic score comparisons. In Figure 6-3, we show such recall-precision curves and the associated mAP measure for the 50 topics on the TREC-6 ad hoc data using three different scoring methods. As expected, the raw likelihood score performs poorly when cross topic score are compared. A normalized likelihood score (normalized by the number of the terms in the topic) gives slightly better results. However, the likelihood ratio score, which is not only normalized by the number of terms in the topic but also by the prior likelihoods of the terms, gives even better performance.

### 6.3.3 Mixture Weights

In this section, we explore the effect of different $\alpha$ mixture weight estimates on retrieval performance and examine topic-specific and topic-independent $\alpha$'s.

To quantify the sensitivity of the model to the mixture weight $\alpha$, we explore a range of possible weight values and measure the resulting retrieval performance. In Figure 6-4A,

| Mixture Weight Estimate | mAP |
|---|---|
| Fixed ($\alpha = 0.5$) | 0.273 |
| Topic-Independent ($\alpha = 0.434$) | 0.275 |
| Topic-Dependent (variable $\alpha$) | 0.278 |

Table 6-1: Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task using different estimates of the mixture weight $\alpha$: a fixed value of $\alpha$ for all topics, an automatically estimated topic-independent value of $\alpha$, and automatically estimated topic-dependent values of $\alpha$.

we plot retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task as a function of the value of the mixture weight $\alpha$. We see that although retrieval performance does vary with the value of $\alpha$, there is a relatively large range of stable and good performance.

A scatter plot of mAP versus the normalized average score of the top retrieved documents for each of the different $\alpha$ weights is shown in Figure 6-4B. The plot shows that retrieval performance is well correlated ($\rho = 0.96$) with the document scores. This means that we can use the document scores to find an appropriate value of $\alpha$ that can be expected to give reasonably good retrieval performance. In fact, the automatic $\alpha$ parameter estimation procedure that we described in Section 6.2.1 tries to maximize the likelihood of topic $Q$ given document $D_i$, $\mathrm{p}(Q|D_i)$, which is the numerator of the document score (6.2). Since the denominator of the score, $\mathrm{p}(Q)$, remains unchanged, this is equivalent to maximizing the entire document score. As shown in Table 6-1, running the preliminary retrieval pass using a fixed weight of $\alpha = 0.5$ results in a retrieval performance of mAP=0.273. Performance improves slightly to mAP=0.275 when we use the automatically estimated topic-independent weight of $\alpha = 0.434$.

Since topic statements can be very different from one another, we can expect that using the same $\alpha$ weight for every topic is probably suboptimal. This is indeed the case as illustrated in Figure 6-5, which plots retrieval performance in average precision (AP) for three different topics (327, 342, and 350) from the TREC-6 ad hoc task as a function of the value of the mixture weight $\alpha$. We see that the optimal value of $\alpha$ for each topic can be very different. To address this issue, we can estimate topic-dependent $\alpha$'s, as discussed in

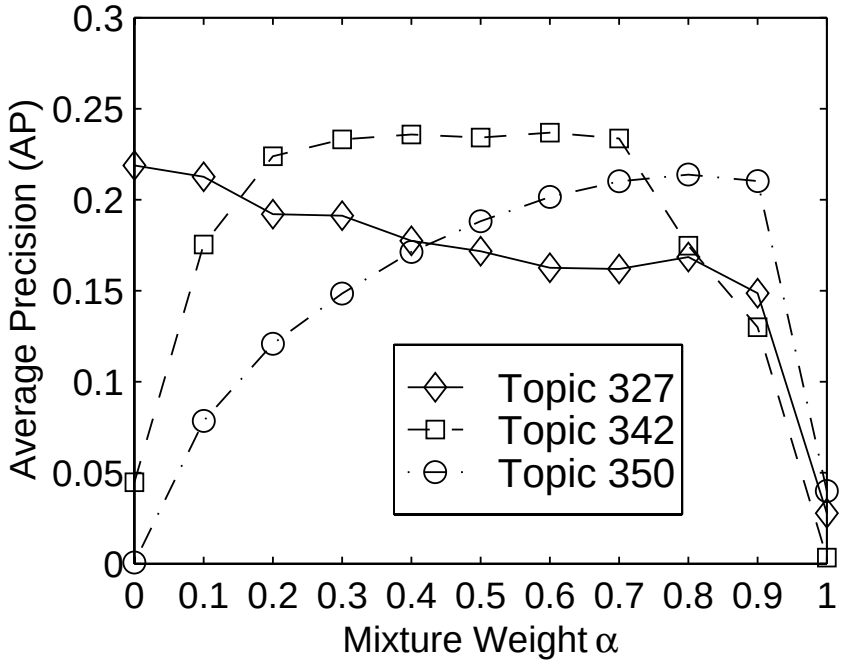


Figure 6-5: Retrieval performance in average precision (AP) for topics 327, 342, and 350 from the TREC-6 ad hoc task as a function of the value of the mixture weight $\alpha$. Each topic has a different optimal value of $\alpha$.

Section 6.2.1. In Figure 6-6, we plot the distribution of the automatically estimated topic-dependent $\alpha$ mixture weights for the 50 topics (301-350) in the TREC-6 task. Many of the weights are centered around the topic-independent estimated value of $\alpha = 0.434$ but there are several topics that have weights at the extreme ends of the range. Using these topic-dependent $\alpha$ mixture weights, retrieval performance is further improved to mAP=0.278 as shown in the last row of Table 6-1.

### 6.3.4 Automatic Feedback

In this section, we evaluate the automatic relevance feedback procedure described in Section 6.2.2 and examine its ability to improve retrieval performance.

Recall that during the feedback process, a new topic $Q'$ is created by removing certain terms from the original topic $Q$ and adding new terms (with appropriate term weights) from the top scoring documents obtained from a preliminary retrieval run. The number of

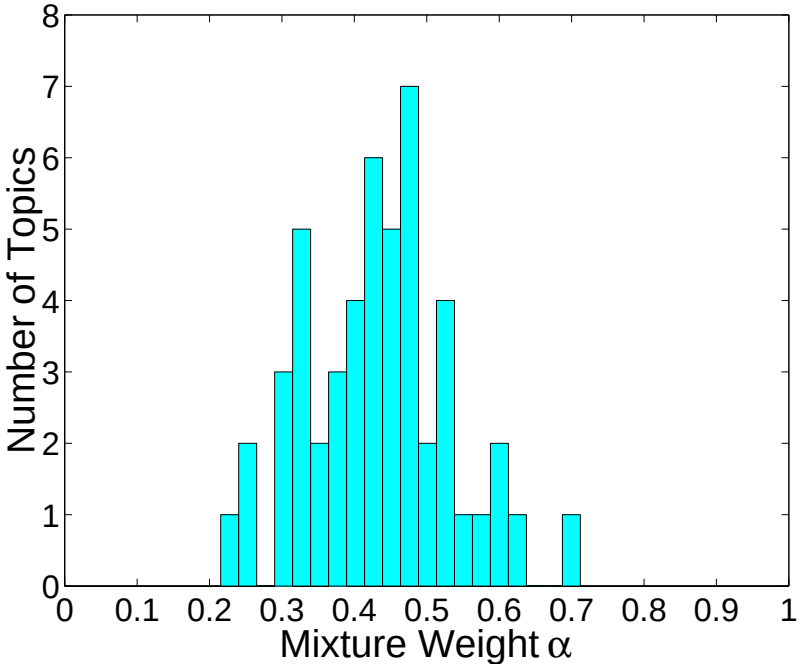


Figure 6-6: Distribution of the automatically estimated topic-dependent $\alpha$ mixture weights for topics 301-350 in the TREC-6 task. The pooled $\alpha$ estimate is 0.434 while the average $\alpha$ value is 0.432.

new terms added to $Q'$ can be controlled by changing the threshold $\phi$ in the term selection criteria (6.43). Lowering the value of $\phi$ adds more terms. Note that new query terms are added in order of decreasing contribution to the total score; terms that contribute most to improving the score are added first.

Retrieval performance, measured in mean average precision (mAP), on the TREC-6 ad hoc task as the number of terms in the new topic $Q'$ is varied is plotted in Figure 6-7. The same information is presented in tabular form in Table 6-2. Running the preliminary retrieval pass using the original topics, which average 27 unique terms each, gives a performance measure of mAP=0.273. Using automatic feedback to modify the topic results in significant performance improvements as illustrated in Figure 6-7 and Table 6-2. As more terms are included in the new topic $Q'$, performance improves sharply, reaches a maximum at around 250-300 terms, declines slightly, and then levels off. The retrieval performance peaks at mAP=0.317 for approximately 243 terms.

It is interesting to note that performance is relatively stable over a wide range of topic

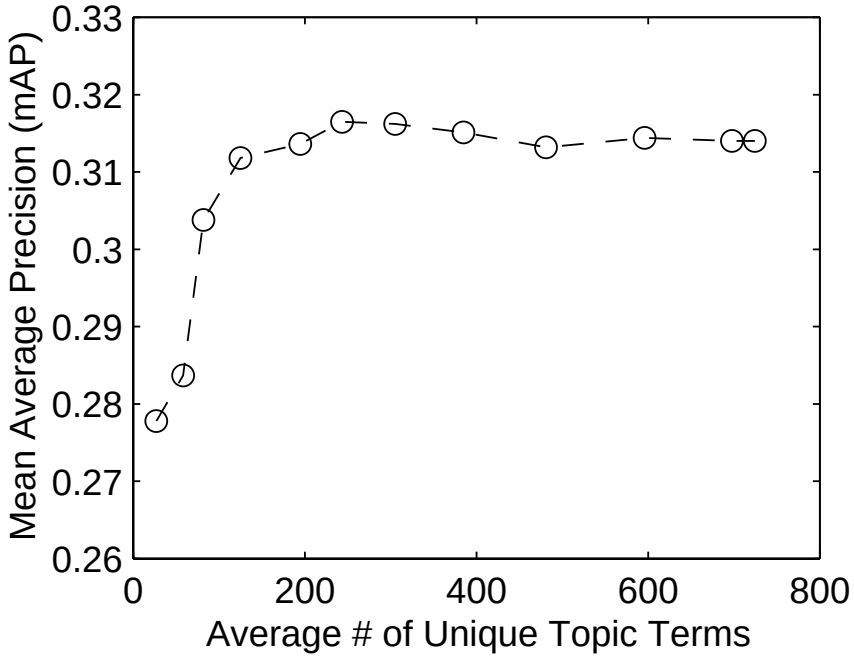


Figure 6-7: Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task using the automatic feedback procedure as the number of terms in the new topic $Q'$ is varied. By lowering the threshold $\phi$ in the term selection criteria (6.43), more terms are included in the new topic.

sizes spanning 200 to 700 terms. By significantly increasing the number of terms in the topic, one may expect that the topic specification may become too broad and, as a result, the retrieval performance will be adversely affected. However, this does not happen in our case because the terms added to the new topic $Q'$ are weighted proportionally to their score contribution as specified in (6.41). As a result, many of the additional terms will only have a small effect on the total score.

In terms of determining an appropriate $\phi$ threshold to use, one possibility is to simply set $\phi = 1.0$ so all terms that contribute positively to the score will be included. This corresponds to adding the maximum number of terms allowed by our procedure. Using this threshold value on the TREC-6 ad hoc task, the average number of unique terms in the new query $Q'$ grows to 724.2. However, from the behavior shown in Figure 6-7, the same or even slightly better performance can be achieved by using many fewer terms. We find empirically that a reasonable threshold to use is $\phi = 0.25 \times \mathrm{S}_{\max}(D_i, Q)$, where $\mathrm{S}_{\max}(D_i, Q)$

| Retrieval Pass | Avg # Unique Topic Terms | mAP |
|---|---|---|
| **Preliminary** | 27.0 | 0.278 |
| **Automatic Feedback** | 57.9 | 0.284 |
| | 81.8 | 0.304 |
| | 124.8 | 0.312 |
| | 194.6 | 0.314 |
| | 243.3 | **0.317** |
| | 305.2 | 0.316 |
| | 384.6 | 0.315 |
| | 481.0 | 0.313 |
| | 595.6 | 0.314 |
| | 697.5 | 0.314 |
| | 724.2 | 0.314 |

Table 6-2: Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task using the automatic relevance feedback procedure as the number of terms in the new topic $Q'$ is varied. This table corresponds to the plot in Figure 6-7.

is the score of the top retrieved document $D_i$ for topic $Q$. This relative threshold value puts us in the stable performance region without adding too many terms to the new topic $Q'$.

We conclude that incorporating the automatic feedback processing stage into the retrieval system significantly improves retrieval performance. Large gains of 0.035 to 0.04 in absolute mean average precision (from mAP=0.278 to 0.317) are obtained.

### 6.3.5 Topic Section Weighting

As described in Section 2.2, the queries or topics statements for the retrieval tasks consist of three different sections: a title, a description, and a narrative. We can expect that the different sections contain different amounts of useful information. To quantify how useful each section is in finding the relevant documents for the topic, we can evaluate the retrieval performance resulting from using each topic section individually. In Table 6-3, we show retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task using the different topic sections. We examine the use of the title, description, and narrative sections individually, the title and description sections combined (T+D), and all three sections together (T+D+N). Retrieval performance after the preliminary and

| Topic Section | Avg # Unique Topic Terms | mAP | |
|---|---|---|---|
| | | Preliminary | Feedback |
| Title (T) | 2.5 | 0.225 | 0.230 |
| Description (D) | 8.8 | 0.178 | 0.221 |
| Narrative (N) | 21.7 | 0.218 | 0.253 |
| T+D | 9.5 | 0.247 | 0.296 |
| T+D+N (All) | 27.0 | 0.278 | 0.317 |

Table 6-3: Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task using different sections of the topics: title, description, and narrative individually, title and description combined (T+D), and all three sections together (T+D+N). The second column shows the average number of unique terms in each section. The third and fourth columns show performance after the preliminary and feedback retrieval stages, respectively.

feedback retrieval stages are shown along with the average number of unique terms in each topic section. We can make several observations. First, the different topic sections vary greatly in their size. The title, description, and narrative sections average 2.5, 8.8, and 21.7 unique terms, respectively. Second, even though the title section contains the fewest terms, its preliminary retrieval performance is better than that of the other two sections. This implies that the terms from the title section are more useful than those from the other sections. Third, using multiple topic sections results in better performance. Combining the title and description (T+D) gives performance that is better than any of the individual sections, and using all three (T+D+N) gives even better performance. Fourth, automatic feedback improves performance in all cases but is more effective when there are more terms in the topic statement. In particular, the gain for the title section is small compared to the gains for the other sections.

In the above experiments, when we combined the different topic sections, we weighted each section equally. This means that in the T+D+N case which combines all three sections, the title section only contributes, on average, 2.5 terms to the combined topic while the narrative section contributes 21.7 terms. From the performance of the individual topic sections in Table 6-3, it is clear that the terms in the title section are more useful than those in the narrative section. Maybe emphasizing terms from some sections (e.g., the title), more than terms from other sections (e.g., the narrative) in the formation of the combined

| | mAP | |
|---|---|---|
| **Topic Section** | **Preliminary** | **Feedback** |
| T+D | 0.247 | 0.296 |
| T+D (weighted) | 0.260 | 0.297 |
| T+D+N | 0.278 | 0.317 |
| T+D+N (weighted) | 0.303 | 0.325 |

Table 6-4: Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task with and without topic section weighting. Performance is shown for two different topic configurations: title and description combined (T+D), and all three sections (title, description, and narrative) together (T+D+N). Performance after the preliminary and feedback retrieval stages are shown.

topic will result in better performance than just equally weighting all the sections. This is indeed the case. In (Miller et al. 1998), they found that weighting the topic terms based on what section they are in improved retrieval performance. In (Robertson et al. 1998), the output from several retrieval runs using the individual topic sections are combined to give improved performance.

We can adopt a similar approach of weighting terms based on their topic section membership to try to further improve retrieval performance. One method is to weight the terms from each topic section in proportion to the average score of the top documents retrieved using that section. The idea is that topic sections that give higher document scores should be emphasized more than those that give lower scores. We are basically using the document score as a predictor of retrieval performance which is consistent with our retrieval model which ranks documents based on descending values of the document scores. Because the scores are normalized (likelihood ratios), we are able to compare them across different topic statements (consisting of different topic sections) to determine which topic formulation is better. Basically, we run three retrieval passes using the title, description, and narrative sections individually, compute the average score of the top retrieved documents from each run, and then use those scores in weighting the terms from the different topic sections. The process used to select the set of top scoring documents is the same as the one used in the automatic feedback procedure (6.25). For each new task, this procedure is used to automatically determine the appropriate section weights. Using this topic section weighting

scheme on the TREC-6 ad hoc task, we get section weights of 4.2 for the title, 1.8 for the description, and 1.0 for the narrative. This weighting emphasizes the title section the most, then the description section, and finally the narrative section.

Weighting the topic sections in this way results in a small but consistent performance improvement over weighting each section equally, as shown in Table 6-4. Retrieval performance in mean average precision (mAP) on the TREC-6 ad hoc task with and without topic section weighting is shown for two different topic configurations: title and description combined (T+D), and all three sections (title, description, and narrative) together (T+D+N). The effect of the topic section weighting is greater on the preliminary retrieval pass than on the automatic feedback pass. Recall that the feedback process already includes term selection and term weighting. As a result, some of the gains from the section weighting may already be accounted for in the feedback processing.

## 6.4 Information Retrieval Performance

All of the above experiments were conducted on the TREC-6 ad hoc text retrieval task. These development experiments were used to configure the system and to tune some system parameters. Specifically, the final retrieval system has the following configuration:

- Dynamic (for each query) and automatic estimation of the mixture parameter $\alpha$ using the procedure described in Section 6.2.1 with the following parameter: $M$=5.
- Use of the second pass automatic relevance feedback procedure described in Section 6.2.2 with the following parameters: $\gamma$=0.75 (Equation 6.25) and $\phi = 0.25 \times \mathrm{S}_{\max}(D_i, Q)$ (Equation 6.43), where $\mathrm{S}_{\max}(D_i, Q)$ is the score of the top retrieved document $D_i$ for topic $Q$.
- Use of the query section weighting procedure described in Section 6.3.5 with the following parameter: $\gamma$=0.75 (Equation 6.25). The section weights are automatically determined for each new set of test queries.

Now that the system configuration is set, we need to evaluate the performance of the final retrieval system on a new set of held-out test data. We use the TREC-7 and TREC-8 ad hoc retrieval tasks, described in Section 2.2 for this purpose.

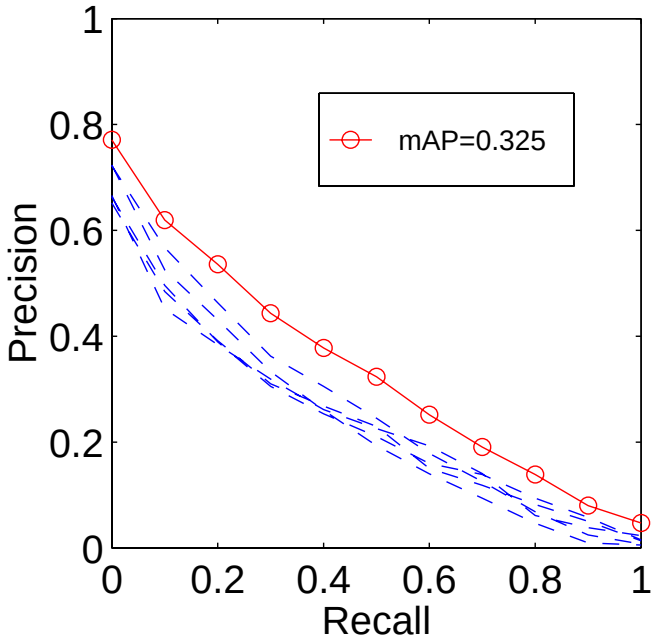


Figure 6-8: Precision-Recall curves for the TREC-6 ad hoc task. Performance for the top 5 systems (out of 57) that participated in the official TREC-6 ad hoc task are shown. Also plotted is the precision-recall curve corresponding to our retrieval model.

### 6.4.1 Retrieval Performance on the Development Set

We first review the performance of the retrieval system on the TREC-6 ad hoc text retrieval task. Figure 6-8 plots the precision-recall curves for the top 5 systems (out of 57) that participated in the official TREC-6 ad hoc retrieval task (Harman 1997). Also plotted is the precision-recall curve corresponding to our retrieval model. Our system achieves a mAP=0.325 which is significantly better than the other systems (the top official system had a mAP=0.260). This performance comparison is, of course, unfair since we used this data as our development set to tune system parameters. As a result, the performance will be unrealistically high. In the next two sections, we objectively evaluate the performance of our retrieval system using new held-out test data: the TREC-7 and TREC-8 ad hoc retrieval tasks.

| | mAP | |
|---|---|---|
| **Topic Section** | **Preliminary** | **Feedback** |
| T+D | 0.212 | 0.243 |
| T+D+N | 0.250 | 0.284 |

Table 6-5: Retrieval performance in mean average precision (mAP) on the TREC-7 ad hoc task using different topic specifications: title and description combined (T+D), and all three sections together (T+D+N). Performance for the preliminary and automatic feedback retrieval stages are shown.

### 6.4.2 Retrieval Performance on the Test Set

In Table 6-5, we show the performance (in mAP) of our system on the TREC-7 ad hoc task. Retrieval is done using two types of topics: one consisting of the title and description sections only (T+D) and the other consisting of all three (title, description, and narrative) sections (T+D+N). Performance is shown for the preliminary retrieval pass and the automatic feedback pass. We observe that automatic feedback significantly improves performance for all conditions and that using longer topic statements is better. Figure 6-9 plots the precision-recall curves for the top 5 systems (out of 36) that participated in the official TREC-7 ad hoc retrieval task using the full topic description (T+D+N) (Harman 1998). Also plotted is the precision-recall curve corresponding to our retrieval model. Our system achieves a mAP=0.284 which ranks third on the list behind the two top systems which both had a mAP=0.296. The fourth ranked system had a mAP=0.282. On this task, we see that our system is very competitive with current state-of-the-art retrieval systems.

### 6.4.3 Retrieval Performance on the Evaluation Set

We participated in the 1999 TREC-8 ad hoc text retrieval evaluation (Harman 1999). Performance on the official TREC-8 ad hoc task using our probabilistic retrieval model is shown in Figure 6-10. Two retrieval runs were submitted: one consisting of the title and description sections only (T+D) and the other consisting of all three (title, description, and narrative) sections (T+D+N). A performance of mAP=0.298 is achieved using the shorter topics while the full topics gave a mAP=0.323. Out of the 55 participating systems that used the short topic description, our system ranked **sixth** behind systems that had mAPs

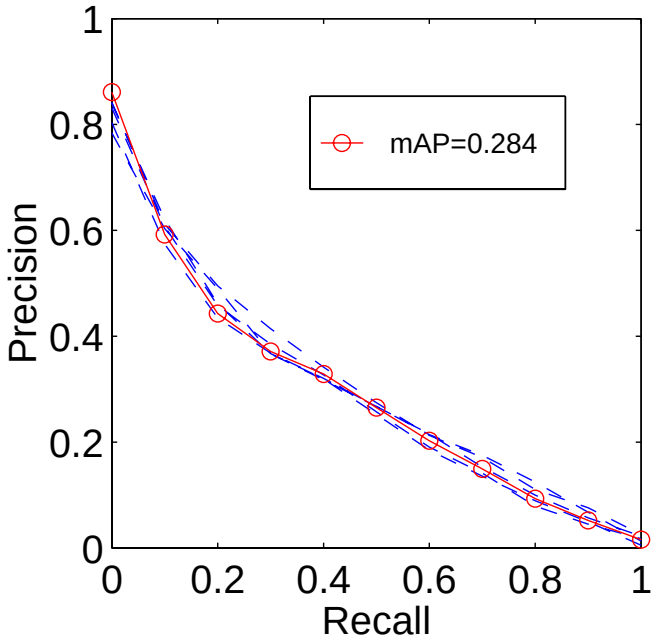


Figure 6-9: Precision-Recall curves for the TREC-7 ad hoc task. Performance for the top 5 systems (out of 36) that participated in the official TREC-7 ad hoc task using the full topic description (T+D+N) are shown. Also plotted is the precision-recall curve corresponding to our retrieval model.

of 0.321, 0.317, 0.317, 0.306, and 0.301. Out of the 37 participating systems that used the entire topic description, our system ranked **fourth** behind systems that had mAPs of 0.330, 0.324, and 0.324. Difference in mAP from the median performance for each of the 50 topics for the full topic run (T+D+N) are shown in Figure 6-11. Of the 50 topics, 40 scored at or above the median level and seven achieved the maximum score. On this task, we again see that our retrieval model is very competitive with current state-of-the-art retrieval systems.

## 6.5 Summary

In this chapter, we presented a novel probabilistic information retrieval model and demonstrated its capability to achieve state-of-the-art performance on large standardized text collections. The retrieval model scores documents based on the relative change in the document likelihoods, expressed as the ratio of the conditional probability of the document given the query and the prior probability of the document before the query is specified.

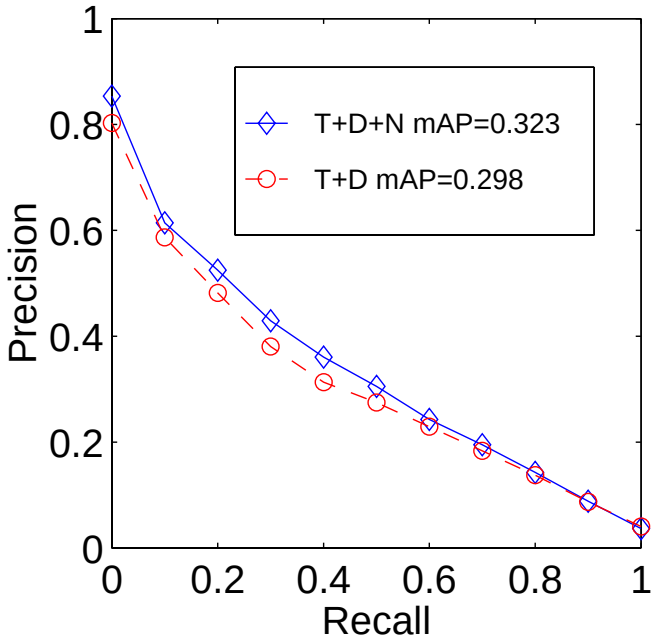


Figure 6-10: Precision-Recall curves for the TREC-8 ad hoc task. Performance using topics consisting of title and description (T+D), and full topics consisting of the title, description, and narrative sections (T+D+N) are shown.

Statistical language modeling techniques are used to compute the document likelihoods and the model parameters are estimated automatically and dynamically for each query to optimize well-specified maximum likelihood objective functions. An automatic relevance feedback strategy that is specific to the probabilistic model was also developed. The procedure automatically creates a new query (based on the original query and a set of top-ranked documents from a preliminary retrieval pass) by selecting and weighting query terms so as to maximize the likelihood ratio scores of the set of documents presumed to be relevant to the query. To benchmark the performance of the new retrieval model, we used the standard ad hoc text retrieval tasks from the TREC-6 and TREC-7 text retrieval conferences. Official evaluation results on the 1999 TREC-8 ad hoc text retrieval task were also reported. Experimental results indicated that the model is able to achieve performance that is competitive with current state-of-the-art retrieval approaches. In the next chapter, Chapter 7, this retrieval model is used to implement a more integrated approach to spoken document retrieval.

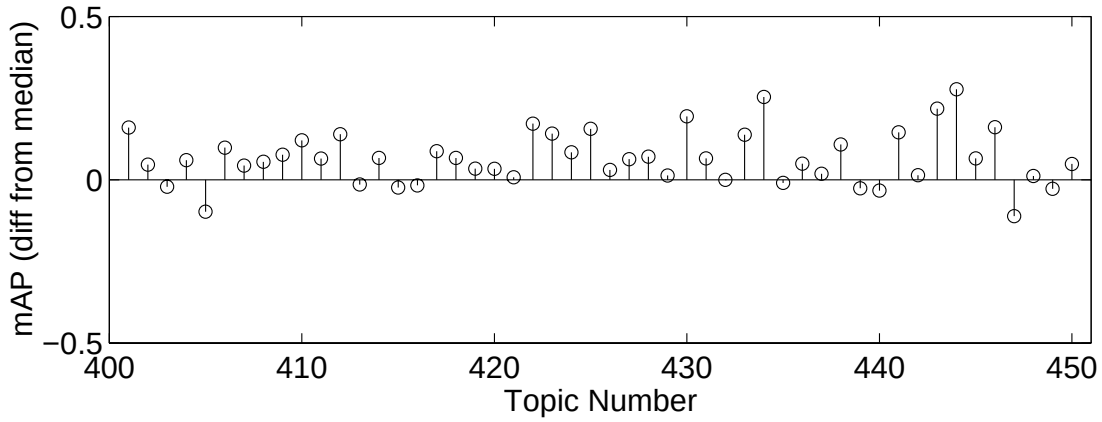


Figure 6-11: Difference (in mean average precision) from the median for each of the 50 topics in the TREC-8 ad hoc task. Full topics consisting of the title, description, and narrative sections are used.

# Chapter 7

# Integrated Speech Recognition and Information Retrieval

In Chapter 5, we explored a number of methods that take into account the characteristics of the speech recognition errors and try to compensate for them. These methods try to incorporate the error information into the indexing and retrieval process while maintaining the existing framework of the traditional vector space retrieval model. This is accomplished by making slight modifications to various components of the vector space model. In particular, near-miss terms generated using the recognition error confusion matrix and terms determined by automatic relevance feedback are added to the system by appropriately expanding the query vector. Similarly, information about likely alternative ($N$-best) speech recognition hypotheses are incorporated into the system by adding additional terms to the document vector. Also, approximate term matching capability is added to the model by modifying the scoring function used to compute the similarity between the document and query vectors. As indicated in Equation 5.3, the change is relatively straightforward and only involves the addition of a new factor that measures the similarity between query and document terms when they are not identical; the rest of the score remains the same.

In addition to operating within the framework of the traditional retrieval model, these methods also try to maintain the architecture of having a cascade of independently operating speech recognition and information retrieval components. This cascade approach has the

advantage of being modular in that different speech recognizers and retrieval models can be easily combined. However, there are some shortcomings. First, there is an input-output mismatch between the two components. The speech recognizer outputs errorful recognition hypotheses while the retrieval model expects error-free text representations of the documents as input. In Section 4.3, we saw that ignoring this mismatch results in a significant drop in retrieval performance. We investigated, in Chapter 5, several ways to try to compensate for the errors and were able to achieve some performance improvements.

Second, the two components have decoupled objectives. The two systems were originally designed to solve different problems and therefore have different objectives and make different assumptions. There is no guarantee that the goals of the two components will be consistent with each other or with the overall goal of the combined system. One issue is that speech recognizers are usually designed to output the most likely symbol sequence (i.e., string of words or phones depending on the vocabulary) corresponding to a given set of acoustic observations. High scoring alternative recognition hypotheses are typically not accounted for. The availability of additional hypotheses may not be important for pure speech recognition purposes. However, it could be useful for information retrieval since it offers the potential of including terms that would otherwise be missed. Another issue is that recognizers are usually trained to try to minimize the error rate of the most likely symbol sequence. Although retrieval performance is correlated with recognition performance (as we saw in Section 4.4), it is not clear that minimizing the error rate is the best thing to do for retrieval purposes. One reason is related to the point mentioned above: error rate is only computed using the single best recognition hypothesis; likely alternatives are not considered. Another reason is that all symbols are treated equally in computing the recognition error rate. This means that in a word based system, for example, function words are just as important as content words. In information retrieval, all words are not created equal. In fact, the removal of commonly occurring function words ("stop words") has generally been shown to improve retrieval performance (Salton and McGill 1983).

Text-based retrieval systems are usually designed to index a collection of text documents and to perform term matching to find relevant documents in response to user-specified queries. Because the retrieval model is originally developed for use on text document col-

lections where the words are assumed to be known with certainty, there is no explicit mechanism for dealing with errors in the document representations. With spoken documents, the speech recognizer output will likely contain errors and the need for error tolerant retrieval methods will be more important. Also, conventional text retrieval models generally do not make use of additional information that can be generated from the recognizer such as likelihood and confidence scores. These scores can be used to weight our belief in the accuracy of different recognition hypotheses which is important when we are dealing with errorful transcriptions. This type of information should be useful for spoken document retrieval.

In this chapter, we propose a different approach to spoken document retrieval where the speech recognition and information retrieval components are more tightly integrated. This new approach represents a step towards moving away from the conventional method of simply cascading the two components: using the speech recognizer to transform the speech into text transcriptions and then feeding those directly into a full-text retrieval system. We do this by developing new recognizer and retrieval models where the interface between the two components is better matched and the goals of the two components are consistent with the overall goal of the combined system.

First, we need a retrieval model that makes direct use of information that can be computed by the speech recognizer. For this, we use the novel probabilistic information retrieval model described in Chapter 6. Recall that the model scores documents based on the relative change in the document likelihoods, expressed as the likelihood ratio of the conditional probability of the document given the query and the prior probability of the document before the query is specified. The idea is that documents that become more likely after the query is specified are probably more useful to the user and should score better and be ranked ahead of those documents whose likelihoods either stay the same or decrease. The document likelihoods are computed using statistical language modeling techniques which eventually make use of the probabilistic quantity $\mathrm{p}(t|D_i)$, the probability that term $t$ occurs in spoken document $D_i$. It is this quantity that will serve as the interface between the two components.

Second, we need to have a speech recognizer that can estimate and output these $\mathrm{p}(t|D_i)$ probabilities given the speech waveform for spoken document $D_i$. To do this, we modify the

objective of the speech recognizer to compute these term occurrence probabilities instead of finding the most likely symbol sequence given the acoustic observations. In this way, the interfaces of the speech recognition and retrieval components are now better matched: the recognizer outputs term occurrence probabilities which the retrieval model expects as input. In addition, the goals of the two components are now consistent with the overall goal of the combined system. The goal of the total system is, of course, to automatically index and retrieve spoken documents. This is consistent with the goal of the retrieval component. The retrieval model makes use of probabilistic quantities that need to be estimated from the spoken documents. This is what the modified speech recognizer now does. Thus the goal of the recognizer is now consistent with the retrieval component and with the goal of the overall system.

In the following sections, we describe the integrated SDR approach in detail and present some experimental results. First, we describe some related work on the use of additional information from the speech recognizer such as likelihoods and confidence measures for spoken document retrieval and topic classification of speech messages. Next, we describe several ways to compute the desired term probabilities including modifying the speech recognizer to enable it to output the occurrence probabilities directly. Finally, we evaluate the performance of the integrated approach and compare it to the performance of the robust methods developed in Chapter 5. We find that the integrated approach performs better than the robust methods and is able to improve retrieval performance by over 28% from the baseline.

In the experiments presented in this chapter, we again use phonetic subword units which are overlapping, fixed-length, phone sequences ranging from $n$=2 to $n$=6 in length with a phone inventory of 41 classes. These phonetic $n$-gram subword units, as described in Section 3.2.1, are derived by successively concatenating the appropriate number of phones from the phonetic transcriptions.

## 7.1 Related Work

In (Siegler et al. 1998; Siegler et al. 1997), the use of word confidence scores and word occurrence probabilities for spoken document retrieval are investigated. In (Siegler et al. 1997), confidence annotations are used to estimate the correctness of each word in the most likely word sequence hypothesized by a large vocabulary speech recognizer. The confidence scores are computed using a decision tree classifier with the following features: acoustic score from the recognizer, trigram language model score from the recognizer, word duration, and $N$-best homogeneity (proportion of the $N$ hypotheses in which the word appears). The confidence scores are incorporated into the retrieval model (a standard vector space IR model using TF×IDF weights) by computing expected term frequency (ETF) and expected inverse document frequency (EIDF) in place of the normal TF and IDF values. The ETF for a word is computed by summing the probability of correctness (confidence score) over all occurrences of the word. The EIDF for a word is computed by summing over all documents the probability that the word occurs in each document and then dividing by the total number of documents. Retrieval experiments on the TREC-6 SDR task (Garofolo et al. 1997) show that if the confidence annotations are accurate, retrieval performance can be significantly improved. However, using actual confidence estimates resulted in only small performance gains.

In (Siegler et al. 1998), occurrence probability estimates are computed for each word in the most likely word sequence hypothesized by the large vocabulary speech recognizer. These word probabilities are estimated from the word lattices created during the recognition search. Specifically, the word probability is a function of the number of competing word hypotheses in the lattice that have overlapping times (i.e., the lattice occupation density). The larger the number of alternative word hypotheses, the less certain we are about the occurrence of that word. A standard vector space IR model using TF×IDF weights is used for retrieval but the weighting is reinterpreted in a probabilistic light to allow the word probability estimates to be directly incorporated into the retrieval model. In particular, they generalize the TF weights to allow non-integral word counts (expressed as word probabilities) and show that the IDF weight can be viewed as variant of mutual information (which makes use of word probabilities). Retrieval experiments on the TREC-7 SDR

task (Garofolo et al. 1998) show that using word probabilities can help improve retrieval performance. It should be noted that in both of these approaches only the words in the most likely (top one) recognition hypothesis are annotated with either confidence scores or occurrence probabilities; additional word hypotheses are not proposed or scored.

In (McDonough et al. 1994), word occurrence probabilities computed by an HMM-based word spotting system are used to perform topic classification of speech messages. The word spotting system processes each speech message and outputs keyword hypotheses with an associated posterior probability of occurrence score. This $\mathrm{p}(w,t)$ score estimates the probability that keyword $w$ ended at time $t$, and is computed using the Baum-Welch algorithm during the recognition search (Rohlicek et al. 1989). An expected number of occurrences for each keyword is obtained by summing up the posterior probability score associated with each putative keyword occurrence in the message. A feature vector consisting of the expected number of occurrences for each keyword is formed and then used as input to a topic classifier which is based on a multinomial model of the keyword occurrences. This feature vector is compared to a recognizer-based feature vector where each component contains the number of times the corresponding keyword appears in the most likely (top one) recognition hypothesis. Topic classification experiments on the Switchboard corpus (Godfrey et al. 1992) show that using word occurrence probabilities can improve performance: classification is better for the wordspotting-based feature vector than the recognizer-based one.

## 7.2 Computing Term Occurrence Probabilities

In this section, we describe several methods for estimating $\mathrm{p}(t|D_i)$, the probability that term $t$ occurs in spoken document $D_i$. This includes using the top one recognition hypothesis, using the $N$-best recognition hypotheses, using an expanded term set approach, and modifying the recognizer to compute the term occurrence probabilities directly. These term occurrence probabilities are used directly by the probabilistic retrieval model.

The simplest approach is to just use the top one recognition hypothesis. In this case, the phonetic recognizer outputs the most likely phone sequence for each document. The appropriate phonetic subword unit indexing terms are generated from the phonetic tran-

scription. And the term counts in each document are used to estimate the term occurrence probabilities:

$$\mathrm{p}_1(t|D_i) = \frac{c_i(t)}{\sum_\tau c_i(\tau)} \tag{7.1}$$

where $c_i(t)$ is the number of times term $t$ occurs in document $D_i$.

A potentially better estimate of the term occurrence probabilities may be obtained by including additional recognition hypotheses. This can be done by using the $N$-best recognition hypotheses, instead of just the top one hypothesis. In this case, the phonetic recognizer outputs the top $N$=100 phone sequences for each document. For each of the $N$ hypothesized phonetic transcriptions, the appropriate phonetic subword unit indexing terms are generated. The term counts in this "expanded" document are then used to estimate the term occurrence probabilities:

$$\mathrm{p}_2(t|D_i) = \frac{\sum_{n=1}^{N} c_i^n(t)}{\sum_{n=1}^{N} \sum_\tau c_i^n(\tau)} \tag{7.2}$$

where $c_i^n(t)$ is the number of times term $t$ occurs in the $n^{\text{th}}$ transcription for document $D_i$. This probability estimate reflects the belief that if a term appears in many of the top $N$ hypotheses, it is more likely to have actually occurred than if it appears in only a few. We note that in (7.1) and (7.2), a probability of zero is assigned to terms that are not observed.

Another way to estimate the term occurrence probability is to incorporate near-miss or approximate match terms. This can be done by first expanding the term $t$ to a larger set of possible realizations of the term $\{t^*\}$ and then summing, over all members of this expanded set, the probability that $t$ can be realized as $t^*$. The occurrence probability of term $t$ in document $D_i$, $\mathrm{p}(t|D_i)$, can therefore be computed according to:

$$\mathrm{p}_3(t|D_i) = \sum_{t^*} \mathrm{p}(t, t^* \,|\, D_i) = \sum_{t^*} \mathrm{p}(t \,|\, t^*, D_i)\, \mathrm{p}(t^*|D_i) \tag{7.3}$$

where $\mathrm{p}(t \,|\, t^*, D_i)$ is an appropriate measure of the probability that $t$ can be realized as $t^*$ in document $D_i$, and the summation is over all possible realizations, $t^*$, of term $t$. The occurrence probability of term $t^*$, $\mathrm{p}(t^*|D_i)$, can be estimated using either $\mathrm{p}_1(t|D_i)$ (7.1) or $\mathrm{p}_2(t|D_i)$ (7.2) described above. In addition to errors in the document transcriptions, this

approach also enables us to deal with the use of synonyms (in a word based system, for example) in a principled way by summing over an appropriately expanded set of possible equivalents, $t^*$, for each original term, $t$. We note that this "expanded term set" approach is very similar to the approximate matching procedure described in Section 5.3. In fact, we use the same method for estimating $\mathrm{p}(t \mid t^*, D_i)$ as we did for estimating $p(i \mid j)$ (Equations 5.5 and 5.6) in Section 5.3. Recall that $p(i \mid j)$ is the conditional probability that the term is really $i$ given that we observe term $j$ in the noisy document, and it is estimated using information about the error characteristics of the speech recognizer (i.e., the recognition error confusion matrix). As in Section 5.3, thresholds can be placed on $\mathrm{p}(t \mid t^*, D_i)$ to limit the size of the expanded term set. For the experiments in this chapter, we use a relatively small threshold value of 1e-5 to allow a large term set.

Finally, we can modify the speech recognizer so that it can output the $\mathrm{p}_4(t|D_i)$ probabilities directly. This can be accomplished by changing the objective of the recognizer to compute occurrence probabilities of the indexing terms given the acoustic observations of the spoken documents instead of finding the most likely phone sequence. We first review what is done in the standard phonetic recognizer. Let $A$ be a sequence of acoustic observations, $W$ be a sequence of phonetic units, and $S$ be a sequence of speech segments. The conventional phonetic recognizer finds the most likely phone sequence by searching for the $W^* = \{w_1, w_2, \ldots, w_N\}$ that has the highest probability $\mathrm{p}(W|A)$:

$$W^* = \arg\max_W \mathrm{p}(W|A) = \arg\max_W \sum_S \mathrm{p}(W, S|A) \tag{7.4}$$

$$= \arg\max_W \sum_S \frac{\mathrm{p}(A|W,S)\ \mathrm{p}(S|W)\ \mathrm{p}(W)}{\mathrm{p}(A)} \tag{7.5}$$

The summation is over all possible segmentations $S$ of the speech utterance for a particular phonetic sequence $W$ and gives the total probability of that phone sequence. However, in the recognition search the sum is approximated with a maximization to simplify the computation. The total probability of the phone sequence is therefore approximated by the probability of the single sequence with the best segmentation:

$$W^* = \arg\max_{W,S} \frac{\mathrm{p}(A|W,S)\ \mathrm{p}(S|W)\ \mathrm{p}(W)}{\mathrm{p}(A)} \tag{7.6}$$

The prior probability of the acoustics $\mathrm{p}(A)$ is constant for a given utterance and can be safely ignored since it doesn't affect the maximization. $\mathrm{p}(A|W,S)$ is the acoustic score corresponding to the specified phone sequence and segmentation and is computed by the acoustic models. $\mathrm{p}(S|W)$ is the duration score and is modeled by a simple segment transition weight that balances segment insertions and deletions. Finally, $\mathrm{p}(W)$ is the language model score and is computed by a statistical $n$-gram (typically bigram) language model.

The Viterbi search algorithm (Viterbi 1967; Forney 1973) is typically used to solve the maximization problem described above. The search essentially finds the best path through a lattice that connects lexical nodes (i.e., phone labels) across time. An example lattice for a segment-based search is shown in Figure 7-1. The x-axis represents time and is marked with possible segment boundaries. The y-axis represents a set of lexical nodes. A vertex in the lattice represents a boundary between two phones. One possible path through the lattice is illustrated by the solid line connecting a set of vertices across time. To find the optimal path, the Viterbi search considers segment boundaries $b$ in a time-synchronous manner. For each node $p$ at the current boundary, the active lexical nodes from the previous boundaries are retrieved. A path from each of these active nodes is extended to the current node if the path is allowed by the pronunciation network. For each extended path, the appropriate acoustic, duration, and language model scores are computed and added to the path score $\delta_b(p)$. Only the best arriving path to each node is kept (this is the maximization step). If the scores from the arriving paths are summed, we have a Baum-Welch search (Rabiner 1989) instead of a Viterbi search. The Viterbi maximization is illustrated at node $(b_4, p_3)$ in Figure 7-1. There are five paths entering the node. The path with the best score, denoted with the solid line, comes from node $(b_2, p_1)$. As a result, only a pointer back to node $(b_2, p_1)$ is kept at node $(b_4, p_3)$. When all the boundaries have been processed, the resulting lattice contains the best path, along with its associated score, from the initial node at the beginning of the lattice to every node in the lattice. To recover the overall best path, we look for the node with the best score at the ending boundary (in this example, $(b_5, p_2)$) and then perform a backtrace following the back-pointers stored at each node. The sequence of phones along this best path (in this example, $p_4\ p_1\ p_3\ p_2$) is the most likely phonetic sequence hypothesized by the recognizer. To reduce computation and memory requirements,

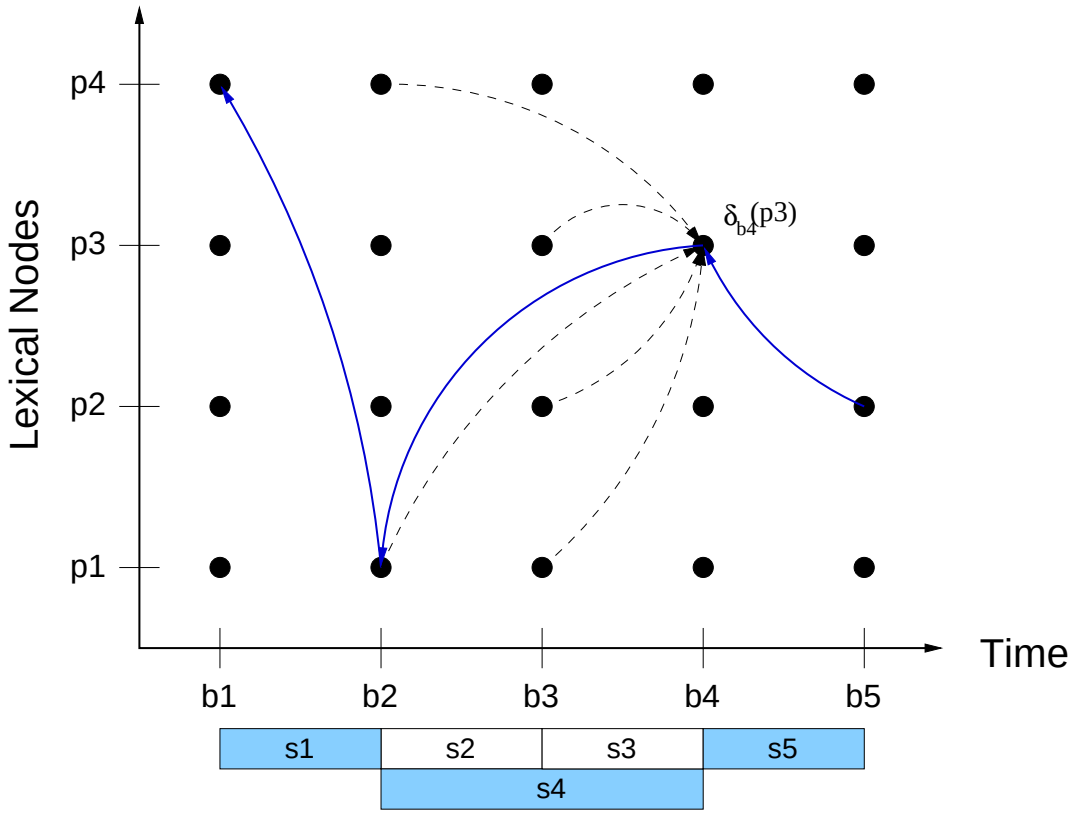


Figure 7-1: An example segment-based Viterbi search lattice. The x-axis represents time and is marked with possible segment boundaries. The y-axis represents a set of lexical nodes (phone labels). A vertex in the lattice represents a boundary between two phones. At each node, only the best arriving path is kept. The search finds the most likely phone sequence by finding the optimal path through the lattice.

beam pruning is usually done after each boundary has been processed. Paths that don't score within a fixed threshold of the maximum scoring path at the current boundary become inactive and can no longer be extended to future boundaries.

Instead of the most likely phonetic sequence, we want the recognizer to output estimates of the probability that indexing term $t$ occurred in the given speech message $D_i$, i.e., $\mathrm{p}(t|D_i)$. Ideally, we would like to compute this quantity by considering all possible phonetic sequences $W$ of the document $D_i$, finding all occurrences of the term $t$ in each sequence $W$, determining the probability of each of these occurrences of $t$, summing these probabilities to get the expected number of times term $t$ occurred in the document, and then normalizing it by the total number of term occurrences in the document to obtain $\mathrm{p}(t|D_i)$.

However, because we cannot consider the exponential number of possible sequences $W$, we need to make some approximations. First, we limit the number of possible phone sequences by considering only those retained in the phone lattice created by the Viterbi recognition search. An example phone lattice is shown in Figure 7-2. The lattice is a connected acyclical graph where each node corresponds to a phone hypothesis and has associated with it a phone label $p$ for the segment $s$, the start time $b_s(s)$ of the segment, the end time $b_e(s)$ of the segment, a score $\delta_{b_e}(p)$ representing the likelihood of the best path from the beginning of the utterance to the current phone (node), and links (arcs) to possible following phones (nodes). The $\delta$ score is computed by the Viterbi search algorithm described above. The x-axis represents time and is marked with possible segment boundary times $(b_1, b_2, b_3, \ldots)$. Boundaries are locations in time where the phonetic segments $s_j$ are allowed to start and end. A second approximation is that instead of considering the term occurrences on all phone sequences represented in the lattice (which is still a very large number), we only consider term occurrences on the *locally most likely* phone sequences. For each possible segment boundary, we consider all possible phones that can terminate at that boundary. And for each phone, we find the term along the most likely phone sequence that terminates at that phone. A third approximation is in the computation of the probability of the term occurrence. We consider the likelihood score of the phone sequence corresponding to the term occurrence and then normalize it estimate the term occurrence probability.

To generate the phonetic subword unit indexing terms and to estimate the associated occurrence probabilities $\mathrm{p}(t|D_i)$ we use the following procedure:

1. Create the set of possible indexing terms:

**let** $n$ be the length of the phonetic subword units of interest
**for each** boundary $b_k$ in the speech message
**for each** segment $s_j$ that terminates at boundary $b_k$
**for each** possible phonetic label $p_i$ for segment $s_j$
**let** $w[0, \ldots, n]$ be the array of phone labels of length $n+1$
corresponding to the phone sequence resulting from
a partial Viterbi backtrace starting from phonetic segment $p_i$
**let** $t = w[1, \ldots, n]$ be the label of the subword unit indexing term
**let** $b_e(t)$ be the ending boundary of term $t$

**let** $b_s(t)$ be the start boundary of term $t$

**let** $S(t) = \exp(\log \delta_{b_e(t)}(w[n]) - \log \delta_{b_s(t)}(w[0]))$ be the score for term $t$.

**store** the tuple $\{t, b_e(t), b_s(t), S(t)\}$

**end**

**end**

**end**

2. Estimate the occurrence probability p($t$) of each term $t$ in the set of tuples by appropriately normalizing its score:

**for each** ending boundary $b_k$ in the set of tuples

**for each** term label $t$ that has ending boundary $b_k$

**let** $c(t)$ += $S(t)/\sum_\tau S(\tau)$

**end**

**end**

**for each** term label $t$ in the set of tuples

**let** $p(t) = c(t)/\sum_\tau c(\tau)$

**end**

We can illustrate the above procedure using the phone lattice in Figure 7-2. The first step is to generate the set of possible indexing terms. This is done by running local backtraces from all possible phonetic segments from all possible boundaries. For example, a backtrace of length $n$=3 starting at boundary $b_5$ and segment $s_5$ with phone label $p_9$ results in the following phone sequence: $[p_2, p_3, p_5, p_9]$ (shaded in the figure) and the following tuple: $\{p_3\_p_5\_p_9, b_2, b_5, \exp(\log \delta_{b_5}(p_9) - \log \delta_{b_2}(p_2))\}$. The $\delta_{b_5}(p_9)$ score corresponds to the likelihood of the best path from the beginning of the utterance to segment $s_5$ ending at boundary $b_5$ with phone label $p_9$. This best path is equivalent to the best path from the beginning of the utterance to segment $s_2$ ending at boundary $b_2$ with phone label $p_2$ plus the following path: $(s_3, p_3) \rightarrow (s_4, p_5) \rightarrow (s5, p_9)$. To determine a score corresponding just to the indexing term of interest, i.e., the phone sequence $p_3$ $p_5$ $p_9$, we can take the difference in the log scores between $\delta_{b_5}(p_9)$ and $\delta_{b_2}(p_2)$ and then exponentiate the result. In the second step, the term scores are appropriately normalized to estimate the term occurrence probability: p($t|D_i$). First, the scores are normalized over ending boundaries so that the scores of all terms that end at a specified boundary $b_k$ sum to one. Next, the scores are nor-

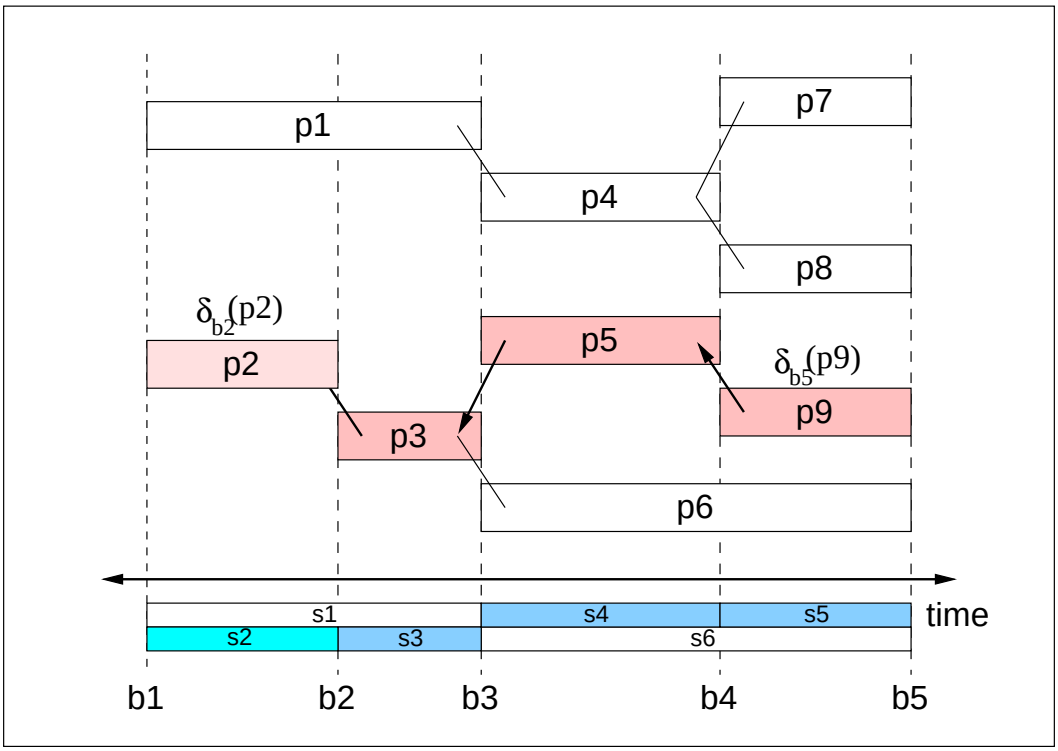


Figure 7-2: An example segment-based phone lattice. The x-axis represents time and is marked with potential segment boundary locations. Each node corresponds to a phone hypothesis and has associated with it a phone label $p$ for the segment $s$, the start time $b_s(s)$ of the segment, the end time $b_e(s)$ of the segment, a score $\delta_{b_e}(p)$ representing the likelihood of the best path from the beginning of the utterance to the current phone (node), and links (arcs) to possible following phones (nodes).

malized over all the terms that occur in the document to come up with the final occurrence probability $\mathrm{p}_4(t|D_i)$ for term $t$.

## 7.3 Spoken Document Retrieval Experiments

In this section, we use the NPR corpus to evaluate the performance of the new probabilistic retrieval model and the performance of the integrated spoken document retrieval approach.

First, we perform retrieval using the new probabilistic retrieval model and compare its performance to that of the initial vector space retrieval model (Section 2.4) that we have been using up until now. We then measure the retrieval performance of the integrated approach as we evaluate the different methods discussed above for estimating the term

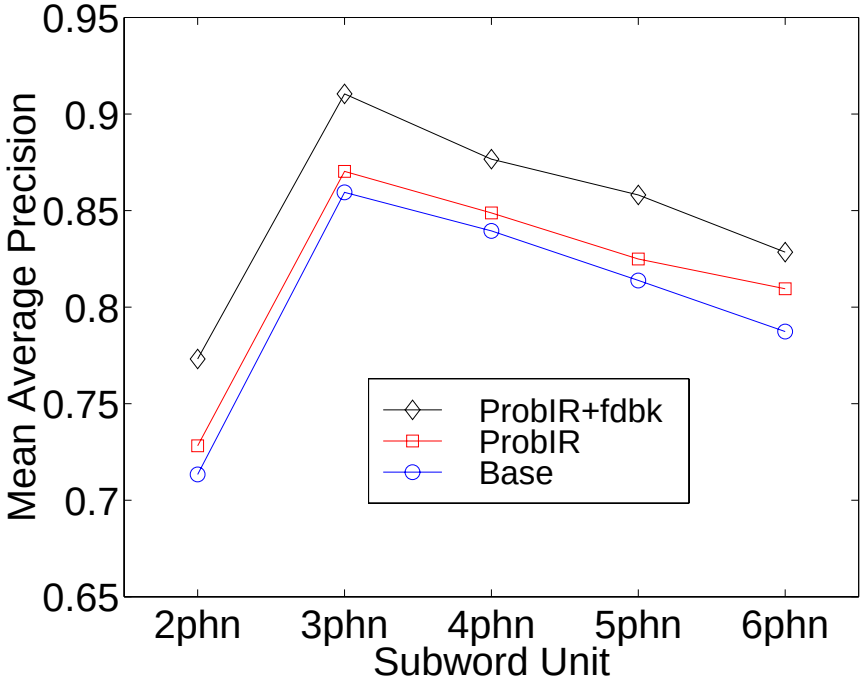


Figure 7-3: Retrieval performance (in mAP) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from clean phonetic transcriptions. Performance for three different retrieval systems is shown: the baseline vector space retrieval model (base), the new probabilistic retrieval model (ProbIR), and the probabilistic retrieval model with the second stage automatic feedback (ProbIR+fdbk).

occurrence probabilities $p(t|D_i)$. We will also compare the performance of the integrated approach to that of the robust methods described in Chapter 5.

Figure 7-3 shows retrieval performance (in mean average precision) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from error-free phonetic transcriptions. Performance for three different retrieval systems is shown. The first system is the baseline vector space retrieval model (base), described in Section 2.4, that we have been using up until now. The second system is the new probabilistic retrieval model (ProbIR) that we described in Chapter 6. The third system is the probabilistic retrieval model again, but this time including the second stage automatic relevance feedback procedure (ProbIR+fdbk). Comparing the baseline retrieval model with the new probabilistic retrieval model, we see that retrieval performance for subword units of all lengths is slightly better with the probabilistic model. Including the second pass automatic relevance feedback procedure results

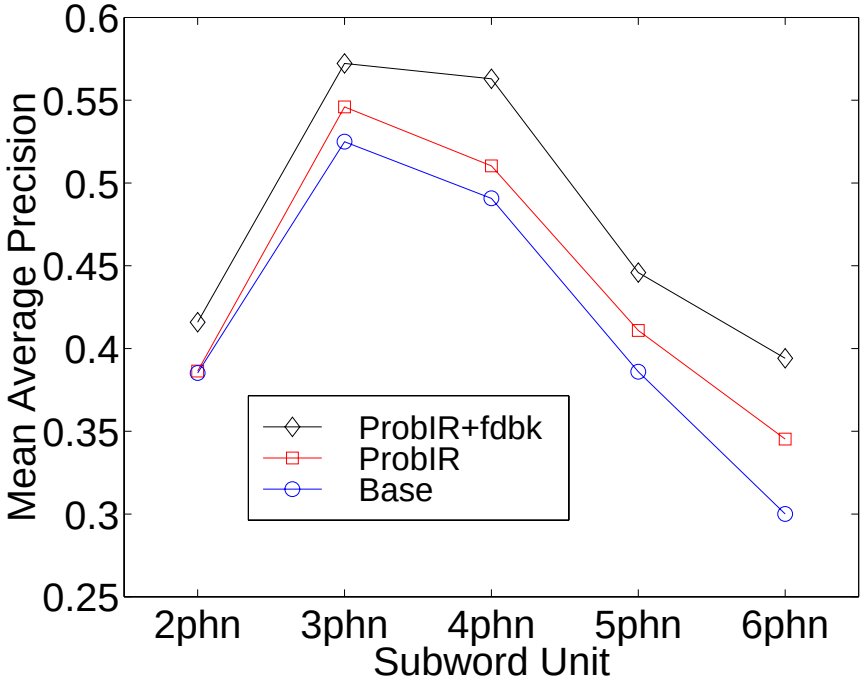


Figure 7-4: Retrieval performance (in mAP) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from errorful phonetic recognizer output. Performance for three different retrieval systems is shown: the baseline vector space retrieval model (base), the new probabilistic retrieval model (ProbIR), and the probabilistic retrieval model with the second stage automatic feedback (ProbIR+fdbk).

in a significant and consistent performance improvement. For the length $n$=3 phonetic subword unit, retrieval performance improves from mAP=0.860 to mAP=0.910. We note that using the automatic relevance feedback procedure designed for the vector space model (i.e., the Rocchio method) described in Section 5.5 also improves performance over the baseline system. However, performance is not as good as using the the probabilistic model with automatic feedback. For example, with the length $n$=3 phonetic subword unit, retrieval performance improves to mAP=0.893 using the vector space model with Rocchio automatic relevance feedback.

Figure 7-4 shows retrieval performance (in mAP) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from noisy phonetic transcriptions generated by a phonetic recognizer. Only the most likely (top one) recognition hypothesis is used. Again, performance for three different retrieval systems is shown: the baseline vector space retrieval

| | Subword Unit | | | | |
|---|---|---|---|---|---|
| **Condition** | **2phn** | **3phn** | **4phn** | **5phn** | **6phn** |
| **base** | 0.385 | 0.525 | 0.491 | 0.386 | 0.300 |
| **probIR** | 0.386 | 0.546 | 0.510 | 0.411 | 0.345 |
| **probIR+fdbk** | 0.416 | 0.572 | 0.563 | 0.444 | 0.394 |
| **top1** | 0.416 | 0.572 | 0.563 | 0.444 | 0.394 |
| **nbest** | 0.434 | 0.583 | 0.580 | 0.446 | 0.403 |
| **expand** | 0.408 | 0.623 | 0.647 | 0.633 | 0.578 |
| **termprob** | 0.394 | 0.629 | 0.673 | 0.643 | 0.590 |
| **text (base)** | 0.713 | 0.860 | 0.839 | 0.814 | 0.787 |
| **text (probIR)** | 0.728 | 0.870 | 0.849 | 0.825 | 0.810 |
| **text (probIR+fdbk)** | 0.773 | 0.910 | 0.877 | 0.858 | 0.828 |

Table 7-1: Retrieval performance (in mAP) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from errorful phonetic recognizer output. Performance of the baseline vector space retrieval model (base) and the probabilistic retrieval model with (probIR+fdbk) and without (probIR) automatic feedback is shown. Performance is also shown for several different methods for estimating $\mathrm{p}(t|D_i)$: using the top one recognition hypothesis to estimate $\mathrm{p}_1(t|D_i)$ (top 1), using the $N$=100 $N$-best recognition hypotheses to estimate $\mathrm{p}_2(t|D_i)$ (nbest), using the expanded term set approach to compute $\mathrm{p}_3(t|D_i)$ (expand), and using term occurrence probabilities, $\mathrm{p}_4(t|D_i)$, computed directly by the recognizer (termprob). Reference performance using subword units generated from clean phonetic transcriptions (text) is also shown for the baseline retrieval model (base) and the probabilistic retrieval model with (probIR+fdbk) and without (probIR) automatic feedback.

model (base), the new probabilistic retrieval model (ProbIR), and the probabilistic retrieval model with automatic feedback (ProbIR+fdbk). As in the case of the clean phonetic transcriptions, retrieval performance using the probabilistic model is slightly better than using the baseline retrieval model. The addition of automatic relevance feedback again results in a significant and consistent performance improvement. For the length $n$=3 phonetic subword unit, retrieval performance improves from mAP=0.525 with the baseline retrieval model to mAP=0.546 with the probabilistic retrieval model and finally to mAP=0.572 with the addition of automatic feedback. In the remaining experiments in this chapter, the probabilistic retrieval model with the second stage automatic relevance feedback procedure will be used.

We now evaluate the performance of the integrated spoken document retrieval approach. We examine the four different methods for estimating the term occurrence probabilities, $\mathrm{p}(t|D_i)$, described in Section 7.2. Figure 7-5 plots the retrieval performance (in mAP) for

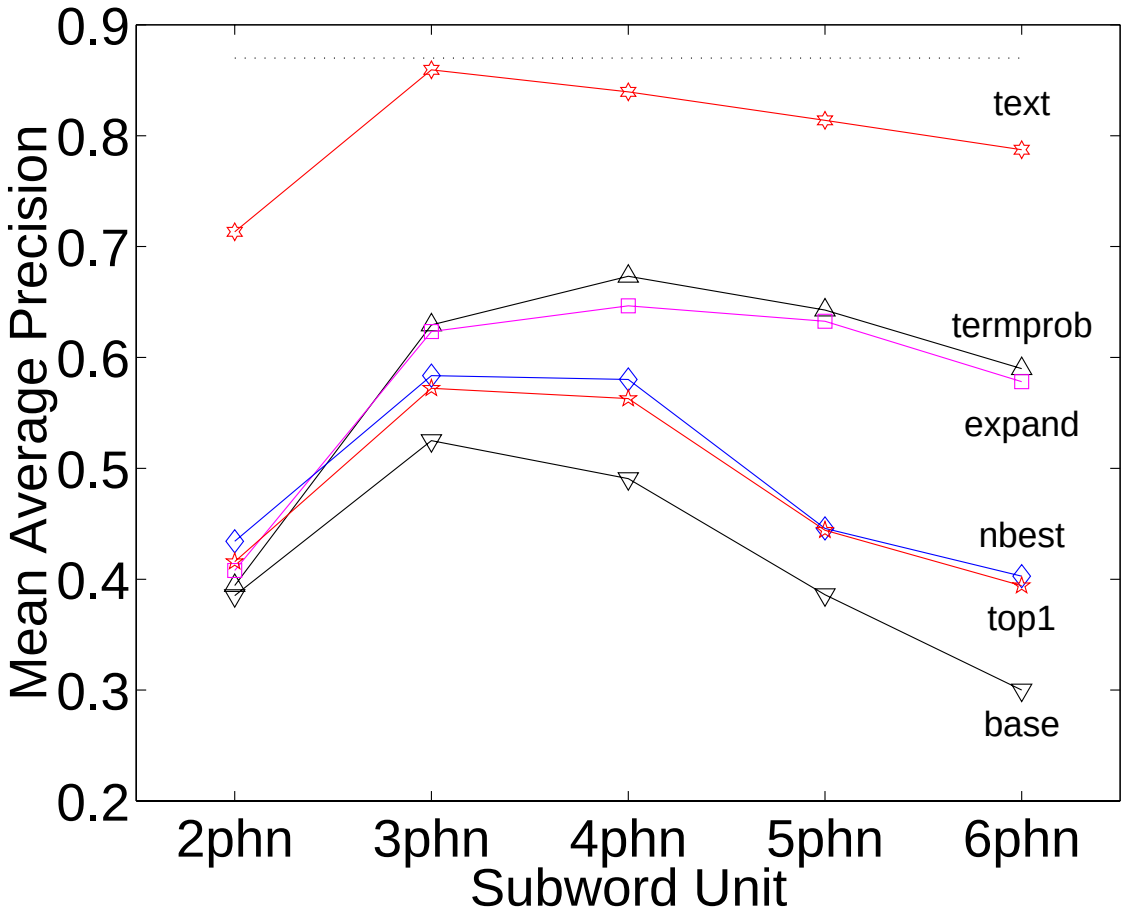


Figure 7-5: Retrieval performance (in mAP) for different length ($n = 2, \ldots, 6$) phonetic subword units generated from errorful phonetic recognizer output. First, performance of the baseline vector space retrieval model (base) is shown. Next, performance using the probabilistic retrieval model with automatic feedback is shown for several different methods for estimating the term occurrence probabilities, $\mathrm{p}(t|D_i)$: using the top one recognition hypothesis to estimate $\mathrm{p}_1(t|D_i)$ (top 1), using the $N$=100 $N$-best recognition hypotheses to estimate $\mathrm{p}_2(t|D_i)$ (nbest), using the expanded term set approach to compute $\mathrm{p}_3(t|D_i)$ (expand), and using term occurrence probabilities, $\mathrm{p}_4(t|D_i)$, computed directly by the recognizer (termprob). The reference performance uses the baseline retrieval model with subword units generated from clean phonetic transcriptions (text). The dotted line shows the reference performance (mAP=0.87) using word units derived from error-free text transcriptions of the spoken documents.

different length ($n = 2, \ldots, 6$) phonetic subword units and Table 7-1 lists the corresponding performance numbers.

First, the performance of the baseline vector space retrieval model (base) is plotted. This is the same baseline that was used in Chapter 5. Next, the performance of the probabilistic retrieval model with automatic feedback using the top one recognition hypothesis to estimate the term occurrence probabilities $\mathrm{p}_1(t|D_i)$ (7.1) is plotted (top1). As we saw previously in Figure 7-4, performance of the probabilistic retrieval model with automatic feedback is significantly better than the baseline retrieval model.

Performance using the $N$-best ($N$=100) recognition hypotheses to estimate $\mathrm{p}_2(t|D_i)$ (7.2) is plotted next (nbest). Performance is slightly but consistently improved over that of using just the top one recognition hypothesis. The use of alternative recognition hypotheses allows additional terms to be included in the document representation and increases the chance of capturing the correct terms. The use of multiple hypotheses also permits a better estimate of the occurrence probability of the hypothesized terms: the more often a term appears in the top $N$ hypotheses, the more likely it is to have actually occurred. Using this method can significantly increase the number of terms added to the document representations. This is illustrated in Figure 7-6 which plots the number of unique indexing terms for the entire document collection for different length ($n = 2, \ldots, 6$) phonetic subword units. Comparing the plots for the top one recognition hypothesis (top1) and the $N$-best recognition hypotheses (nbest), we see that the number of terms grows with the length of the subword units. The number of terms can almost triple using the $N$-best outputs.

Next, performance using the expanded term set approach to compute term occurrence probabilities $\mathrm{p}_3(t|D_i)$ (7.3) is shown (expand). Performance of the short subword unit ($n$=2) gets worse. This is due to an increase in the number of spurious matches caused by the expanded set of terms. The additional terms are likely to match terms that occur in many of the documents due to the short length of the units and the small number of possible terms ($41^2 = 1681$). The performance of the longer subword units ($n$=3,4,5,6), however, are significantly improved. In this case, the expanded set of terms are allowing matches between the clean query terms and the noisy document terms but the longer subword unit sequence length makes it more difficult to get spurious matches. We saw similar behavior

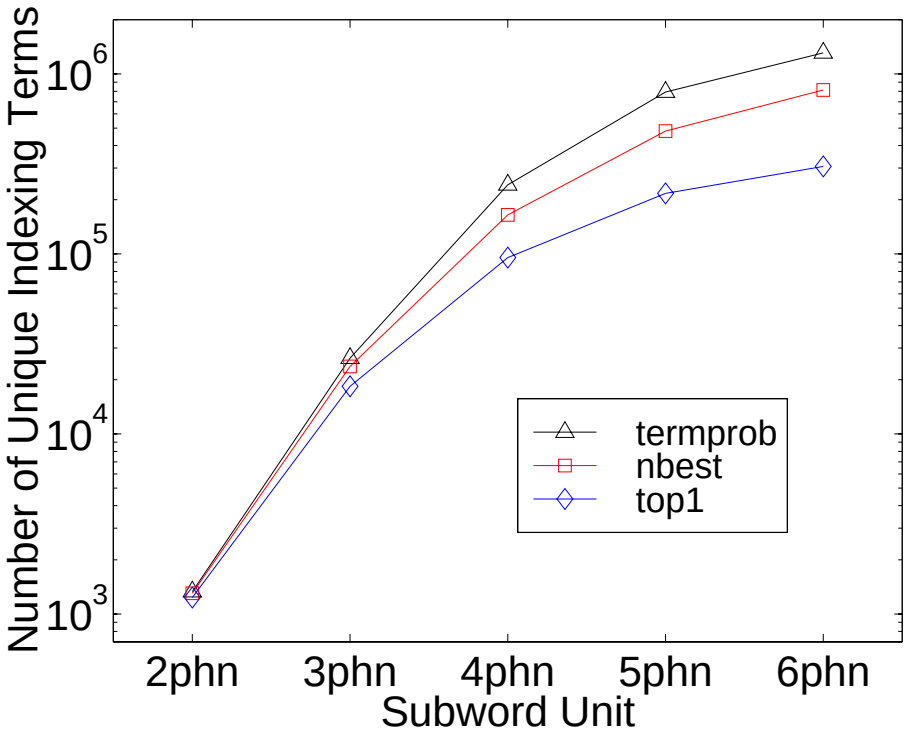


Figure 7-6: The number of unique indexing terms for the document collection for different length ($n = 2, \ldots, 6$) phonetic subword units. Three different methods for determining the indexing terms are shown: using the top one recognition hypothesis (top1), using the $N$=100 $N$-best recognition hypotheses (nbest), and using the term occurrence probabilities computed directly by the recognizer (termprob).

with the use of approximate term matching in Section 5.3. We note that the length $n$=4 subword units now outperform the length $n$=3 units. Also, the performance of the longer subword units ($n$=5,6) are now much closer to the medium length units ($n$=3,4) than before. Previously performance dropped off rapidly as the subword units got longer (e.g., the baseline or top1 curves). This is no longer the case. With the expanded set of terms, the issue of longer subword units being too specific and not matching enough terms has become less of an issue.

Finally, the use of term occurrence probabilities, $\mathrm{p}_4(t|D_i)$, computed directly by the recognizer is shown (termprob). Performance of the short subword unit ($n$=2) is poor and is actually worse than using expanded term sets. The problem of spurious matches is magnified in this case because even more term possibilities are created using the term

probability approach. The performance of the other subword units ($n$=3,4,5,6), however, are all improved and are better than using the expanded term set approach. Similar to the behavior we saw above with expanded term sets, the additional document terms generated by the term probability approach are allowing more matches with the clean query terms but the longer subword unit sequence length is reducing the number of spurious matches resulting in a net positive effect. Even more terms are generated by the term probability approach than with the $N$-best approach. Again, this is illustrated in Figure 7-6 which plots the number of unique indexing terms for the document collection for different length ($n = 2, \ldots, 6$) phonetic subword units. Comparing the plots, we see that the number of terms associated with the term probability approach (termprob) is much larger than the number of terms from using either the $N$-best (nbest) or top one (top1) recognition hypotheses. The number of terms can increase almost four-fold over using the one best recognition output.

Overall, we see that spoken document retrieval performance improves as more sophisticated estimates of the term occurrence probabilities are used. The combined factors of more term hypotheses and improved probability of occurrence estimates to appropriately weight the additional terms lead to better retrieval performance. The best performance is obtained using term occurrence probabilities computed directly from the speech recognizer: $\mathrm{p}_4(t|D_i)$. The integrated approach improves spoken document retrieval performance using subword units by over 28% from the baseline: mAP=0.52 to mAP=0.67. This improvement is better than the 23% gain (mAP=0.52 to mAP=0.64) obtained using the robust methods described in Chapter 5.

## 7.4 Summary

In this chapter, we presented a novel approach to spoken document retrieval where the speech recognition and information retrieval components are more tightly integrated. This was accomplished by developing new recognizer and retrieval models where the interface between the two components is better matched and the goals of the two components are consistent with each other and with the overall goal of the combined system. We presented

a new probabilistic retrieval model which makes direct use of term occurrence probabilities that can be computed by the recognizer. We then described several ways to compute the desired term probabilities including using the top one recognition hypothesis, using $N$-best recognition hypotheses, expanding the term set to include approximate match terms, and modifying the speech recognizer to enable it to output the term occurrence probabilities directly. We evaluated the performance of the integrated approach using the NPR corpus. We found that the probabilistic model performs slightly better than the baseline vector space retrieval model and the addition of automatic relevance feedback resulted in a significant performance improvement. We then measured the retrieval performance of the integrated approach as different methods for estimating the term occurrence probabilities are used. We found that retrieval performance improves as more sophisticated estimates are used. The best performance was obtained using term occurrence probabilities computed directly from the speech recognizer. The integrated approach improved retrieval performance by over 28% from the baseline. This is compared to an improvement of 23% using the robust methods described in Chapter 5.

# Chapter 8

# Summary and Future Work

This thesis explored approaches to the problem of spoken document retrieval (SDR), which is the task of automatically indexing and then retrieving relevant items from a large collection of recorded speech messages in response to a user specified natural language text query. We investigated the use of subword unit representations for SDR as an alternative to words generated by either keyword spotting or continuous speech recognition. Our investigation is motivated by the observation that word-based retrieval approaches face the problem of either having to know the keywords to search for *a priori*, or requiring a very large recognition vocabulary in order to cover the contents of growing and diverse message collections. The use of subword units in the recognizer constrains the size of the vocabulary needed to cover the language; and the use of subword units as indexing terms allows for the detection of new user-specified query terms during retrieval. Four research issues were addressed:

1. What are suitable subword units and how well can they perform?
2. How can these units be reliably extracted from the speech signal?
3. What is the behavior of the subword units when there are speech recognition errors and how well do they perform?
4. How can the indexing and retrieval methods be modified to take into account the fact that the speech recognition output will be errorful?

In this thesis, we made the following contributions to research in the area of spoken document retrieval:

- An empirical study of the ability of different subword units to perform spoken document retrieval and their behavior in the presence of speech recognition errors.
- The development of a number of robust indexing and retrieval methods that can improve retrieval performance when there are speech recognition errors.
- The development of a novel spoken document retrieval approach with a tighter coupling between the recognition and retrieval components that results in improved retrieval performance when there are speech recognition errors.
- The development of a novel probabilistic information retrieval model that achieves state-of-the-art performance on standardized text retrieval tasks.

The main goal of this research was to investigate the feasibility of using subword unit representations for spoken document retrieval as an alternative to word units generated by either keyword spotting or continuous speech recognition. We mentioned some word-based approaches to spoken document retrieval, but we did not report on their performance levels and did not directly compare the performance of word-based methods to subword based methods. Here we briefly describe the performance of state-of-the-art word-based approaches to spoken document retrieval on the TREC-7 SDR task (Garofolo et al. 1998). The predominate word-based approach is to first transform the spoken documents into text using a large vocabulary speech recognizer and then to use a conventional full-text retrieval system to perform retrieval. With word recognition error rates (in percent) in the mid to high 20's, the best speech retrieval performance reaches 90% of the performance of clean text transcriptions. Compared to subword units, the degradation due to speech recognition errors with word units is much less. However, these performance results cannot be directly compared because of a number of factors. First, the systems are evaluated on different spoken document collections. Second, the number of parameters in the speech recognition systems (i.e., system complexity) and the amount of data used to train them are very different. The large vocabulary systems contain recognition vocabularies on the order of 70,000 words, and use close to one hundred hours of transcribed speech for acoustic model training and hundreds of millions of words from text documents for language model training. The subword phonetic recognizer, in contrast, only has a vocabulary of 61 phones and uses

only 5 hours of speech for training the acoustic models and hundreds of thousands of phone occurrences for training the language models. It is also important to note that in the large vocabulary methods, almost all of the words in the test queries happen to be included in the recognition vocabulary. This means that there are essentially no out-of-vocabulary (OOV) query words in the test. It is precisely this issue of new query words that can be problematic for the large vocabulary word-based approach. In these experiments, the OOV problem is not being tested and the effects of OOV query words on retrieval performance are not being explored. Both word-based and subword-based approaches to SDR have advantages and disadvantages. Exploring methods to effectively combine both approaches is an interesting area for future research.

In the following sections, we give a brief summary of the main chapters in this thesis and finally close by mentioning some possible directions for future work.

## 8.1 Feasibility of Subword Units for Information Retrieval

We explored a range of subword units of varying complexity derived from error-free phonetic transcriptions and measured their ability to effectively index and retrieve speech messages. These experiments provide an *upper bound* on the performance of the different subword units since they assume that the underlying phonetic recognition is error-free. In particular, we examined overlapping, fixed-length phone sequences and broad phonetic class sequences, and non-overlapping, variable-length, phone sequences derived automatically (multigrams) and by rule (syllables). We found that many different subword units are able to capture enough information to perform effective retrieval. We saw that overlapping subword units perform better than non-overlapping units. There is also a tradeoff between the number of phonetic class labels and the sequence length required to achieve good performance. With the appropriate choice of subword units it is possible to achieve retrieval performance approaching that of text-based word units if the underlying phonetic units are recognized correctly. Although we were able to automatically derive a meaningful set of subword "stop" terms, experiments using the stop-list did not result in significant improvements in retrieval performance.

## 8.2 Extracting Subword Units from Spoken Documents

We trained and tuned a phonetic recognizer to operate on the radio broadcast news domain and used it to process the entire spoken document collection to generate phonetic transcriptions. We then explored a range of subword unit indexing terms of varying complexity derived from these errorful phonetic transcriptions and measured their ability to perform spoken document retrieval. We found that in the presence of phonetic recognition errors, retrieval performance degrades, as expected, compared to using error-free phonetic transcriptions or word-level text units: performance falls to 60% of the clean reference performance. However, many subword unit indexing terms are still give reasonable performance even without the use of any error compensation techniques. We also observed that there is a strong correlation between recognition and retrieval performance: better phonetic recognition performance leads to improved retrieval performance. These experiments establish a *lower bound* on the retrieval performance of the different subword units since no error compensation techniques are used. We know that there are speech recognition errors, but we are not doing anything about them. Hopefully improving the performance of the recognizer and developing robust indexing and retrieval methods to deal with the recognition errors will help improve retrieval performance.

## 8.3 Robust Indexing and Retrieval Methods

We investigated a number of robust methods in an effort to improve spoken document retrieval performance when there are speech recognition errors. In the first approach, the original query is modified to include near-miss terms that could match erroneously recognized speech. The second approach involves developing a new document-query retrieval measure using approximate term matching designed to be less sensitive to speech recognition errors. In the third method, the document is expanded to include multiple recognition candidates to increase the chance of capturing the correct hypothesis. The fourth method modifies the original query using automatic relevance feedback to include new terms as well as approximate match terms. The last method involves combining information from multiple subword unit representations. We studied the different methods individually and then

explored the effects of combining them. We found that using a new approximate match retrieval metric, modifying the queries via automatic relevance feedback, and expanding the documents with $N$-best recognition hypotheses improved performance; subword unit fusion, however, resulted in only marginal gains. Combining the approaches resulted in additive performance improvements. Using these robust methods improved retrieval performance using subword units generated from errorful phonetic recognition transcriptions by 23%.

## 8.4 Probabilistic Information Retrieval Model

We presented a novel probabilistic information retrieval model and demonstrated its capability to achieve state-of-the-art performance on large standardized text collections. The retrieval model scores documents based on the relative change in the document likelihoods, expressed as the ratio of the conditional probability of the document given the query and the prior probability of the document before the query is specified. Statistical language modeling techniques are used to compute the document likelihoods and the model parameters are estimated automatically and dynamically for each query to optimize well-specified maximum likelihood objective functions. An automatic relevance feedback strategy that is specific to the probabilistic model was also developed. The procedure automatically creates a new query (based on the original query and a set of top-ranked documents from a preliminary retrieval pass) by selecting and weighting query terms so as to maximize the likelihood ratio scores of the set of documents presumed to be relevant to the query. To benchmark the performance of the new retrieval model, we used the standard ad hoc text retrieval tasks from the TREC-6 and TREC-7 text retrieval conferences. Official evaluation results on the 1999 TREC-8 ad hoc text retrieval task were also reported. Experimental results indicated that the model is able to achieve performance that is competitive with current state-of-the-art retrieval approaches.

## 8.5 Integrated Recognition and Retrieval

We presented a novel approach to spoken document retrieval where the speech recognition and information retrieval components are more tightly integrated. This was accomplished

by developing new recognizer and retrieval models where the interface between the two components is better matched and the goals of the two components are consistent with each other and with the overall goal of the combined system. We presented a new probabilistic retrieval model which makes direct use of term occurrence probabilities that can be computed by the recognizer. We then described several ways to compute the desired term probabilities including using the top one recognition hypothesis, using $N$-best recognition hypotheses, expanding the term set to include approximate match terms, and modifying the speech recognizer to enable it to output the term occurrence probabilities directly. We evaluated the performance of the integrated approach using the NPR corpus. We found that the probabilistic model performs slightly better than the baseline vector space retrieval model and the addition of automatic relevance feedback resulted in a significant performance improvement. We then measured the retrieval performance of the integrated approach as different methods for estimating the term occurrence probabilities are used. We found that retrieval performance improves as more sophisticated estimates are used. The best performance was obtained using term occurrence probabilities computed directly from the speech recognizer. The integrated approach improved retrieval performance by over 28% from the baseline. This is compared to an improvement of 23% using the robust methods described in Chapter 5.

## 8.6 Future Directions

The experimental results presented in this thesis demonstrate that subword-based approaches to spoken document retrieval are feasible and merit further research. There are a large number of areas for extension of this work. In this section, we mention some of these possible directions for future work.

One area is to improve the performance of the extraction of the subword units from the speech signal. Although we investigated the use of different acoustic and language models in the speech recognizer in an effort to improve phonetic recognition performance in Section 4.2, much additional work can be done. For example, more training data can be used to improve model robustness and more detailed and complex models can be used to try

to capture more information from the speech signal. Another approach would be to modify the speech recognizer to recognize the subword units, e.g., broad class units, syllables, or other multi-phone sequences, *directly* from the speech, rather than constructing them from a phonetic string. An advantage of this approach is that the recognizer can be optimized specifically for the subword units of interest instead of for an intermediate set of phonetic units. In addition, the recognition units are larger and should be easier to recognize. A disadvantage is that the vocabulary size is significantly increased. The key tradeoff that needs to be considered is the selection of an inventory of subword units that is both restricted in size but can still provide good coverage for the application domain. In Section 7.2, we described one method of modifying the speech recognizer to estimate probabilities of term occurrence. Alternate methods for computing this type of quantity, such as the scoring functions used in traditional wordspotting systems (Rohlicek et al. 1989; Rose and Paul 1990), should also be examined.

Another area of work is to improve the probabilistic information retrieval model. In the current implementation of our retrieval model described in Chapter 6, simple unigram language models are used to estimate the probabilistic quantities $\mathrm{p}(Q|D_i)$ and $\mathrm{p}(Q)$. More sophisticated models such as higher order statistical $n$-gram language models can also be used. Although these models make fewer assumptions about word independence and can capture longer range context such as phrases, they impose increased demands on the quantity of training data needed for parameter estimation. As mentioned in Section 6.2.1, alternative smoothing techniques for the probability models can also be explored. In addition to the use of more complex models mentioned above, another extension is to increase the flexibility of the model by allowing "approximate" term matching. For example, the occurrence probability of term $t$ in document $D_i$, $\mathrm{p}(t|D_i)$, can be computed according to (7.3):

$$\mathrm{p}(t|D_i) = \sum_{t^*} \mathrm{p}(t \,|\, t^*, D_i)\, \mathrm{p}(t^*|D_i)$$

where $\mathrm{p}(t \,|\, t^*, D_i)$ is an appropriate measure of the similarity between terms $t$ and $t^*$, and the summation is over all possible approximate matches, $t^*$, for term $t$. This extension was used in Section 7.2 to deal with transcription errors in the documents by summing over an appropriately expanded set of terms, $t^*$, for each original term, $t$. This extension can also

allow the retrieval model to deal with the use of synonyms in a principled way. The probability of a term $t$ is computed by summing over appropriately weighted probabilities of its synonyms $\{t^*\}$. Similarly, this extension can also be applied to cross language retrieval (the document collection is in one language and the query is in another language) by capturing the mapping of terms from one language to the other. In Section 7.2, we used a simple model based on the phonetic recognition error confusion matrix to compute $\mathrm{p}(t \mid t^*, D_i)$, the probability that term $t$ can be recognized as term $t^*$. Other more complex models of the relationship between terms $t$ and $t^*$ remains to be explored.

Another interesting and potentially profitable area of research is on information fusion methods for spoken document retrieval. In Section 5.6, we briefly looked at some simple methods for combining multiple subword unit representations. Although the performance improvements we obtained were small, the method of information combination still holds promise. As we mentioned before, maybe the use of more sophisticated non-linear combination methods such as bagging (Breiman 1994), boosting (Freund and Schapire 1996), or stacking (Wolpert 1992) will lead to better performance. In addition to multiple subword units, other types of information can also be combined. For example, multiple recognition hypotheses from different automatic speech recognizers can be combined or subword units can be combined with word units.

Since subword-based approaches to spoken document retrieval have reduced demands on the amount of required training data, it should be easier and faster to port subword-based SDR approaches to new application domains and new languages. This hypothesis should be verified by trying to bring up SDR systems in different domains and languages. It will also be interesting to explore the use of subword-based approaches on multi-lingual speech message collections.

Finally, the methods presented in this thesis should be evaluated on larger sets of data. This includes both the spoken document collection and the training set for the speech recognizer. More data will allow us to build more robust models and to further test the scalability and behavior of our systems.

# Appendix A

# Details of the NPR Speech Corpus

## A.1 Description of the Query Set

Table A-1 list the 50 queries in the NPR Corpus. The query identification number, the text of the query, and the number of relevant documents is shown for each query.

## A.2 Subword Unit Frequency Statistics

This section contains tables of the 100 most frequent subword units and their frequency of occurrence counts. The subword units are generated from clean phonetic transcriptions of the NPR corpus.

Table A-2 lists the 100 most frequent phonetic trigram ($n$=3) subword units. A description of the phone labels used can be found in Tables 4-1 and 4-4. Table A-3 list the broad class subword units ($c$=20, $n$=4). The mapping of the 41 phone labels to the 20 broad classes is shown in Table A-4. The mapping is determined by hierarchical clustering based on acoustic similarity as described in Section 3.2.2 and illustrated in Figure 3-2. Table A-5 lists the automatically derived multigram ($m$=4) subword units. Finally, Table A-6 lists the most frequent syllable subword units.

| Id | Query | # Rel. Docs |
|---|---|---|
| 1 | Commuter traffic updates and reports | 35 |
| 2 | Weather forecast | 27 |
| 3 | Sports news | 10 |
| 4 | Bob Dole's campaign for the presidency | 24 |
| 5 | Republican presidential primaries | 21 |
| 6 | Conflict in Bosnia | 12 |
| 7 | Business and financial news | 12 |
| 8 | Whitewater controversy hearings investigations and trials | 10 |
| 9 | Israel and the Palestinians | 12 |
| 10 | Space shuttle | 8 |
| 11 | Politics in Russia: the Russian presidential election | 6 |
| 12 | Military actions in Iraq | 7 |
| 13 | Dayton peace agreement | 7 |
| 14 | Auto workers' strike at General Motors | 6 |
| 15 | Temporary spending bill | 5 |
| 16 | Immigration reform bill | 4 |
| 17 | Terrorism is condemned by world leaders at a summit in Egypt | 5 |
| 18 | GOP republican national convention in San Diego | 5 |
| 19 | Boston Red Sox baseball team | 5 |
| 20 | Capital gains tax | 4 |
| 21 | Drug traffic from Mexico | 2 |
| 22 | Hearings on the branch Davidians and Waco | 4 |
| 23 | Occupied territories on the West Bank and Ghaza Strip | 5 |
| 24 | Remedial education courses | 4 |
| 25 | John Salvi trial | 4 |
| 26 | Bank of Boston announcement | 3 |
| 27 | The United States and China avoid trade war over copyright violations | 3 |
| 28 | Human rights in China | 2 |
| 29 | Governor William Weld | 4 |
| 30 | Boston oil spill | 3 |
| 31 | Tensions between China and Taiwan | 4 |
| 32 | IRA bomb explosion in England | 3 |
| 33 | Doctor assisted suicide ruling | 3 |
| 34 | University of Massachusetts | 3 |
| 35 | US Supreme Court | 3 |
| 36 | Health insurance programs | 3 |
| 37 | Telecommunications bill | 3 |
| 38 | Proposal to increase the minimum wage | 2 |
| 39 | F. Lee Bailey jail term | 2 |
| 40 | Cuba shoots down civilian plane | 2 |
| 41 | Britain's sale of weapons to Iraq | 1 |
| 42 | Airplanes forced to land | 2 |
| 43 | Bosnian war crimes tribunal | 3 |
| 44 | Ban on assault weapons | 2 |
| 45 | Welfare reform bill | 2 |
| 46 | Massachusetts board of education | 2 |
| 47 | Fiscal ninety seven federal budget proposal | 2 |
| 48 | Mad cow disease | 2 |
| 49 | Treatment for AIDS | 2 |
| 50 | Deregulation of public utility companies | 3 |

Table A-1: Table of the 50 queries in the NPR Corpus.

| Term | Freq. | Term | Freq. | Term | Freq. |
|---|---|---|---|---|---|
| ae_n_d | 3593 | ih_t_s | 667 | uw_dh_ax | 515 |
| ax_n_t | 2475 | n_d_dh | 652 | p_ao_r | 515 |
| sh_ax_n | 2178 | ao_r_t | 650 | s_eh_n | 514 |
| dh_ae_t | 1869 | t_ix_ng | 623 | ah_n_d | 513 |
| f_ao_r | 1517 | d_dh_ax | 622 | n_s_t | 497 |
| m_ax_n | 1261 | ax_n_d | 621 | p_r_eh | 490 |
| n_dh_ax | 1244 | ax_n_ax | 619 | n_t_uw | 490 |
| k_ax_n | 1160 | s_t_r | 617 | w_eh_n | 488 |
| ax_n_s | 1030 | p_ax_l | 611 | z_ax_n | 487 |
| t_ax_n | 1025 | ix_ng_t | 611 | eh_n_d | 485 |
| ah_v_dh | 1005 | z_ae_n | 610 | ax_t_iy | 480 |
| ey_sh_ax | 979 | eh_s_t | 606 | s_ah_m | 476 |
| t_ax_d | 966 | w_aa_z | 605 | z_t_uw | 475 |
| ax_n_z | 954 | f_ow_r | 601 | ax_l_iy | 471 |
| t_dh_ax | 940 | z_dh_ax | 600 | ix_k_s | 468 |
| dh_eh_r | 935 | aa_r_t | 597 | n_ix_ng | 467 |
| eh_n_t | 883 | eh_r_iy | 591 | z_ah_v | 465 |
| v_dh_ax | 875 | ae_t_dh | 591 | n_d_ix | 465 |
| dh_ax_s | 865 | f_r_ah | 590 | v_ax_n | 463 |
| dh_ih_s | 850 | t_ah_v | 589 | b_ax_l | 460 |
| ix_n_dh | 781 | dh_ax_p | 588 | p_r_ax | 457 |
| hh_ae_v | 766 | m_ao_r | 582 | ax_d_ax | 453 |
| s_t_ax | 751 | n_aa_t | 580 | n_t_r | 451 |
| w_ih_dh | 748 | r_ah_m | 572 | n_ay_n | 450 |
| ax_k_ax | 747 | d_t_uw | 569 | t_ix_n | 448 |
| n_t_s | 716 | n_t_ax | 568 | ax_d_ey | 448 |
| d_ax_n | 711 | t_t_uw | 563 | dh_ax_m | 445 |
| ax_s_t | 692 | n_ax_l | 551 | dh_ax_f | 445 |
| dh_ax_k | 686 | ix_ng_dh | 548 | eh_k_t | 440 |
| n_t_iy | 679 | hh_ae_z | 548 | t_ae_n | 439 |
| w_ah_n | 674 | z_ix_n | 541 | y_uw_n | 437 |
| b_ah_t | 672 | w_ih_l | 537 | p_aa_r | 434 |
| n_d_ax | 669 | s_t_ey | 536 | | |
| k_ae_n | 667 | t_uw_dh | 520 | | |

Table A-2: Table of the 100 most frequent phonetic trigram ($n$=3) subword units generated from clean phonetic transcriptions of the NPR corpus.

| Term | Freq. | Term | Freq. | Term | Freq. |
|---|---|---|---|---|---|
| flx_nsl_ust_blx | 1104 | vfr_blx_ust_rtr | 410 | liq_flx_nsl_ust | 318 |
| flx_nsl_vfr_blx | 1016 | flx_nsl_ust_rtr | 407 | blx_nsl_sfr_ust | 317 |
| eyy_pfr_blx_nsl | 972 | lbv_nsl_vst_ust | 406 | lfv_vst_liq_blx | 317 |
| sfr_lbv_nsl_vst | 914 | ust_pfr_blx_nsl | 396 | ust_wfr_ooo_rtr | 314 |
| nsl_blx_nsl_ust | 815 | nsl_frt_blx_nsl | 394 | blx_ust_blx_liq | 312 |
| lfv_vfr_vfr_blx | 809 | rtr_low_sfr_blx | 393 | ust_blx_vst_eyy | 309 |
| ust_lbv_nsl_vst | 752 | ust_lfv_vst_liq | 389 | vfr_blx_ust_lfv | 309 |
| nsl_ust_blx_nsl | 690 | blx_nsl_blx_liq | 381 | blx_ust_lbv_nsl | 309 |
| ust_blx_nsl_sfr | 670 | flx_sfr_ust_rtr | 375 | blx_vst_low_nsl | 308 |
| ust_ooo_rtr_ust | 656 | lbv_ust_vfr_blx | 374 | liq_lfv_nsl_ust | 308 |
| ust_blx_nsl_ust | 652 | sfr_flx_ust_sfr | 373 | ust_flx_nsl_vfr | 308 |
| sfr_ust_blx_nsl | 629 | wfr_ooo_rtr_nsl | 372 | flx_nsl_flx_nsl | 307 |
| blx_ust_blx_nsl | 607 | nsl_vfr_blx_ust | 371 | blx_nsl_lfv_vfr | 307 |
| wfr_rtr_lfv_nsl | 589 | flx_pfr_blx_nsl | 371 | sfr_wfr_ooo_rtr | 306 |
| lfv_nsl_vst_rtr | 562 | ust_ust_blx_vst | 370 | sfr_flx_nsl_ust | 305 |
| vst_blx_nsl_ust | 558 | low_nsl_ust_frt | 367 | lfv_nsl_ust_rtr | 305 |
| flx_ust_sfr_ust | 539 | vfr_lbv_ust_vfr | 365 | nsl_vst_flx_nsl | 302 |
| pfr_blx_nsl_sfr | 538 | nsl_ust_blx_vst | 361 | blx_nsl_ust_blx | 300 |
| ust_lfv_rtr_ust | 535 | nsl_wfr_ooo_rtr | 359 | afr_lfv_sfr_ust | 300 |
| nsl_lbv_nsl_vst | 530 | low_sfr_blx_vst | 353 | lbv_nsl_vst_liq | 299 |
| blx_vst_blx_nsl | 523 | blx_ust_lfv_nsl | 351 | vst_flx_sfr_ust | 296 |
| lbv_nsl_vst_vfr | 514 | rtr_nsl_blx_nsl | 347 | lbv_nsl_vst_flx | 295 |
| blx_vst_lfv_ust | 501 | nsl_dip_nsl_ust | 341 | vst_lfv_vfr_rtr | 294 |
| ust_blx_vfr_blx | 475 | frt_ust_blx_liq | 340 | sfr_flx_nsl_vfr | 293 |
| sfr_vfr_lbv_ust | 469 | pfr_blx_nsl_blx | 333 | nsl_vst_blx_vst | 291 |
| ust_lfv_nsl_ust | 468 | nsl_vst_rtr_blx | 333 | ust_lfv_nsl_sfr | 290 |
| blx_nsl_ust_sfr | 464 | ust_blx_ust_blx | 331 | ust_vfr_lbv_ust | 290 |
| lfv_nsl_vfr_blx | 456 | wfr_ooo_rtr_vfr | 330 | vst_lbv_nsl_vst | 288 |
| sfr_ust_blx_vst | 455 | blx_nsl_lbv_nsl | 327 | nsl_vfr_lbv_ust | 288 |
| ust_rtr_low_sfr | 453 | ust_flx_nsl_ust | 326 | vfr_blx_sfr_ust | 284 |
| sfr_ust_eyy_ust | 442 | dip_nsl_ust_frt | 326 | liq_low_nsl_ust | 282 |
| ust_frt_ust_blx | 437 | lbv_nsl_vst_blx | 323 | ust_lbv_nsl_ust | 281 |
| sfr_low_nsl_ust | 418 | nsl_vst_vfr_blx | 322 | | |
| sfr_blx_vst_blx | 416 | wfr_ooo_rtr_ust | 320 | | |

Table A-3: Table of the 100 most frequent broad class subword units ($c$=20, $n$=4) generated from clean phonetic transcriptions of the NPR corpus.

| Class | Phones | Class | Phones | Class | Phones | Class | Phones |
|---|---|---|---|---|---|---|---|
| **afr** | ch jh | **blx** | ax uh uw | **dip** | ay oy | **eyy** | ey |
| **flx** | ih ix | **frt** | iy y | **hhh** | hh | **lbv** | ae |
| **lfv** | aa ah aw | **liq** | l w | **low** | eh | **nsl** | m n ng |
| **ooo** | ao ow | **pfr** | sh zh | **rtr** | er r | **sfr** | s z |
| **ust** | k p t | **vfr** | dh v | **vst** | b d g | **wfr** | f th |

Table A-4: Mapping of the 41 phone labels to the 20 broad classes. The mapping is determined by hierarchical clustering based on acoustic similarity.

| Term | Freq. | Term | Freq. | Term | Freq. |
|---|---|---|---|---|---|
| dh_ax | 3789 | w_ih_l | 432 | s_eh_z | 267 |
| ae_n_d | 2652 | aa_r | 425 | ax_n_z | 266 |
| ix_ng | 1961 | er | 402 | ax_d | 266 |
| t_uw | 1900 | d | 395 | aa_n_dh_ax | 258 |
| ax_n | 1835 | sh_ax_n | 383 | b_ay | 256 |
| ix_n | 1557 | dh_ey | 382 | s_t | 254 |
| dh_ae_t | 1546 | s_ah_m | 379 | ax_d_ax_n | 252 |
| s | 1544 | s_t_ey_t | 370 | t_s | 251 |
| ax_l | 1527 | m_ao_r | 368 | ix_ng_dh_ax | 245 |
| ah_v | 1331 | ax_b_aw_t | 367 | z_ae_n_d | 244 |
| t | 1136 | r_iy | 366 | w_iy | 244 |
| f_ao_r | 1054 | aa_n | 365 | t_ax_d_ey | 242 |
| z | 954 | f_ow_r | 362 | g_ah_v_er | 241 |
| ih_z | 918 | p_aa_r_t | 360 | ix_k_s | 240 |
| ax | 847 | hh_ih_z | 355 | n_ay_n_t | 237 |
| ey_sh_ax_n | 811 | p_ao_r_t | 351 | ow_v_er | 236 |
| dh_ih_s | 789 | p_r_eh_z | 342 | m_ow_s_t | 232 |
| dh_eh_r | 784 | k_ax_n | 326 | ax_n_t_s | 227 |
| ah_v_dh_ax | 741 | y_uw | 320 | s_ow | 226 |
| m_ax_n_t | 673 | ax_s | 320 | ix_ng_t_uw | 226 |
| ix_n_dh_ax | 617 | t_uw_dh_ax | 317 | t_iy | 225 |
| w_ih_dh | 591 | w_ah_t | 304 | k_ah_m | 224 |
| w_ah_n | 581 | iy | 303 | t_uw_b_iy | 222 |
| hh_ae_v | 575 | jh_ah_s_t | 300 | n_ow | 216 |
| b_ah_t | 567 | er_z | 297 | ax_t | 216 |
| l_iy | 561 | b_iy | 297 | ao_l_s_ow | 214 |
| ax_n_t | 559 | ao_l | 295 | ae_t | 211 |
| f_r_ah_m | 556 | ah_dh_er | 290 | l_ae_s_t | 210 |
| ih_t_s | 535 | w_uh_d | 284 | hh_ih_r | 210 |
| n_aa_t | 532 | s_eh_n_t | 284 | g_eh_t | 210 |
| w_aa_z | 529 | ax_n_s | 283 | ax_k_ax_n | 210 |
| ih_t | 506 | b_ih_n | 281 | n_ax_l | 208 |
| hh_iy | 469 | ae_z | 275 | | |
| hh_ae_z | 459 | t_ax_d | 270 | | |

Table A-5: Table of the 100 most frequent multigram ($m$=4) subword units generated from clean phonetic transcriptions of the NPR corpus.

| Term | Freq. | Term | Freq. | Term | Freq. |
|---|---|---|---|---|---|
| dh_ax | 7761 | dh_ae | 726 | k_ae | 440 |
| t_uw | 3949 | r_ix | 724 | p_r_ax | 432 |
| ax | 2084 | m_eh | 723 | m_ao_r | 429 |
| t_ax | 1883 | p_iy | 703 | t_ih | 427 |
| t_iy | 1799 | dh_ih | 699 | hh_ae_z | 424 |
| n_ax | 1751 | p_ax | 684 | t_ax_d | 423 |
| l_iy | 1696 | hh_iy | 681 | m_ey | 420 |
| r_iy | 1565 | dh_ey | 676 | t_ax_n | 419 |
| s_ax | 1501 | b_ix | 661 | s_t_ax | 418 |
| t_er | 1316 | s_ow | 651 | n_uw | 417 |
| z_ax | 1288 | v_ax | 646 | p_ah | 415 |
| dh_ae_t | 1151 | dh_eh_r | 634 | m_ax_n_t | 408 |
| d_ix | 1117 | s_iy | 615 | s_ey | 401 |
| d_ax | 1082 | hh_ae | 577 | w_er | 399 |
| r_ax | 1054 | hh_ae_v | 565 | hh_uw | 394 |
| b_iy | 1042 | b_ay | 561 | n_ay_n | 392 |
| sh_ax_n | 1040 | t_ix_ng | 554 | b_ah_t | 391 |
| sh_ax | 1012 | n_ow | 551 | p_r_eh | 390 |
| n_iy | 1006 | dh_er | 551 | ih | 388 |
| v_er | 988 | g_ow | 533 | f_r_ah_m | 387 |
| f_ao_r | 971 | f_ow_r | 511 | ow | 386 |
| l_ax | 870 | m_ih | 499 | t_ah_v | 384 |
| k_ax_n | 864 | eh | 484 | n_aa | 384 |
| m_ax | 860 | p_er | 479 | b_ah | 383 |
| ix_ng | 854 | ey | 475 | d_uw | 382 |
| k_ax | 852 | l_ih | 472 | r_eh | 379 |
| y_uw | 824 | ah_v | 472 | ix | 377 |
| ix_n | 820 | ae_n_d | 469 | k_ey | 370 |
| s_eh | 794 | z_ix_n | 466 | b_er | 369 |
| d_ey | 778 | d_ih | 459 | ae | 365 |
| d_iy | 747 | m_ax_n | 457 | z_ah_v | 363 |
| er | 746 | t_eh | 456 | w_ih_dh | 361 |
| s_ih | 739 | n_ix_ng | 455 | | |
| d_er | 730 | s_er | 444 | | |

Table A-6: Table of the 100 most frequent syllable subword units generated from clean phonetic transcriptions of the NPR corpus.

# Bibliography


Acero, A. and R. Stern (1990). Environmental robustness in automatic speech recognition. In *Proc. ICASSP '90*, Albuquerque, NM, pp. 849–852.

Bertsekas, D. (1982). *Constrained Optimization and Lagrange Multiplier Methods*. Cambridge, MA: Academic Press.

Breiman, L. (1994). Bagging predictors. Technical Report 421, Department of Statistics, University of California Berkeley.

Brieman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Buckley, C. (1985). Implementation of the smart information retrieval system. Technical Report 85-686, Computer Science Department, Cornell University, Ithaca, New York.

Carey, M. and E. Parris (1995). Topic spotting with task independent models. In *Proc. Eurospeech '95*, Madrid, Spain, pp. 2133–2136.

Carlson, B. A. (1996). Unsupervised topic clustering of SWITCHBOARD speech messages. In *Proc. ICASSP '96*, Atlanta, GA, pp. 315–318.

Cavnar, W. B. (1994). Using an n-gram-based document representation with a vector processing retrieval model. In D. K. Harman (Ed.), *Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, USA, pp. 269–278. National Institute for Standards and Technology. NIST-SP 500-226.

Chang, J. W. and J. R. Glass (1997). Segmentation and modeling in segment-based recognition. In *Proc. Eurospeech '97*, Rhodes, Greece, pp. 1199–1202.

Chase, L. (1997). Word and acoustic confidence annotation for large vocabulary speech recognition. In *Proc. Eurospeech '97*, Rhodes, Greece, pp. 815–818.

Chinchor, N. and B. Sundheim (1995). Message understanding conference MUC tests of discourse processing. In *AAAI 95 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford University, pp. 21–26.

Chomsky, N. and M. Halle (1968). *The Sound Pattern of English*. New York, NY: Harper & Row. republished in paperback, Cambridge, MA: MIT Press, 1991.

Cover, T. and J. Thomas (1991). *Elements of Information Theory*. New York: John Wiley and Sons.

Crestani, F., M. Lalmas, C. J. V. Rijsbergen, and I. Campbell (1998). Is this document relevant?...Probably: A survey of probabilistic models in information retrieval. *ACM Computing Surveys 30*(4), 528–552.

Croft, B. (1980). Model of cluster searching based on classification. *Information Systems 5*, 189–195.

Damashek, M. (1995). Gauging similarity via n-grams: Language-independent categorization of text. *Science 246*, 843–848.

Deligne, S. and F. Bimbot (1995). Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Proc. ICASSP '95*, Detroit, MI, pp. 169–172.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39*(1), 1–38.

Dharanipragada, S., M. Franz, and S. Roukos (1998). Audio indexing for broadcast news. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. New York, NY: John Wiley & Sons.

Fellbaum, C. (Ed.) (1998). *WordNet - An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Fisher, W. (1996). Automatic syllabification program based on algorithm in (Kahn 1976). Available at ftp://jaguar.ncsl.nist.gov/pub/tsylb2-1.1.tar.Z.

Foote, J., G. Jones, K. S. Jones, and S. Young (1995). Talker independent keyword spotting for information retrieval. In *Proc. Eurospeech '95*, Madrid, Spain, pp. 2145–2148.

Forney, G. (Mar., 1973). The viterbi algorithm. In *Proc. IEEE*, pp. 268–278.

Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of 12th International Conference*.

Gale, W. A. and G. Sampson (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics* (2), 217–237.

Garofolo, J., E. Voorhees, C. Auzanne, V. Stanford, and B. Lund (1998). 1998 TREC-7 spoken document retrieval track overview and results. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Garofolo, J., E. Voorhees, and V. Stanford (1997). 1997 TREC-6 spoken document retrieval track overview and results. In *Sixth Text REtrieval Conference (TREC-6)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-240.

Garofolo, J. S., L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren (1993). DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. National Institute of Standards and Technology, NISTIR 4930.

Gillick, L., J. Baker, and et. al. (1993). Application of large vocabulary continuous speech recognition to topic and speaker identification using telephone speech. In *Proc. ICASSP '93*, Volume 2, Minneapolis, MN, pp. II–471 – II–474.

Glass, J. (1988). *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition*. Ph. D. thesis, Massachusetts Institute of Technology.

Glass, J., J. Chang, and M. McCandless (1996). A probabilistic framework for feature-based speech recognition. In *Proc. ICSLP '96*, Volume 4, Philadelphia, PA, pp. 2277–2280.

Glavitsch, U. and P. Schauble (1992). A system for retrieving speech documents. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Language Processing, pp. 168–176.

Godfrey, J., E. Holliman, and J. McDaniel (1992). SWITCHBOARD: telephone speech corpus for research and development. In *Proc. ICASSP '92*, Volume 1, San Francisco, CA, pp. I–517 – I–520.

Halberstadt, A. K. (1998). *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition*. Ph. D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA.

Harman, D. K. (Ed.) (1994). *Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-226.

Harman, D. K. (Ed.) (1995). *Fourth Text REtrieval Conference (TREC-4)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-236.

Harman, D. K. (Ed.) (1997). *Sixth Text REtrieval Conference (TREC-6)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-240.

Harman, D. K. (Ed.) (1998). *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-242.

Harman, D. K. (Ed.) (1999). *Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP.

Hartigan, J. (1975). *Clustering Algorithms*. New York, NY: John Wiley & Sons.

Hauptmann, A. and H. Wactlar (1997). Indexing and search of multimodal information. In *Proc. ICASSP '97*, Munich, Germany, pp. 195–198.

Hazen, T. and A. Halberstadt (1998). Using aggregation to improve the performance of mixture gaussian acoustic models. In *Proc. ICASSP '98*, Seattle, WA, USA, pp. 653–657.

Hiemstra, D. and W. Kraaij (1998). Twenty-one at TREC-7: Ad-hoc and cross-language track. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Huber, P. (1981). *Robust Statistics*. New York: Wiley.

Huffman, S. (1995). Acquaintance: Language-independent document categorization by n-grams. In D. K. Harman (Ed.), *Fourth Text REtrieval Conference (TREC-4)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-236.

James, D. A. (1995). *The Application of Classical Information Retrieval Techniques to Spoken Documents*. Ph. D. thesis, University of Cambridge, Cambridge, U.K.

Jelinek, F. (1999). *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press.

Johnson, S., P. Jourlin, G. Moore, K. S. Jones, and P. Woodland (1998). Spoken document retrieval for TREC-7 at cambridge university. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Jones, G., J. Foote, K. S. Jones, and S. Young (1995a). Video mail retrieval: The effect of word spotting accuracy on precision. In *Proc. ICASSP '95*, Detroit, MI, pp. 309–312.

Jones, G., J. Foote, K. S. Jones, and S. Young (1995b). Video mail retrieval: The effect of word spotting accuracy on precision. In *Proc. ICASSP '95*, Detroit, MI, USA, pp. 309–312.

Jones, G. J. F., J. T. Foote, K. S. Jones, and S. J. Young (1996). Retrieving spoken documents by combining multiple index sources. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, pp. 30–39.

Jourlin, P., S. Johnson, K. S. Jones, and P. Woodland (1999). Improving retrieval on imperfect speech transcriptions. In *Proceedings of the 22st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 283–284.

Kahn, D. (1976). *Syllable-based Generalizations in English Phonology*. Ph. D. thesis, Department of Linguistics and Philosophy, Massachusetts Institute of Technology, Cambridge, MA.

Kupiec, J., J. Pedersen, and F. Chen (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, USA, pp. 68–73.

Lee, K.-F. (1989). *Automatic Speech Recognition: The Development of the SPHINX System*. Boston: Kluwer Academic Publishers.

Lewis, D. (1992). Representation and learning in information retrieval. Technical Report 91–93, Computer Science Dept., Univ. of Massachusetts.

Livescu, K. (1999). Analysis and modeling of non-native speech for automatic speech recognition. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA.

Marukawa, K., T. Hu, H. Fujisawa, and Y. Shima (1997). Document retrieval tolerating character recognition errors – evaluation and application. *Pattern Recognition 30*(8), 1361–1371.

McDonough, J. and H. Gish (1994). Issues in topic identification on the switchboard corpus. In *Proc. ICSLP '94*, Yokohama, Japan, pp. 2163–2166.

McDonough, J., K. Ng, P. Jeanrenaud, H. Gish, and J. R. Rohlicek (1994). Approaches to topic identification on the switchboard corpus. In *Proc. ICASSP '94*, Adelaide, Austrailia, pp. I–385 – I–388.

McLemore, C. (1997). Pronlex american english lexicon. URL http://morph.ldc.upenn.edu/Catalog/LDC97L20.html.

Miller, D., T. Leek, and R. Schwartz (1998). BBN at TREC-7: Using hidden markov models for information retrieval. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Ng, C. and J. Zobel (1998). Speech retrieval using phonemes with error correction. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 365–366.

Ng, K. (1998). Towards robust methods for spoken document retrieval. In *Proc. ICSLP '98*, Sydney, Australia.

Ng, K. (1999). A maximum likelihood ratio information retrieval model. In *Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, MD, USA. NIST-SP.

Ng, K. and V. Zue (1997a). An investigation of subword unit representations for spoken document retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Posters: Abstracts, pp. 339.

Ng, K. and V. Zue (1997b). Subword unit representations for spoken document retrieval. In *Proc. Eurospeech '97*, Rhodes, Greece, pp. 1607–1610.

Ng, K. and V. Zue (1998). Phonetic recognition for spoken document retrieval. In *Proc. ICASSP '98*, Seattle, WA, USA, pp. 325–328.

Nowell, P. and R. K. Moore (1994). A non-word based approach to topic spotting. In *Proceedings of the 14th Speech Research Symposium*.

Ponte, J. and W. B. Croft (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–281.

Porter, M. (1980). An algorithm for suffix stripping. *Program 14*(3), 130–137.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE 77*, 257–286.

Rijsbergen, C. V. (1979). *Information Retrieval*. London: Butterworths.

Robertson, S. and K. S. Jones (1976). Relevance weighting of search terms. *Journal of the American Society of Information Science*, 129–146.

Robertson, S., S. Walker, and M. Beaulieu (1998). Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Rohlicek, J., D. Ayuso, and et. al. (1992). Gisting conversational speech. In *Proc. ICASSP '92*, Volume 2, San Francisco, CA, pp. II–113 – II–116.

Rohlicek, J. R., W. Russell, S. Roukos, and H. Gish (1989). Continuous hidden Markov modeling for speaker-independent word spotting. In *Proc. ICASSP '89*, Glasgow, Scotland, pp. 627–630.

Rose, R. C., E. I. Chang, and R. P. Lippmann (1991). Techniques for information retrieval from voice messages. In *Proc. ICASSP '91*, Toronto, Canada, pp. 317–320.

Rose, R. C. and D. B. Paul (1990). A hidden Markov model based keyword recongition system. In *Proc. ICASSP '90*, Albuquerque, NM, pp. 129–132.

Salton, G. and M. J. McGill (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Sankoff, D. and J. Kruskal (1983). *Time Warps, string edits and macromolecules; the theory and practice of sequence comparison*, Chapter 5. Addison-Wesley.

Schäuble, P. and U. Glavitsch (1994). Assessing the retrieval effectiveness of a speech retrieval system by simulating recognition errors. In *Proc. ARPA Human Language Technology Workshop '94*, Princeton, NJ, USA, pp. 370–372.

Schmandt, C. (1994). *Voice Communication with Computers (Conversational Systems)*. New York: Van Nostrand Reinhold.

Siegler, M., A. Berger, A. Hauptmann, and M. Witbrock (1998). Experiments in spoken document retrieval at CMU. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Siegler, M., S. Slattery, K. Seymore, R. Jones, A. Hauptmann, and M. Witbrock (1997). Experiments in spoken document retrieval at CMU. In *Sixth Text REtrieval Conference (TREC-6)*, Gaithersburg, MD, USA. National Institute for Standards and Technology. NIST-SP 500-240.

Singhal, A., J. Choi, D. Hindle, D. Lewis, and F. Pereira (1998). AT&T at TREC-7. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, USA. NIST-SP 500-242.

Siu, M.-H., H. Gish, and F. Richardson (1997). Improved estimation,evaluation and applications of confidence measures for speech recognition. In *Proc. Eurospeech '97*, Rhodes, Greece, pp. 831–834.

Skilling, A., P. Nowell, and R.K.Moore (1995). Acoustic based topic spotting. In *Proceedings of the 15th Speech Research Symposium*.

Spina, M. S. and V. Zue (1996). Automatic transcription of general audio data: Preliminary analyses. In *Proc. ICSLP '96*, Volume 2, Philadelphia, PA, pp. 594–597.

Spina, M. S. and V. Zue (1997). Automatic transcription of general audio data: Effect of environment segmentation on phonetic recognition. In *Proc. Eurospeech '97*, Rhodes, Greece, pp. 1547–1550.

Sproat, R. (1998). *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Kluwer Academic Publishers.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory IT-13*, 260–269.

Wechsler, M., E. Munteanu, and P. Schauble (1998). New techniques for open-vocabulary spoken document retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 20–27.

Wechsler, M. and P. Schauble (1995). Indexing methods for a speech retrieval system. In *Proceedings of the MIRO Workshop*, Glasgow, Scotland, U.K.

Winston, P. (1992). *Artificial Intelligence* (Third ed.). Reading, MA: Addison-Wesley.

Witbrock, M. J. and A. G. Hauptmann (1997). Speech recognition and information retrieval: Experiments in retrieving spoken documents. In *Proc. DARPA Speech Recognition Workshop '97*, Chantilly, VA, USA.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks 5*, 241–259.

Wright, J. H., M. J. Carey, and E. S. Parris (1995). Improved topic spotting through statistical modelling of keyword dependencies. In *Proc. ICASSP '95*, Detroit, MI, pp. 313–316.

Wright, J. H., M. J. Carey, and E. S. Parris (1996). Statistical models for topic identification using phoneme substrings. In *Proc. ICASSP '96*, Atlanta, GA, pp. 307–310.

Young, S., J. Odell, D. Ollason, V. Valtchev, and P. Woodland (1997). *The HTK Book*. Cambridge, UK: Cambridge University.

Zhai, C., X. Tong, N. Milic-Frayling, and D. Evans (1996). OCR correction and query expansion for retrieval on OCR data – CLARIT TREC-5 confusion track report. In *Proceedings of Fifth Text REtrieval Conference (TREC-5)*, Gaithersburg, MD, USA. NIST-SP 500-238.