ABSTRACT

Title of dissertation: Combining Evidence from Unconstrained Spoken Term Frequency Estimation for Improved Speech Retrieval

J. Scott Olsson, Doctor of Philosophy, 2008

Dissertation directed by: Associate Professor Douglas W. Oard
College of Information Studies

This dissertation considers the problem of information retrieval in speech. Today's speech retrieval systems generally use a large vocabulary continuous speech recognition system to first hypothesize the words which were spoken. Because these systems have a predefined lexicon, words which fall outside of the lexicon can significantly reduce search quality—as measured by Mean Average Precision (MAP). This is particularly important because these Out-Of-Vocabulary (OOV) words are often rare and therefore good discriminators for topically relevant speech segments.

The focus of this dissertation is on handling these out-of-vocabulary query words. The approach is to combine results from a word-based speech retrieval system with those from vocabulary-independent ranked utterance retrieval. The goal of ranked utterance retrieval is to rank speech utterances by the system's confidence that they contain a particular spoken word, which is accomplished by ranking the utterances by the estimated frequency of the word in the utterance. Several new approaches for estimating this frequency are considered, which are motivated by the disparity between reference and errorfully hypothesized phoneme sequences. The first method learns alternate pronunciations or degradations from actual recognition hypotheses and incorporates these variants into a new generative estimator for term frequency.

A second method learns transformations of several easily computed features in a discriminative model for the same task. Both methods significantly improved ranked utterance retrieval in an experimental validation on new speech.

The best of these ranked utterance retrieval methods is then combined with a word-based speech retrieval system. The combination approach uses a normalization learned in an additive model, which maps the retrieval status values from each system into estimated probabilities of relevance that are easily combined. Using this combination, much of the MAP lost because of OOV words is recovered. Evaluated on a collection of spontaneous, conversational speech, the system recovers 57.5% of the MAP lost on short (title-only) queries and 41.3% on longer (title plus description) queries.

COMBINING EVIDENCE FROM UNCONSTRAINED SPOKEN TERM FREQUENCY ESTIMATION FOR IMPROVED SPEECH RETRIEVAL

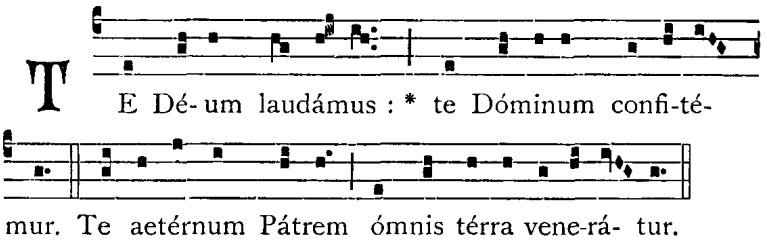
by

James Scott Olsson

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

Advisory Committee:
Associate Professor Douglas W. Oard, Chair/Advisor
Professor Eric Slud
Professor Mary Harper
Dr. Owen Kimball
Associate Professor Philip Resnik

DEDICATION



&

Jacquellae,

Quae quam dulcissime fecit nunquam
expectationem huius laureae
(quamquam saepe laborans aberam)
expectationem felicis vitae.

ACKNOWLEDGEMENTS

There are many people I'd like to thank for making this research and my education possible.

First, Dr. Douglas W. Oard has been an excellent adviser throughout my graduate studies. He has significantly shaped the way I size-up and attack research problems; he has never shied from helping me professionally or academically. For these things, I am exceedingly grateful.

I had the opportunity to teach several introductory courses in mathematics and statistics as a teaching assistant through the Applied Mathematics and Scientific Computation program at UMD. I am thankful for both the accompanying financial support and the many opportunities I enjoyed to both teach and learn with my students.

I was privileged to spend summer internships at BBN Technologies and with the Department of Defense. Both of these experiences helped to shape my research interests, and they will continue to do so. After my summer at BBN, they generously continued to support my research through critical software and computing resources. Without that aid, I could not have conducted this study.

My first work in information retrieval was supported by the National Science Foundation's MALACH project under Dr. Oard (NSF IIS award 0122466). I am grateful for that support. I am also grateful to the USC Shoah Foundation Institute for Visual History and Education, who permitted my use of their invaluable collection for this research.

In the final years of my studies I held a graduate fellowship with the Human Language Technology Center of Excellence at Johns Hopkins University. I am thankful for their financial aid, but more importantly for the many opportunities I've had with them to present and refine this work.

I am thankful to each of my committee members: Dr. Eric Slud, Dr. Owen Kimball, Dr. Mary Harper, Dr. Philip Resnik and Dr. Oard. Their feedback has improved both the execution and the presentation of this work dramatically.

Thanks are long overdue to my parents. From my earliest years, they have never hesitated in their love or support. They always told me, "You could do anything you want," and they always meant it.

Last, but far from least, thank you Jackie. You've given me a happy home and three beautiful kids: Kephas, Felicity, and Philomena. I might have been able to do this work if I had never met you, but it wouldn't have been worth it.

Table of Contents

Bibliography 104

List of Tables

List of Figures

List of Abbreviations

ADEC	speaker Adapted DECoding
CLEF	Cross-Language Evaluation Forum
CMNT	Combining by Monotonic Normalizing Transformations
DA	Domain-Adapted. Signifies that the underlying lexicon was extended to include words for the new topic domain
DTFE	Discriminative Term Frequency Estimation
EER	Equal Error Rate
EM	Expectation-Maximization
FRM	Fraction of Recovered Mean average precision
GTFE	Generative Term Frequency Estimation
GTFE-QD	Generative Term Frequency Estimation with Query Degradations
HMM	Hidden-Markov Model
IR	Information Retrieval
IV	In-Vocabulary
LVCSR	Large Vocabulary Continuous Speech Recognition
MALACH	Multilingual Access to Large spoken ArCHives
MAP	Mean Average Precision
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
NIST	National Institute of Standards and Technology
OOD	Out-Of-Domain. Signifies that the underlying lexicon was not extended to include words for the new topic domain
OOV	Out-Of-Vocabulary
p.m.f.	Probability Mass Function
PER	Phone Error Rate
PBQD	Phrase-Based Query Degradation
RMS	Root Mean Square
RSV	Retrieval Status Value. The score used by an IR system to rank documents
SR	Speech Retrieval
STD	Spoken Term Detection
TREC	Text REtrieval Conference. An annual, NIST sponsored, conference on information retrieval from large text collections
UDEC	Un-adapted DECoding
WER	Word Error Rate

Chapter 1

Introduction

Our goal is to help people find useful information in speech. Everyday we search the Web, looking for relevant information amongst billions of written documents. But the vastly larger set of information we produce—the information we speak—remains mostly unsearchable.

Only a few decades ago, written text was likewise unsearchable: a useful document could only be found if it had previously been indexed or you had extensive time to search for it. The early Web adopted this framework, as seen in pages of link collections or the early category-based Yahoo. But it was not until the advent of ranked Web document retrieval that the Web, as it is today, could be so extensively utilized.

Information Retrieval (IR)[1] is the task of satisfying a user's information need, expressed as a query, by one or more topically relevant items (e.g., documents). When improving a system, it is always necessary to know what "better" means, and so it was with text-based information retrieval. That is, in addition to a concrete specification of the problem (e.g., notions of *topic* and *relevance*), we must also systematize our evaluation. In this way, improvements to IR were facilitated perhaps most extensively by the National Institute for Science and Technology's yearly *Text REtrieval Conference* (TREC). Insofar as speech retrieval is simply information retrieval from speech, we adopt the TREC framework.

Following TREC convention, retrieval systems are commonly evaluated using *Mean Average Precision* (MAP). Given a ranked list of "documents" being searched through, we define the *precision* at position i in the list as the proportion of the top

[1]Note, a table of abbreviations is included in the front matter of this dissertation on page viii.

i documents which are relevant to the corresponding query. Average Precision (AP) is the average of the precision values computed for each position containing a relevant document. To assess the effectiveness of a system across multiple queries, Mean Average Precision is defined as the arithmetic mean of per-query average precision,

$$\text{MAP} = \frac{1}{n}\sum_{n} \text{AP}_n.$$

Of course, it was also TREC that once famously declared Speech Retrieval (SR) to be a "solved problem" [16]. And indeed, for certain tasks, such as retrieving professionally read broadcast news, this is arguably true. Using state-of-the-art Large-Vocabulary Continuous Speech Recognition (LVCSR) systems with carefully constructed recognition dictionaries, one can do *about* as well searching this speech as one can searching its manual transcription. But most speech is not professionally read in recording studios. Speech recognizers are sensitive to variations in training and testing conditions. Speech recognizers make errors with the words they do know, and they generally cannot anticipate the words they do not. These are only a few of the reasons why, despite innumerable potential applications, speech retrieval has not widely been commercially used today. When speech is emotional, spontaneous or conversational, or when it is captured from different and diverse media, the utility of speech retrieval systems is, at best, quite far from assured. In the wild, the problem is far from solved.

The foundational problems of SR are essentially common to all areas of IR. For example, how can we help users choose amongst the items returned by a search system? And after users have chosen, how can they use them? We leave aside these particular issues, although we note that they present significant challenges for retrieval in speech.

As we must in all IR problems, we must decide what task we will evaluate. When

searching text, a standard task is to put the relevant *documents* near the top in a ranked list of documents. In speech, however, browsing complete recordings for the few useful passages tends to be difficult and tedious. Moreover, human speech is generally not structured as documents commonly are (e.g., in spontaneous conversation we freely meander from one topic to the next). Primarily to facilitate evaluation then, our focus is the task of retrieving topically relevant speech *segments*. We define segments as topically coherent passages of the speech which were, in our collection, manually specified by professional indexers. Throughout this work, we prefer the term "segment" to "document" when referring to passages of continuous speech. We particularly avoid the common, if infelicitous, designation "spoken document retrieval," preferring instead simply "speech retrieval."

We use speech, segments, topics, and human-assigned ground truth relevance judgments created in the MALACH project (Multilingual Access to Large spoken ArCHives) [36]. The speech is a collection of interviews with survivors and witnesses of the Holocaust. It is dominated by heavily accented, emotional, elderly and spontaneous conversational speech and thus provides both a challenging collection for research and an important educational resource for future generations.[2] The specific collection of interviews, segments, topics and relevance annotations have previously been used by the Cross Language Evaluation Forum's Cross-Language Speech Retrieval (CLEF CL-SR) track [41] We introduce additional specifics about the CLEF CL-SR collection as necessary.

There are three general IR problems to which we[3] make particular contributions in the context of retrieving speech. First, we must decide how to represent our speech segments when we have only noisy hypotheses of the speech content. Similar problems are shared by retrieval from optical character recognition output, cross-language IR

[2]For examples of the types of interviews used in this work, as well as additional information about how and why they were collected, see [55].

[3]The "we" is stylistic. The author alone is responsible for the contents of this dissertation.

and, in a sense, even IR from "clean" text. In the latter, we may know the words, but they may only be a noisy representation of the underlying semantic content. Second, we must decide how to represent queries. This may include determining pronunciations for query words or selecting subsystems for different query word types. Lastly, we must decide how to combine speech hypotheses and query representations to best rank the segments.

1.1 Representing Speech Segments

Presently, the best way we know to model speech is through the Hidden-Markov Model based LVCSR approach [44], which we adopt in part. While automatic speech recognition has primarily been designed to minimize errors in *transcription* (i.e., a single best hypothesis of words spoken), there is little reason to ignore other, less probable utterance hypotheses in the context of speech *retrieval*. Just as cross-language IR can be improved by using probabilistic term frequency translation instead of a one-best hypothesis (i.e., by more fully capturing the uncertainty in translation), spoken-document IR can be improved by searching a *lattice* of utterance hypotheses rather than only the one deemed most probable. A lattice is a directed acyclic graph that is used to compactly represent the search space for a speech recognition system. Each node represents a point in time and arcs between nodes indicates a word occurs between the connected nodes' times. Arcs are weighted by the probability of the word occurring, so that the so-called "one-best" path through the lattice (what a system might return as a transcription) is the path through the lattice having highest probability. Figure 1.1 shows a simple mock example, which encodes multiple hypotheses and their probabilities.

These hypothesis probabilities are composed, chiefly, from the LVCSR system's acoustic and language model scores for a word in a particular location. Simply stated, an acoustic model tells us how likely it is that a portion of speech audio was generated

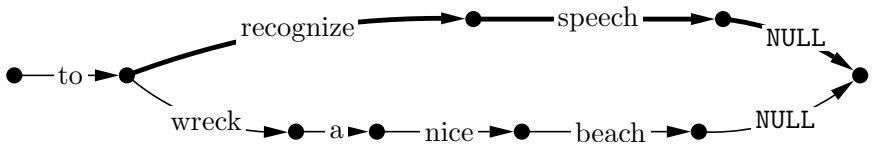


Figure 1.1: A mock example lattice. Arc thickness is proportional to model probability.

from a particular word, while a language model tells us how likely it is that a word occurs following one or more previous words. Finding the most probable sequence of words $\hat{W}$ given an observed audio segment O then amounts to solving

$$\hat{W} = \arg\max_W \frac{P(O|W)P(W)}{P(O)} = \arg\max_W P(O|W)P(W), \tag{1.1}$$

where $P(O|W)$ and $P(W)$ denote the acoustic and language models, respectively, and the denominator has been dropped because we need only to rank the hypotheses. This process of finding one or more best hypotheses is referred to as *decoding*, which can be efficiently computed using the Viterbi algorithm [44].

Automatic speech recognition is an exceedingly complex art, the details of which we do not here discuss at length. For the purposes of this research, LVCSR is essentially a black box which, given a dictionary of pronounced words (or subwords) and some audio, outputs a lattice of speech hypotheses. Nevertheless, it is necessary to understand a few details of the LVCSR system. We turn therefore to briefly outline some of its components, insofar as is necessary to understand our experimental design and goals.

1.1.1 The Decoding Dictionary

A key problem with representing speech segments with word-based LVCSR is that LVCSR systems generally have fixed recognition *dictionaries*. These decoding dictionaries specify, for each word that may be hypothesized, the word's pronunciation and orthography. However, fixed dictionaries can not anticipate every word which may

be used to express a query. Indeed, since the most common words are known to be the least informative for retrieval, the standard transcription evaluation measures—which roughly measure the proportion of tokens that are correctly recognized—have largely driven LVCSR research to excel at precisely the words that IR users don't care about. The upshot of this is that you cannot easily find a segment about the notorious Adolf Eichmann with the query "Eichmann" unless, quite fortuitously or through anticipation, your LVCSR system knows the word "Eichmann."

When an LVCSR system does not know about a term, we say that the term is Out-Of-Vocabulary (OOV). In a restricted sense, words have most often been considered OOV whenever they are not within the LVCSR dictionary. We use the term OOV more forcefully, requiring that the LVCSR system neither contains the word in its dictionary nor has ever seen the word in acoustic or language model training. That is, an OOV term is a word that was both not anticipated and could not have reasonably been anticipated when constructing the LVCSR system.

In order to maximize transcription accuracy without unduly increasing search complexity, LVCSR systems have over the years included increasingly larger decoding dictionaries. The words in these dictionaries must be chosen in view of a target domain to keep the OOV rate low, but of course not every potential word may be anticipated. In particular, when a new topic domain is encountered, the decoding dictionary may be quite poorly matched to the target, making it very difficult for users to find segments relevant to their information need.

We use this scenario of domain switching (i.e., an LVCSR system is developed for one *topic* domain but then used on another), to create a plausible distribution of OOV terms for our experiments. We will investigate IR systems built with LVCSR, both when the decoding dictionary has not been adapted for the topic and when it has. We refer to a system built using a domain-adapted dictionary as being *Domain-Adapted* (**DA**). When the dictionary has not been extended for the new topic domain,

we refer to the system as being *Out-Of-Domain* (**OOD**).[4] While we expect an SR system built on DA LVCSR to perform best, and thus consider its performance an upper bound on retrieval utility, it will for the foreseeable future remain impossible to build one LVCSR system having good lexical coverage of all possible topic domains. We emphasize that we are considering a shift in topic domain, and that other shifts in a collection's characteristics (e.g., dialect, age, channel, or signal conditions) may also present serious difficulties that are beyond the scope of this dissertation.

Rather than expanding a dictionary to include words which may occur in a new domain, we may alternatively consider recognition units that can combine to form arbitrary new words. Since a *word* in an LVCSR system is really only a sequence of phonemes[5] and some orthographic representation, we can instead recognize a closed set of subword units (e.g., the phonemes themselves) that will cover the complete set of possible words in the language. This has the advantage of potentially recognizing every word in the language. At the same time, it is well known that, owing to the paucity of information (e.g., context) which such a recognizer can exploit, subword recognition accuracy tends to be poor. In Chapter 2 we present two subword-level dictionaries that we use for our vocabulary-independent speech segment representations. In Chapter 3, we explore how to handle recognition errors when detecting occurrences of OOV terms. In Chapter 4, we consider how to combine these

[4]Our terminology is similar to other work in adapting language processing applications to new test domains. For example, in [11] data are modeled as mixtures of "truly in-domain", "general-domain" and "truly out-of-domain" distributions. Both our OOD and DA dictionaries contain "general-domain" words (e.g., function words). Both also contain "out-of-domain" words, because the DA dictionary is a superset of the OOD dictionary which presumably contains some out-of-domain words. The DA dictionary is extended with "in-domain" words, seen in in-domain transcripts. Nevertheless, we denote our DA dictionary as "domain-adapted" rather than "in-domain" because, in our task, having extended the dictionary with in-domain data does not guarantee our dictionary will cover the test data. That is, even our extended dictionary is still "out-of-domain," although it has been adapted.

[5]A *phoneme* is the smallest possible unit of a particular language's speech that can distinguish meaning. This differs from particular recognizable sounds, or *phones*, in that they are an abstract categorization of many phones into one cognitive unit. However, for various reasons, the speech recognition literature has been inconsistent in its application of *phone* and *phoneme*, using them almost interchangeably.

vocabulary-independent term detection results with output from a conventional SR system with an OOD word dictionary. This combination will allow us to significantly improve our retrieval effectiveness when the topic domain is not well covered by the LVCSR system's recognition dictionary.

1.1.2 Language Models

We can often make a good guess about the next word to be spoken if we know some of the preceding words. This intuition is exploited in automatic speech recognition by incorporating a *language model*, as we noted in Equation 1.1. A language model $P(w_1, \ldots, w_m)$ estimates the probability of observing a string of words, $w_1, \ldots, w_m$. Most commonly, and in this work, we model this as an n^{th} order Markov process. That is, the probability of observing word w_i may be computed solely from the preceding $n-1$ words.

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i | w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$

Generally, these parameters are chosen by maximum likelihood estimation (MLE), so that

$$P(w_i | w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{count(w_{i-(n-1)}, \ldots, w_i)}{count(w_{i-(n-1)}, \ldots, w_{i-1})}.$$

Because words often appear in new contexts it is unwise to assign occurrences zero probability mass, and so these probability estimates must be smoothed—a discussion we defer for now.[6] This method of language modeling is referred to as an n-gram model and, by convention, second and third order models are referred to as *bigram* and *trigram* models, respectively.

When discussing language models for words, unless context makes it clear, we refer

[6]Different smoothing techniques are most appropriate for different language modeling tasks. For example, we use Kneser-Ney smoothing [8] for words in Section 2.2.3 and Witten-Bell smoothing [61] in Sections 2.2.3 & 3.2 for phonemes.

to them as *word-level* or simply *word* language models. We will use the same modeling approach to predict the occurrence of phonemes or multiple-phoneme units, and in that case we follow the same naming convention (e.g., denoting them as *phoneme* or *phoneme-level* language models).

For each type of language modeling, we require example corpora from which to estimate our language model parameters. In every case, as noted above, we are diligent to ensure that OOV terms are never seen in language model training. This is particularly important for subword-level transcripts. That is, a subword-level transcript must not contain a subword-level transcription of an OOV term, because that would unfairly bias the system to predict that sequence. We produce our phoneme level transcriptions by simply using the appropriate dictionary to rewrite our word-level transcripts with phonemes. We also consider longer recognition units, called phone multigrams, and a segmentation of phoneme transcripts into these phoneme multigrams in Chapter 2.

1.1.3 Acoustic Models

An acoustic model $P(O|W)$ gives the probability of an observed audio sequence O having been generated by a particular phoneme (or, by composition, one or more words or multigrams) W. In this work, and generally, acoustic models are constructed using Hidden-Markov Models (HMMs). In this framework, the observations are features extracted from the audio signal (e.g., Mel-Cepstra coefficients) and the hidden state sequence is comprised of the phonemes (or perhaps, portions of the phonemes) which gave rise to the observations. The likelihood of each observation is computed using mixtures of diagonal covariance Gaussians.

HMMs are attractive for several reasons. First, they provide a plausible model for human speech (under the assumption that speech is, at a short time scale, an approximately stationary process). Secondly, and perhaps more importantly, effi-

cient methods for estimating their parameters are well known—namely by MLE via Expectation-Maximization [44, 13].

A comprehensive discussion of acoustic modeling is beyond our present scope, although to the basic method outlined above we must add one additional complexity. Because phonemes vary with context (e.g., because of coarticulation), acoustic models in state-of-the-art LVCSR systems are generally context dependent. That is, phonemes with different left and right contexts correspond to different HMM states. Consequently, if phoneme models are *trained* on particular words and their set of contexts, but then those words are removed from the recognition or *decoding* dictionary, we can not fairly refer to those words as being OOV. Certainly, if these acoustic models were then used for subword (e.g., phoneme level) recognition, we might be misled to believe the system was particularly good at hypothesizing phonemes in new contexts (when, really, the contexts were not new at all). We avoid this difficulty by training new acoustic models for each of our six recognition systems. Specifically, we train LVCSR systems using words, phonemes, and phoneme multigrams, with and without extending an OOD dictionary with the words from a DA dictionary.

1.1.4 Indexing

Finally, after producing a lattice representation—which may contain word- or subword-level units—we must index the data to reduce the time and space search costs. In Chapter 2 we construct an inverted index of speech segments, using the expected counts of words or subword sequences in the lattices.

1.2 Representing Queries

To match a query to the utterances in our index, we of course need a representation of the query. Query words which are in-vocabulary present no specially difficulty and

can be modeled using a bag of words (i.e., an unordered collection of the words). When words are OOV, we first represent them using a sequence of phonemes specifying the word's pronunciation. Ideally, locating an OOV word in the index would be as simple as retrieving segments that contain this lexical pronunciation, but of course this is not the case due to errors in recognition. We can account for these errors both when indexing and when processing a query. Our lattice based representations of speech segments address these errors on the recognition and indexing side of the search problem. However, as these pre-search models grow in complexity (e.g., as lattice sizes grow) the corresponding costs to index and search them grows considerably. We also know that mismatches between query and recognized phoneme sequences may occur *even if the recognition is perfect*. For example, conversational and lexical pronunciations frequently differ. These facts suggest that some search-time uncertainty modeling of an OOV word's pronunciation may improve retrieval accuracy.

Building on the subword-level indices produced in Chapter 2, we consider in Chapter 3 several new methods for detecting the occurrence of OOV query terms. In particular, we consider a discriminative model for term frequency estimation which, with respect to a strong baseline, is able to improve term detection utility under noisy subword speech representations. Secondly, we consider a generally applicable technique for term detection which considers alternate phonemic sequences for the query word, which we call query *degradations*. Similar approaches have been taken in other IR tasks, e.g., OCR document retrieval [9, 10]. We use these hypothesized degradations to transform a query's lexical phoneme sequence into the sequences we expect were actually obtained via recognition. That is, we consider a distribution of degradations for each query term, and rank the segments by their probability of containing the term. This probability is the weighted average of the occurrence scores for each degradation; that is, it is the occurrence probability in expectation with respect to

the distribution of degradations. We explore two methods for producing these degradations and, in particular, consider degradations obtained through a state-of-the-art phrase-based statistical machine translation system.

With the popularization of search, users have become both comfortable with and adept at producing short queries to find information. In part, Web search has become commonplace precisely because it is easy (e.g., users need not master a specific query language). That is, users have been able to conceptualize a mental model for the short-query search *process*. Certainly, we might expect that Web searchers would be disrupted if informed that their query was invalid simply because it contained a rare term or, worse, a common one. It may therefore be useful for an SR system to handle both OOV terms and LVCSR words in combination and, when desired, for any word-specific handling to be hidden from the user. Fortunately, since the system always knows which words are in its LVCSR dictionary, hiding this complexity from the user presents no special difficulties. In Chapter 4 we present one such system, which allows users to naturally express SR queries using both in- and out-of-vocabulary terms. Using our combination scheme, we are able to recover much of the MAP that would otherwise be lost because of terms which could not be anticipated.

1.3 Contributions

In this dissertation we make the following contributions to the field of speech retrieval:

- We introduce a new vocabulary-independent indexing scheme which allows us to efficiently store and search utterances. In this approach, we first convert multigram or word LVCSR lattices into phoneme lattices. Then, we index short sequences of phonemes by the number of times they occur, in expectation, in the phoneme lattices.

- We show that, by using multigrams or words rather than individual phonemes,

we can drastically reduce the size of our index while improving recognition accuracy (phoneme error rate).

- We propose and evaluate a new approach to spoken term detection using hypothesized alternate pronunciations or *degradations* of the query word's phoneme sequence.

- We construct a system to hypothesize pronunciation degradations using a phrase-based statistical machine translation system. Using this approach, we significantly improve our ranked utterance retrieval MAP with respect to a baseline degradation approach.

- We propose a discriminative model for ranked utterance retrieval. In an evaluation on new data, it performed significantly better than a strong generative baseline.

- We combine a speech retrieval system built on OOD LVCSR and a vocabulary-independent ranked utterance retrieval system to recover much of the MAP lost due to OOV query words. This new combination approach learns normalizing transformations of the non-combined retrieval scores to significantly improve upon several baseline combination methods.

1.4 Organization of Dissertation

We begin in Chapter 2 by presenting our method for obtaining speech segment representations and indices. We conclude with an intrinsic[7] experimental validation of each of our LVCSR systems, measuring both their accuracy when hypothesizing phoneme

[7]We define an *intrinsic* measure to be a measure of the performance of one component of a language processing system on its particular subtask (e.g., phoneme error rate for a phoneme recognizer). In contrast, an *extrinsic* measure measures the performance of a method on a larger, often downstream task. For example, MAP in a SR task might be an extrinsic measure of the quality of the speech recognition system.

and word sequences on new data and the space costs required to index their lattices using expected recognition unit counts.

In Chapter 3, we use our subword-level LVCSR systems to present several new methods for predicting occurrences in speech of OOV query words. We present an evaluation of these techniques, which is both an extrinsic measure of these LVCSR systems and an intrinsic measure of their utility for detecting spoken terms.

In Chapter 4, we again build on the previous chapters, developing a simple method for combining word-level SR results and vocabulary-independent term detection results, for complete *ad hoc*[8] speech retrieval. Using a standard measure of an IR system's utility, MAP, we show that, by combining our vocabulary independent and OOD LVCSR systems' results, we are able to recover much of the loss incurred from our LVCSR system not having been adapted to the new topic domain.

Finally, we conclude in Chapter 5 with some analysis and discussion, a recap of the limitations of this dissertation, and directions for future work.

Please note that page viii in the front matter contains, as an aid to the reader, a table of abbreviations which are used throughout this dissertation.

[8]The phrase *ad hoc* is used by the IR community to emphasize that the task is to satisfy an information need given a previously unseen topic specified as a query. This is in contrast to tasks such as "known item retrieval".

Chapter 2

Indexing Speech for Retrieval

2.1 Introduction

Before we can efficiently search audio, we must first index it. In this chapter, we discuss how we construct these indices, using both word- and subword-level recognition dictionaries. We consider, in turn, many components of this process, from segmenting complete speech recordings into manageable *utterances* to decoding these utterances into lattices and extracting from them indexable features. Of course, before we can recognize speech audio, we must first train our LVCSR systems, which we consider briefly.

In this research, we essentially consider our LVCSR systems to be black boxes, into which we input our audio and a decoding dictionary and from which we obtain, as output, a lattice representation of the speech utterances, where lattice arcs contain elements from the designated dictionary. Our principle interest therefore, in this chapter, is to consider precisely what units should be included in our dictionary. We desire units that are both flexible enough to allow the recognition of arbitrary phoneme sequences (so that we may detect OOV terms), while achieving the best possible recognition accuracy. As an intrinsic evaluation of these systems, we consider standard measures of phoneme and word recognition accuracy in Section 2.3. We defer until Chapter 3 an extrinsic evaluation of the systems, in which we utilize them for detecting the presence of OOV terms.

2.1.1 Simulating OOV Terms by Shifting Topic Domain

Since we are interested in handling realistic OOV terms in our speech retrieval queries, we must first establish our set of in-vocabulary terms and the training corpora on which to build our LVCSR systems. While some previous work has investigated handling OOV terms by artificially shrinking the dictionary to simulate them, we chose not to take this approach. First, it is not clear how to fairly remove terms. For example, it makes little sense to remove a term which would normally have been included in a reasonable dictionary (and removing such a term would likely inflate the utility of the OOV handling system). On the other hand, if the dictionary is unreasonably *suitable* for the domain, then leaving other terms *in the decoding dictionary* could distort our results: if a decoding dictionary unfairly contains a word (even if it is not a word of interest, that is, a query term), it might artificially inflate the utility of the recognition system (which would normally have to handle the higher OOV rate during decoding). Moreover, since our downstream task of interest is information retrieval, if we choose to remove terms from a dictionary, then we must also consider removing terms having the same stem or even terms which might be added to a query after blind relevance feedback.

To avoid these difficulties, we conceptualize the problem as that of handling the onset of new speech topics. That is, we suppose we have a system trained for similar types of speakers (e.g., similar distribution of age, gender, and accent), but who have not previously talked extensively about our topic—namely, experiences during and surrounding the Holocaust. Accordingly, we take as our OOD training and decoding dictionary, a large dictionary previously used for English newswire and conversational telephone speech transcription. For comparison, we then also consider a dictionary that has been adapted to our new topic domain; we refer to this as the DA dictionary.

This chapter is organized as follows. First, in Section 2.2 we introduce our phoneme and word-level dictionaries. We also explain how we construct our phoneme

multigram dictionary and give some examples of its use. Section 2.3 presents our evaluation results and analysis for each of the systems.

2.2 Representing Speech for Search

We first consider what set of recognition units, that is what dictionary, we use for decoding our speech. We use dictionary to mean a complete specification of the units which the LVCSR system may hypothesize. A single dictionary element is specified uniquely by its pronunciation (one or more phonemes) and, in the case of words, its orthography. It is not necessary to specify an orthographic representation of subwords, since we will only use them internally, for indexing and searching. We discuss later how we obtain pronunciations for new, OOV, words.

2.2.1 Phoneme and Word Dictionaries

Words and phonemes are natural candidates for members in a decoding dictionary, and we consider them both. Phonemes are attractive in that they provide a smallest possible dictionary able to distinguish all potential words in our language of interest, English. On the other hand, one could imagine a *complete* dictionary of words, by definition also able to distinguish every English word. However, this will remain impracticable for the foreseeable future—both in implementation and simply because we are constantly creating new words.

Our phoneme inventory includes 39 English phonemes from the ARPABET phoneme set plus silence and several noise phonemes (e.g., coughing, laughing, and breathing) which, taken together, comprise our phoneme dictionary. We do not add any additional phonemes for non-English words that may be present in the collection (these are modeled as being pronounced using a reasonably close English phone). Table 2.1 lists the complete phoneme set.

Phoneme	Example word	Pronunciation
AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
EY	ate	EY T
F	fee	F IY
G	green	G R IY N
HH	he	HH IY
IH	it	IH T
IY	eat	IY T
JH	gee	JH IY
K	key	K IY
L	lee	L IY
M	me	M IY
N	knee	N IY
OW	oat	OW T
OY	toy	T OY
P	pee	P IY
R	read	R IY D
S	sea	S IY
SH	she	SH IY
T	tea	T IY
TH	theta	TH EY T AH
UH	hood	HH UH D
UW	two	T UW
V	vee	V IY
W	we	W IY
Y	yield	Y IY L D
Z	zee	Z IY
ZH	seizure	S IY ZH ER

Table 2.1: The phoneme inventory. Example words are adapted from [54].

Our OOD dictionary contains about 50,000 words with manually specified pronunciations, and was previously utilized for conversational telephone speech transcription. To that OOD dictionary we added roughly 10,000 automatically pronounced words to cover our complete set of training transcripts (we elaborate in Section 2.2.3.1). We say that a dictionary *covers* a set of transcripts if at least every word in the transcript is within the dictionary. This gave us our DA word dictionary, containing 60,378 pronounced entries.

If our phoneme recognizer could be perfect, finding the occurrences of previously unseen terms would be a comparatively straightforward task: given a lexical pronunciation for a query term, we would need only to find its occurrences in the perfect recognition output. Unfortunately, in the case of continuous and spontaneous speech, the accuracy from these recognizers tends to be quite low. It is commonly argued that this is so because phoneme recognizers using low order (e.g., bigram or trigram) phoneme-level language models cannot consider the same amount of context as the same order word-level models and because longer recognition units (e.g., words) provide strong constraints on permissible phoneme sequences (i.e., the system only admits phoneme sequences that can be constructed from available pronunciations). One consequence is that phoneme recognizers can easily hypothesize phonotactically impossible phoneme sequences, which can not occur using word-level models.

On the other hand, phone-level recognition benefits from much weaker assumptions. If a word-level recognizer encounters an OOV term, it is constrained to hypothesize one or more incorrect terms (those most likely under its acoustic and language models). A phone level model, of course, is never forced to hypothesize an incorrect symbol, since all recognition units are within the system's vocabulary.[1] A consequence of this is that, while a word-level system is virtually guaranteed to have a lower *phoneme* error rate, it is not entirely clear that the phonemes which are correct

[1]This doesn't consider the problem of phonemes which may be present from non-English languages.

are the phonemes we care about—that is, the phonemes on OOV terms.

2.2.2 Constructing a Phoneme Multigram Dictionary

It appears then, that there may be a trade off between having too little knowledge (phoneme recognition) and knowing *too much* (word recognition). We consider therefore a middle ground dictionary, constructed using phoneme *multigrams*. Originally proposed by Deligne and Bimbot [12] to model variable length regularities in streams of symbols (e.g., words, graphemes, or phonemes), phoneme multigrams are short sequences of one or more phonemes. Precisely because they are trained to capture *regularities* in streams of symbols (e.g., phonemes), we hope our multigram-based LVCSR system will generalize to new sequences. Because we learn our set of multigrams from phoneme transcripts, the multigrams can capture the most common words (or word combinations), while also providing the flexibility to cover previously unseen phoneme sequences (OOV terms). We learn our set of multigrams using dictionary rendered[2] versions of our acoustic training transcripts and, of course, exclude any utterances not covered by our OOD dictionary.

One significant difference in this work, regarding the use of multigrams [35], is that we are interested in finding useful subword units *prior* to recognition. That is to say, our approach will build up a lattice of subword units from which we can extract expected counts of phoneme sequences to then be used in term frequency estimation and retrieval. This differs from [35] which extracted multigrams from a one-best phoneme transcript *after* recognition to use as indexing units for search.

Short sequences of phonemes (or *word-fragments*) have previously been used to generate phoneme lattices for vocabulary-independent indexing. Closely related is [52], in which a word-fragment dictionary was created by pruning a phoneme language

[2]Here, by "dictionary rendered", we mean only that we map the word level transcripts into phoneme transcripts through the dictionary. We do not force align the word level transcripts to find the phonemes that were most likely actually spoken.

model to remove redundant, high order n-grams. Their chief contribution was the proposal of a scalable indexing approach, using discriminative paths in the phoneme lattices, to produce a candidate set of utterances which could then be ranked in a second step. Our approach differs in the criterion for selecting our word-fragments (i.e., multigrams) and in that we retrieve utterances using the index in only one step.[3]

We now present a simplified derivation of the multigram model and how its parameters may be estimated. In the multigram model, a sequence of multigrams is emitted *independently* from a set of multigrams $\{z_i\}$. Each multigram is composed of one or more phonemes which will be observed. The only observable output of this process is the string of concatenated phonemes O. Our goal then is to find the underlying (hidden) segmentation S of O such that we recover the original multigrams. From independence, the data log likelihood is simply

$$\mathcal{L}(O, S|\{z_i\}) = \sum_{i=1}^{m} c(z_i|S) \log p(z_i), \tag{2.1}$$

where $c(z_i|S)$ denotes the number of occurrences of multigram z_i in segmentation S.

If we could observe the hidden segmentation S, we could trivially count the multigrams and thus produce maximum-likelihood estimates for the model parameters $p(z_i\}$. Since the segmentation is hidden from us, this suggests an Expectation-Maximization approach [12]. Let $\mathcal{L}^{(k)}$ be the data likelihood at the k^{th} iteration. Dempster et al. [13] showed that, if we define an auxiliary function

$$Q(k, k+1) = \sum_{S} \mathcal{L}^{(k)}(O, S) \log \mathcal{L}^{(k+1)}(O, S) \tag{2.2}$$

and can update our parameter estimates such that $Q(k, k+1) \geq Q(k, k)$, then we also increase the data likelihood: $\mathcal{L}^{(k+1)} \geq \mathcal{L}^{(k)}$, with equality only if our parameters

[3] We defer discussion of how our indices are used for ranking until Chapter 3.

are identical before and after the update. Combining Eqs. 2.1 & 2.2, we have

$$Q(k, k+1) = \sum_{i=1}^{m} \sum_{S} c(z_i|S) \log p^{(k+1)}(z_i) \mathcal{L}^{(k)},$$

which we may maximize w.r.t the parameters $\{p^{(k+1)}(z_i)\}$ by the method of Lagrange multipliers,[4] with the solution

$$p^{(k+1)}(z_i) = \frac{\sum_S c(z_i|S)\mathcal{L}^{(k)}}{\sum_S c(S)\mathcal{L}^{(k)}} \tag{2.3}$$

Therefore, to find maximum-likelihood estimates of the multigram parameters, we need only to iteratively apply the update formula, Equation 2.3, and to recompute the data likelihood with each successive set of parameter estimates.[5]

Looking at Equation 2.3, we see we require the expected count of each multigram with respect to the distribution of possible segmentations. This suggests a simple and practical implementation using an off-the-shelf implementation of the forward-backward algorithm [53]. The idea is to construct a lattice which traverses all possible segmentations of each utterance, where the arc weights for each multigram z_i are simply the current parameter estimates $p^{(k)}(z_i)$. The complete path probability of that traversal (or segmentation) is then the likelihood $\mathcal{L}^{(k)}(O, S)$. Figure 2.1 shows an example segmentation lattice. Now, having obtained the expected multigram counts from forward-backward, we need only to normalize them by the sum of all expected counts to obtain our updated parameter estimates $p^{(k+1)}(z_i)$.

We defined Z^k to be the set of multigrams $\{z_i\}$ after the k^{th} EM iteration. At convergence, Z will be our multigram dictionary. Since we prefer a small dictionary,

[4]Writing $p^{(k+1)}(z_i)$ as p_i, the method gives $\frac{\partial}{\partial p_i}\left[\sum_{i=1}^{m}\sum_S c(z_i|S)\log p_i \mathcal{L}^{(k)} + \lambda(\sum_i p_i - 1)\right] = 0$ which, differentiating, gives us $-\lambda p_i = \sum_S c(z_i|S)\mathcal{L}^{(k)}$.

But $\sum_{i=1}^{m} p_i = 1$, so that $\sum_{i=1}^{m} -\lambda p_i = -\lambda = \sum_{i=1}^{m}\sum_S c(z_i|S)\mathcal{L}^{(k)}$, and $\sum_{i=1}^{m} c(z_i|S) = c(S)$, which gives us Equation 2.3.

[5]Note, there is no guarantee that our estimates globally maximize the likelihood function. We don't take any precautions to avoid this possibility. We may address this in future work, e.g., by restarting with random estimates.

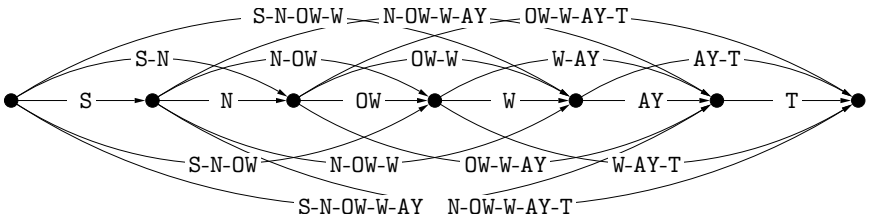


Figure 2.1: A segmentation lattice used for the forward-backward implementation of multigram estimation. Shown is a segmentation of the phoneme sequence for the words "snow white".

we discard all multigrams with expected count less than one in a given iteration—*unless* the multigram is a single phoneme. We preserve all length one multigrams, to guarantee that we can segment any new phoneme sequence.

We segment a new phoneme sequence by finding the segmentation $\hat{S}$ which maximizes the likelihood, $\hat{S} = \arg\max_S \mathcal{L}(O, S|\{z_i\})$. This is easily computed, also with standard tools for Viterbi decoding [53], as the one-best traversal of our segmentation lattice. Both the parameter estimation and decoding steps are trivially parallelizable, a useful property in that our experiments require segmentations for approximately 200,000 utterances.

Algorithm 1 outlines our implementation. It differs from previous work in that, for each iteration, we segment the entire collection of phoneme transcripts (rather than only a dictionary of pronounced words). This will encourage subwords from more frequent terms to occur more frequently, and also allows common cross-word phoneme sequences to be included as multigrams.

For starting estimates of our parameters $p^{(0)}(z_i)$, we use the relative frequency of each phoneme n-gram, $n \leq 5$, in the collection. For the DA and OOD training transcripts,[6] we have initial multigram dictionary sizes of $|Z^0| = 730{,}499$ and 595,596 respectively. Note, the OOD transcripts have many fewer potential multigrams because they contain many fewer unique terms. We set $\tau = 1$ and following common

[6]We ran our multigram segmentation algorithm on both the complete set of DA transcripts which are available for acoustic model training and on the reduced set of utterances which contain no words outside of the OOD dictionary. For more information about the data used, see Section 2.2.3.1.

Algorithm 1 The multigram training algorithm.

Get initial parameter estimates, Z^0.
$k \leftarrow 0$
repeat
  for each training utterance O_i **do**
    Construct a lattice L_i traversing O_i, with arcs for every entry in Z^k.
  end for
  for each lattice L_i **do**
    Run forward-backward on L_i and extract the expected count of each multigram.
  end for
  Aggregate the counts for all multigrams across the lattices.
  if any multigram z_m has aggregated expected count $C(z_m) < \tau$ and z_m is not a unigram **then**
    Remove z_m from dictionary.
  end if
  Produce a new dictionary Z^{k+1}, by MLE from these expected multigram counts.
  $k \leftarrow k + 1$
until stopping criterion is met.
Return the multigram dictionary Z_k.

practice, terminated after a fixed number of iterations, $k = 30$. This gave us a final dictionary size of 21,153 or 16,409 subwords, for the DA and OOD systems respectively. Figure 2.2 depicts the size of the multigram dictionary after each iteration on the MALACH training data. Note, while the OOD system has many fewer multigrams, it is still able to span all possible phoneme sequences—since we never remove phoneme unigrams from the multigram dictionary.

Algorithm 2 shows how we apply the subword model to segment a phoneme sequence into multigrams. Again, we construct a lattice traversing the utterance's phonemes with arcs for each subword in the dictionary, but now run Viterbi decoding to select the one most probable path.

Table 2.2 shows some example transcripts, segmented at the word, phone, and phoneme multigram level from the MALACH collection, using Algorithm 2. We observe that the most common words often correspond to a multigram, as do very common pairs of words. Examples include (`RIGHT`, `R-AY-T`), (`TO DO`, `T-AX-D-UW`).

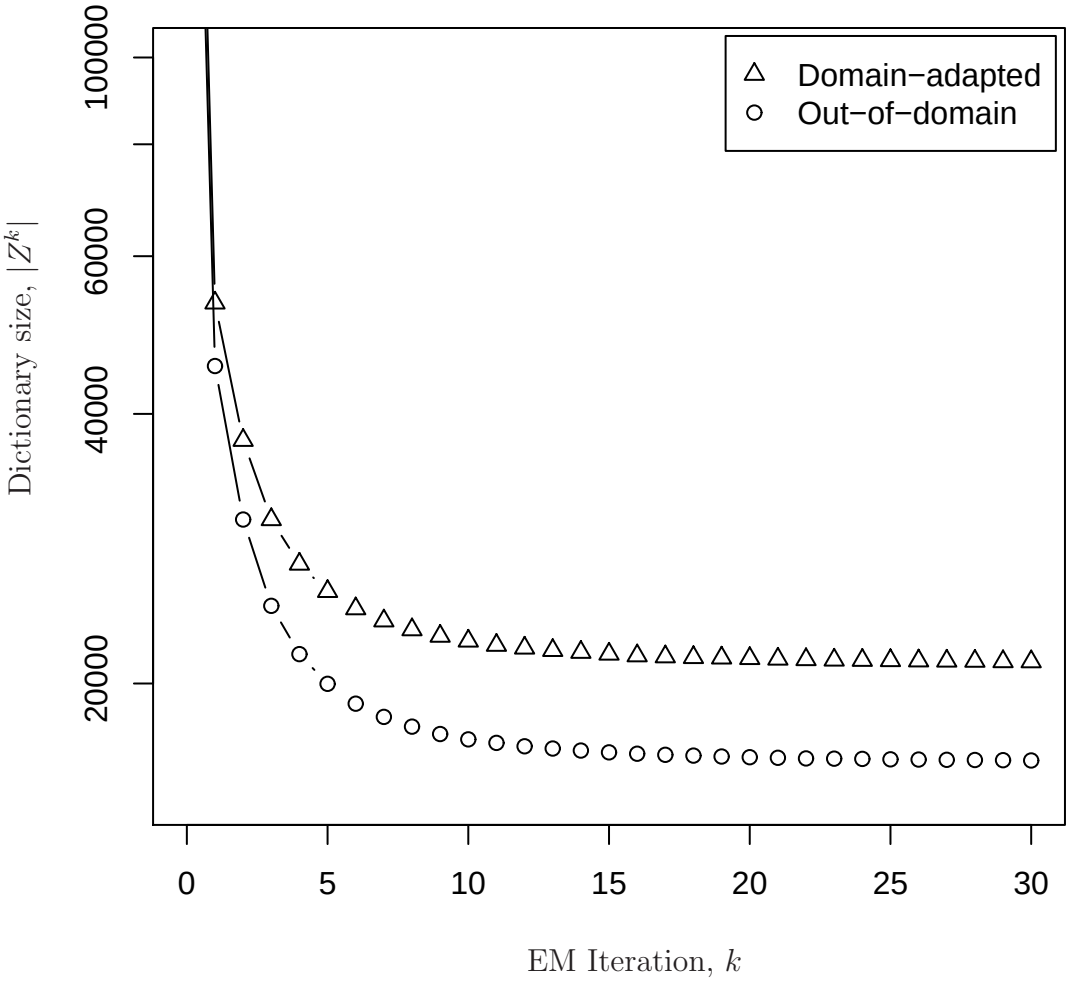


Figure 2.2: The size of the multigram dictionary after each iteration of subword training, Algorithm 1, on the MALACH transcripts. Results are shown using both the OOD (◯) and DA (△) resources.

Common subsequences within words tend to become their own multigram, so that, for example, `RUSSIANS` is segmented as the two multigrams `R-AH-SH-AX-N` and `Z`. The multigram `Z` is retained because it is often able to form the plural of words. Table 2.2 shows multigram segmentations using both the DA and OOD systems; we observe that the hypothesized segmentations rarely differ.

Algorithm 2 The multigram decoding algorithm.

for each utterance O_i **do**
  Construct a lattice L_i traversing O_i, with arcs for every entry in Z.
end for
for each lattice L_i **do**
  Extract one best path through L_i by Viterbi decoding.
end for

2.2.3 Automatic Speech Recognition

We were graciously permitted to use BBN Technology's speech recognition system *Byblos* [43, 33] for our speech recognition experiments.

The MALACH interviews were typically recorded with the interviewer and interviewee sitting near each other, each with separate microphones being recorded on different (left or right) stereo channels of the tape. There is generally strong cross-over (i.e., both speakers may be heard on either channel of the tape). Moreover, the transcripts are not marked as to which channel contains which speaker. Accordingly, because most of the talking is from interviewees, we selected the one channel of the audio having largest RMS amplitude for training and decoding; this follows previous work on the same collection [4]. All audio was down-sampled to 8kHz.

2.2.3.1 Training

Because our OOD dictionary does not cover the entire training audio transcripts, we used pronunciations for roughly 14k additional words from the CMU dictionary [54]

Type	Transcript
words	BECAUSE SHE DIDN'T HAVE TIME TO DO THE HOUSEWORK
phonemes	B IY K AO Z SH IY D IH D AX N T HH AE V T AY M T AX D UW DH AX HH AA UW S W ER K
DA multigrams	B-IY-K-AO-Z SH-IY-D-IH-D AX-N-T HH-AE-V T-AY-M T-AX-D-UW DH-AX HH-AA-UW-S W-ER-K
OOD multigrams	B-IY-K-AO-Z SH-IY-D-IH-D AX-N-T HH-AE-V T-AY-M T-AX-D-UW DH-AX HH-AA-UW-S W-ER-K
words	RIGHT SO EVERYBODY RAN AWAY
phonemes	R AY T S OW EH V R IY B AA D IY R AE N AX W EY
DA multigrams	R-AY-T S-OW EH-V-R-IY B-AA-D-IY R-AE-N AX-W-EY
OOD multigrams	R-AY-T S-OW EH-V-R-IY B-AA-D-IY R-AE-N AX-W-EY
words	WHEN MY HUSBAND DIED WE MADE HER COME TO TO FRANCE
phonemes	W EH N M AY HH AH Z B AX N D D AY D W IY M EY D HH ER K AH M T AX T AX F R AE N S
DA multigrams	W-EH-N M-AY-HH-AH Z-B-AX-N-D D-AY-D W-IY-M-EY-D HH-ER K-AH-M-T-AX T-AX F-R-AE-N-S
OOD multigrams	W-EH-N M-AY-HH-AH Z-B-AX-N-D D-AY-D W-IY M-EY-D-HH-ER K-AH-M-T-AX T-AX F-R-AE-N-S
words	EH- ONE DAY THE RUSSIANS THREW A BOMB
phonemes	EH W AH N D EY DH AX R AH SH AX N Z TH R UW AX B AA M
DA multigrams	EH W-AH-N-D-EY DH-AX R-AH-SH-AX-N Z TH-R-UW AX-B-AA-M
OOD multigrams	EH W-AH-N-D-EY DH-AX R-AH-SH-AX-N Z TH-R-UW AX-B-AA-M
words	I LOST ABOUT FORTY FIVE POUNDS
phonemes	AY L AO S T AX B AA UW T F AO R T IY F AY V P AA UW N D Z
DA multigrams	AY-L-AO-S-T AX-B-AA-UW-T F-AO-R T-IY-F-AY-V P-AA-UW-N-D Z
OOD multigrams	AY-L-AO-S-T AX-B-AA-UW-T F-AO-R T-IY-F-AY-V P-AA-UW-N-D Z

Table 2.2: Some example utterance transcripts, segmented automatically at the word, phoneme and phoneme multigram level. Multigram transcripts trained on both the DA and OOD transcript set are shown. Segmentation disagreements are underlined.

and, when a word was not present there, by backing off to a a rule-based word-to-phoneme transliterator [48]. This gave us our larger, DA dictionary of roughly 64,000 words.

For training, we have approximately 200 hours of audio transcribed in 197,220 utterances. In the MALACH training data, an utterance is a short snippet of speech manually specified by a human annotator. We use this complete set for our DA experiments. For our OOD experiments, we subset the complete set of transcriptions to exclude any utterances not covered by our OOD dictionary. This reduces the training set by 12.8% from 197,220 to 172,027 utterances. In this way, we hope to model the speaker and channel characteristics, without unfairly aiding the acoustic

or language models. The DA system is trained on the complete set of utterances. We do not control for the differing quantity of acoustic training data (i.e., the DA system has slightly more data available).

We trained acoustic models as described in [43]. Approximately 800 unique speakers are included in the acoustic training transcriptions. The training transcripts specify when speakers change, and this was utilized for speaker adaptation. For phoneme and phoneme multigram language models, we used only the pronounced training transcripts (or subset of transcripts) for language model training.

The word-level language models were trained using a mixture of the reduced training transcription set, newswire, and conversational telephone speech transcripts. Multigram and phoneme language models were trained using only the acoustic training transcripts. We used Kneser-Ney smoothing [8] for word and multigram level models and used Witten-Bell [61] for phoneme-level models (since there are no phonemes which occur only once, which is necessary for Kneser-Ney smoothing). We did not attempt to prime the language model for particular interviewees or otherwise utilize any interview-level metadata.

2.2.3.2 Decoding

We decode both a small 4.3 hour collection for speech recognition evaluation and a large 589 hour collection for the SR experiments reported in Chapter 4. For decoding the speech retrieval collection, we first ran the down-sampled audio side with greatest RMS amplitude through an available broadcast news speech segmenter. In a few cases, the system was unable to find pauses on which to segment, in which case we arbitrarily segmented the audio into ten-second chunks. On the other hand, because our smaller test set is fully transcribed, we used available manual utterance markers, which include speaker changes, for our LVCSR test data.[7] We ran the same fast

[7]This suggests our speech recognition evaluation results (e.g., WER) may be slightly optimistic, although this is not a problem for inter-system comparisons.

(approximately 1 times real time) system on both collections, as described in [33]. Our decoding dictionary is always the corresponding dictionary used in training. Our multigram dictionary was learned as in Section 2.2.2, using the appropriate set of training transcripts (i.e., the OOD system does not train multigrams on words from the new domain).

All of the speech recognition results reported here use the smaller, transcribed evaluation collection. We report results for each of the several decoding passes. First, the system runs an unadapted decode (**UDEC**), then both a forward and backward pass after speaker adaptation (**ADEC**), and finally one pass of lattice rescoring. The reader is referred to [33] for details on each decoding pass.

2.2.4 Indexing

The output from decoding is a collection of lattices, with phoneme sequences on the arcs. Figures 2.3 and 2.4 show phoneme and multigram examples for the same utterance, containing the words "So in Neustadt". Both lattices were heavily pruned for plotting (discarding all paths with posterior probability less than 20% of the one-best path).

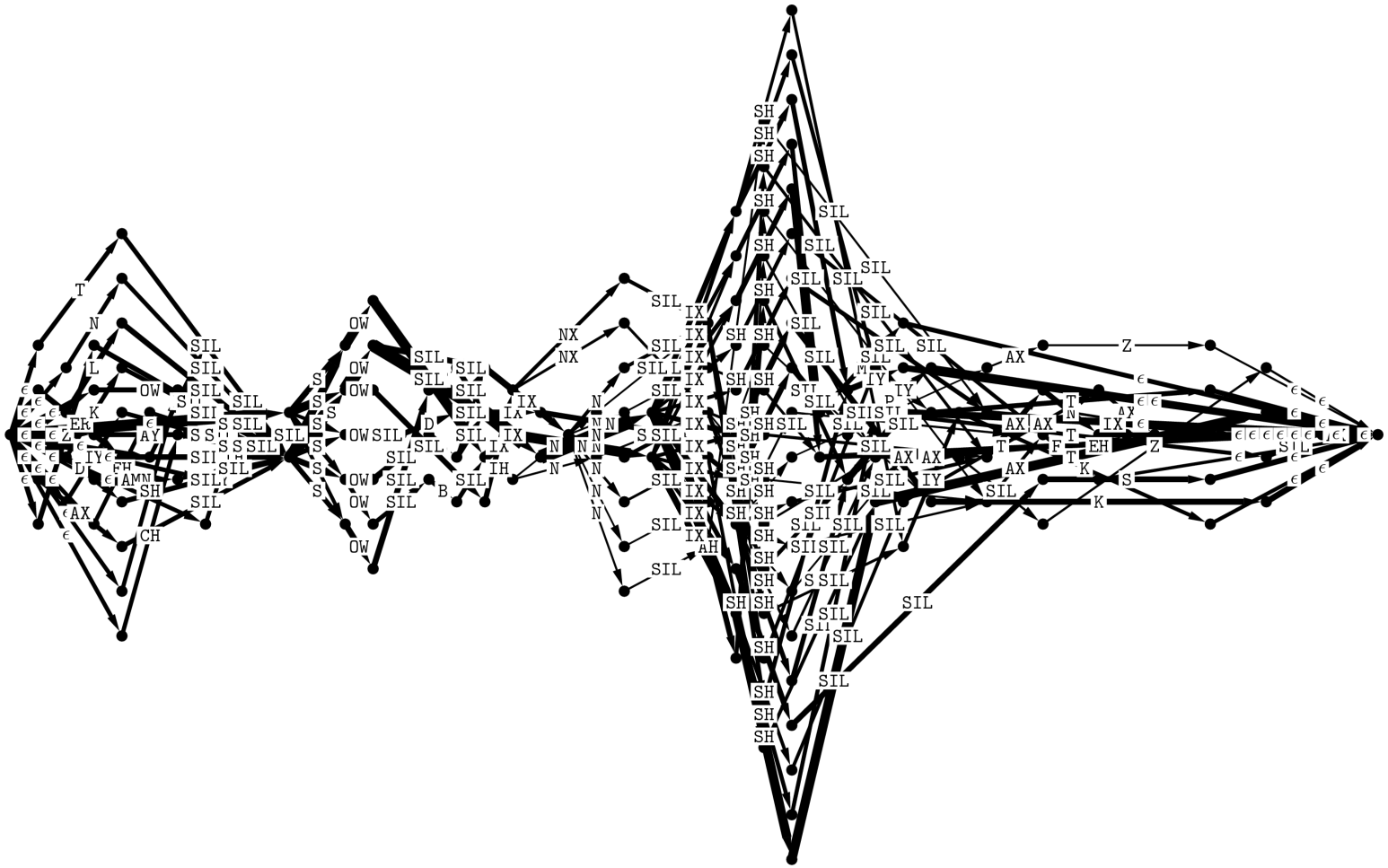


Figure 2.3: A heavily pruned phoneme lattice, for the utterance "So in Neustadt". The most probable phoneme sequence may be roughly pronounced as "so in isht" (`S OW SIL IX N IX SH SIL T`.)

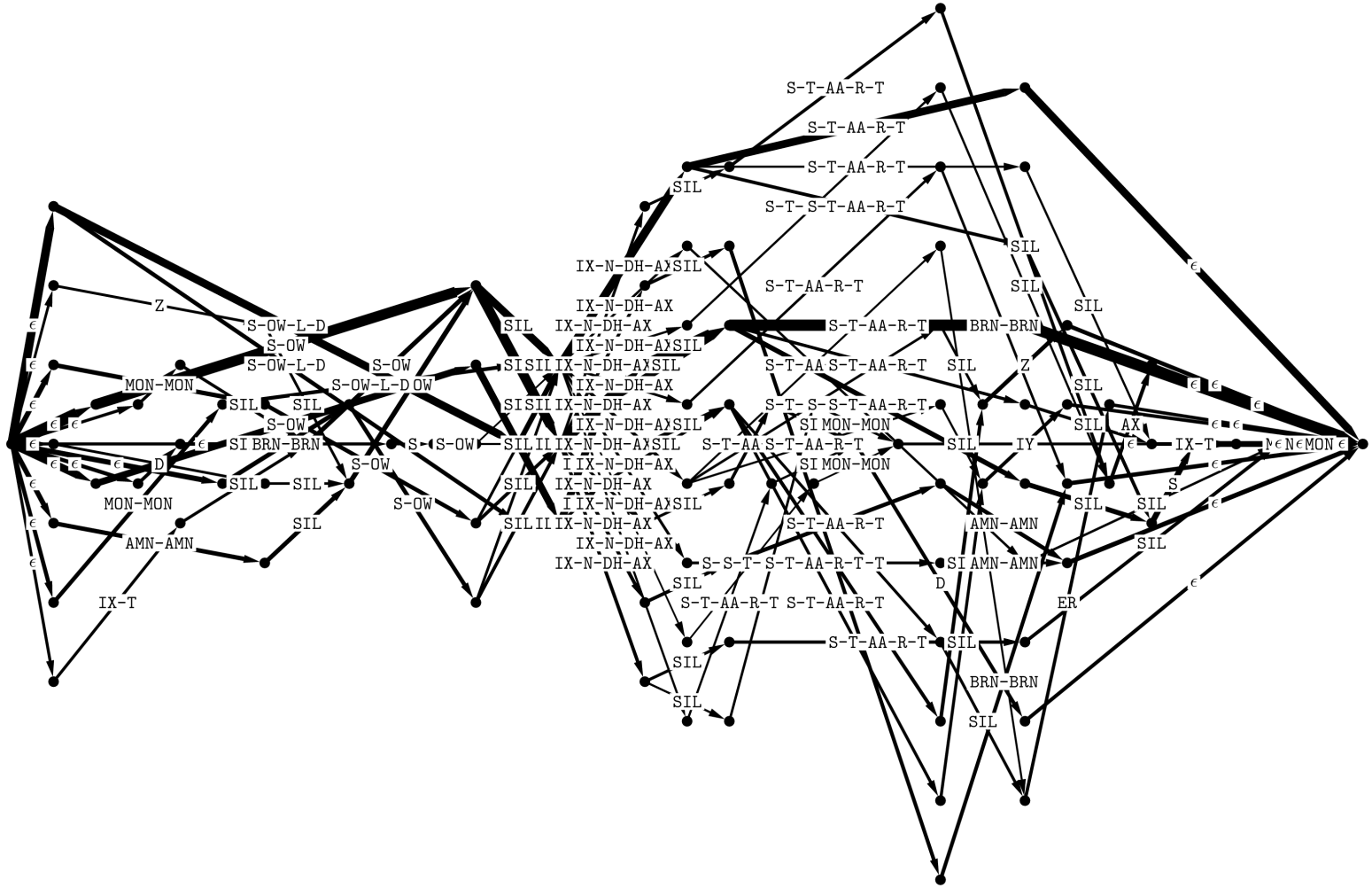


Figure 2.4: A heavily pruned multigram lattice, for the utterance "So in Neustadt". The most probable phoneme sequence may be roughly pronounced as "so in the start".

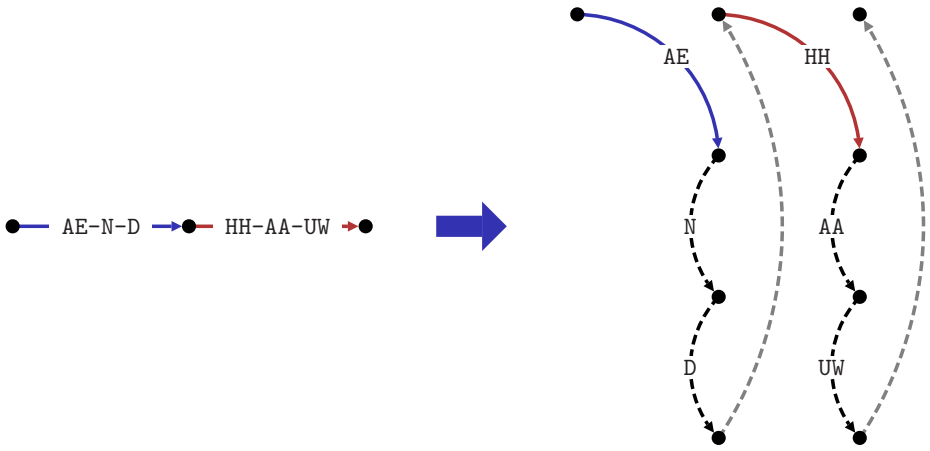


Figure 2.5: An illustration of how we expand multigram and word lattice arcs into phoneme arcs. The multigram arc `AE-N-D` has some non-zero log-probability, which is assigned to the expanded phoneme arc `AE`. Subsequent arcs `N` and `D` do not cover any additional time (indicated by their nodes' horizontal position). Dashed arcs indicate zero log-probabilities.

Since we are interested in indexing large collections of speech and in allowing fast lookup, we'd like to extract features from these lattices that can be inserted into a standard inverted index. Our approach is simple. We convert word and multigram lattices into phoneme lattices by expanding the multi-phoneme arcs into multiple single-phoneme arcs. Because any path through a multigram *must* traverse each of its constituent phonemes (without branching), we assign the original multigram's probability to the multigram's first phoneme and subsequent phonemes are traversed with probability one. Additionally, wince we do not know the time spans which each constituent phones cover, arcs are expanded so that the first phoneme covers the entire time span and subsequent phonemes cover no additional time. Figure 2.5 illustrates this arc expansion process. After lattices have been converted to phoneme lattices, indexing proceeds identically for each system.

Given a phoneme lattice $\mathcal{L}$ containing many paths ℓ (i.e., $\ell \in \mathcal{L}$), the expected number of occurrences for phoneme n-gram $q_1, \ldots, q_n$ over all paths is

$$E_{P_\mathcal{L}}[C(q_1, \ldots, q_n)] = \sum_{\ell \in \mathcal{L}} P_\mathcal{L}(\ell)\ C_\ell(q_1, \ldots, q_n). \tag{2.4}$$

Here, $C_\ell(q_1, \ldots, q_n)$ denotes the number of times phoneme n-gram $q_1, \ldots, q_n$ occurs in lattice path ℓ. The posterior distribution $P_\mathcal{L}(\ell)$ is defined as

$$P_\mathcal{L}(\ell) = \frac{\exp\left\{\sum_{\alpha \in \ell} S(\alpha)\right\}}{\sum_{\nu \in \mathcal{L}} \exp\left\{\sum_{\beta \in \nu} S(\beta)\right\}},$$

where $\exp\{\cdot\}$ denotes exponentiation, as we assume the score $S(\alpha)$ for an arc α on the path is a log probability (e.g., the sum of the acoustic and language model log probabilities). Equation 2.4 can be efficiently computed using a variant of the forward-backward algorithm.[8] Our index then stores, for each phoneme n-gram sequence $n \leq 5$, the set of lattices containing the sequence and their expected counts. We take the same approach for word-level indexing—that is, for indexing words that are *within* the OOD dictionary, indexing the expected counts for each word unigram in each lattice.

Our indexing approach is most closely related to [67], which indexes the expected counts of phoneme sequences from phoneme lattices. This is a state-of-the-art baseline to which we compare our own indexing approach. Our approach differs in that we consider longer, possibly learned, recognition units (i.e., words or multigrams) and in that we first transform our recognition lattices into phoneme lattices as described above. Our baseline technique (the phoneme system) is, essentially, the method proposed in [67].

2.3 Evaluation

2.3.1 OOV Rate

To assess the extent of OOV terms in a collection, the OOV *rate* is commonly reported. The OOV rate is simply the proportion of tokens in a test set that are outside of the recognition dictionary. For IR tasks, however, this way of computing

[8] We use the SRI language modeling toolkit's implementation [53].

Title	**Hasidism**
Description	**Hasidim** and their unquestioning faith
Narrative	The relevant material should talk about **Hasidism** before, during, and after the Holocaust. The information about **Hasidic** dynasties and geographic localities that were established and destroyed.
Title	**Sonderkommando**
Description	Interested in descriptions of the daily horror witnessed by these **Sonderkommando** units. Also want information about the events that culminated in the blowing up of **Crematorium** III in **Birkenau** on Octorber 7, 1944 (**Sonderkommando** Uprising).
Narrative	In particular, looking for information about: How were the men chosen for **Sonderkommando**? How long did they usually work in the crematorium before they were replaced by fresh recruits? What was the process? What did they actually see? How did the **Sonderkommando** relate to the victims as they were led from the disrobing room into the gas chamber? What were the special privileges afforded to **Sonderkommando** in terms of living arrangements and food?
Title	**Kindertransport**
Description	We are looking for first-hand accounts of people who were saved by the **Kindertransport** program, specifically on the **Dunera** and **Arandora** Star.
Narrative	The relevant material should include interviews, photographs, or artifacts that deal with the **Kindertransport**.

Table 2.3: Title, description, and narrative fields for several topics in the MALACH collection. OOV terms are shown in bold.

OOV rate may not be a good assessor of vocabulary mismatch, since it counts OOV terms which may never be used in queries. For example, *Eichmann* is outside the vocabulary of our OOD dictionary, but minimally effects the OOV rate as normally computed, because it is rare. Nevertheless, since *Eichmann* is used in a SR topic's two-word title field, "Eichmann witnesses", its absence will seriously degrade SR performance on that topic. Accordingly, we instead report both the standard OOV rate (on the complete set of test transcripts) and the "rate" on each of the topic's fields from the complete set of SR topics. For example, the OOV rate on the title field is

Dictionary	T	TD	TDN	Transcripts
DA	4.1	2.0	1.7	1.6
OOD	12.2	5.2	4.0	2.2

Table 2.4: OOV rates (%) for the test transcripts, title (T), title plus description fields (TD), and title plus description plus narrative (TDN) words, using both the OOD and DA decoding dictionaries.

the proportion of all tokens in the field which are OOV.

Table 2.3 shows several example topics, each having a short title (T), description (D), and narrative (N) field. Roughly, the title field may be viewed as a short query representation of the information need (akin to what a user might enter into a search engine). The description is a longer representation, akin to what a user might first say to a librarian when requesting assistance. Lastly, the narrative is a more complete specification of the information need, akin to what a librarian might understand as the information need after several iterations of clarification. That is, a narrative is intended to contain enough information for a human to reasonably assess whether a new document is relevant. By convention, when a query is formulated using just the title field, we refer to it as T. When using both title and description, TD. When using title, description, plus narrative, TDN.

Table 2.4 shows the OOV rates for both the OOD dictionary and the DA dictionary. Rates are shown on the evaluation transcript set, as well as on query words for T, TD, and TDN queries. Measured on transcripts, we observe the OOV rate increases only slightly from 1.6% to 2.2% from using the OOD rather than the DA dictionary. However, the rate increases more sharply when measured with respect to query words. For example, if we consider the OOV rate on title words (words in very short queries), the rate increases substantially from 4.1 to 12.2%. For comparison, an OOV rate of 12% was previously reported for query words in a live search engine, indexing speech audio from the Web [19].

We also see from Table 2.4 that the OOV rate decreases for TD and TDN terms

(and further for transcripts). We expect this is primarily because the mean Document Frequency[9] (DF) for a term will typically be larger in D than in T (and higher still in N), since we would expect a human to use increasingly discriminative terms to express a topic when becoming more terse. This is illustrated by Figure 2.6, which displays the OOV rate for each set vs. the average normalized DF of its terms. As average normalized DF increases, we observe the OOV rate with respect to the OOD dictionary fall quickly. When the OOV rate is computed with respect to the DA dictionary, it falls off comparatively much more slowly.

2.3.2 Word Error Rate

Word Error Rate (WER) is calculated by first producing an alignment of the hypothesis and reference transcripts (such that the the total penalty for insertions, substitutions, and deletions is minimized). The counts of these errors are then used to compute WER as,

$$WER = 100 \cdot \frac{S + D + I}{N},$$

where S, D, I are the number of substitutions, insertions, and deletions respectively, while N is the length (in words) of the reference.

Tables 2.5 and 2.6 give word error rates for our test collection, using both the OOD and DA decoding dictionaries. As we expect, for successive passes of the recognition system, WER decreases. We obtain a best WER of 31.63. Surprisingly, we obtain a better WER on the OOD system (31.63) than on the DA system (32.40)—despite the OOD system having 12.8% less transcribed audio for acoustic modeling. It is difficult to know precisely why this occurred, although it may be because DA words present additional modeling difficulties. In particular, in the MALACH collection, these words

[9] *Document frequency* is the number of segments in a collection which contain a term. Recall, by *segment* we here mean the short topically coherent passages of continuous speech manually defined by human annotators. We define *normalized document frequency* as the proportion of segments in the collection which contain a term.

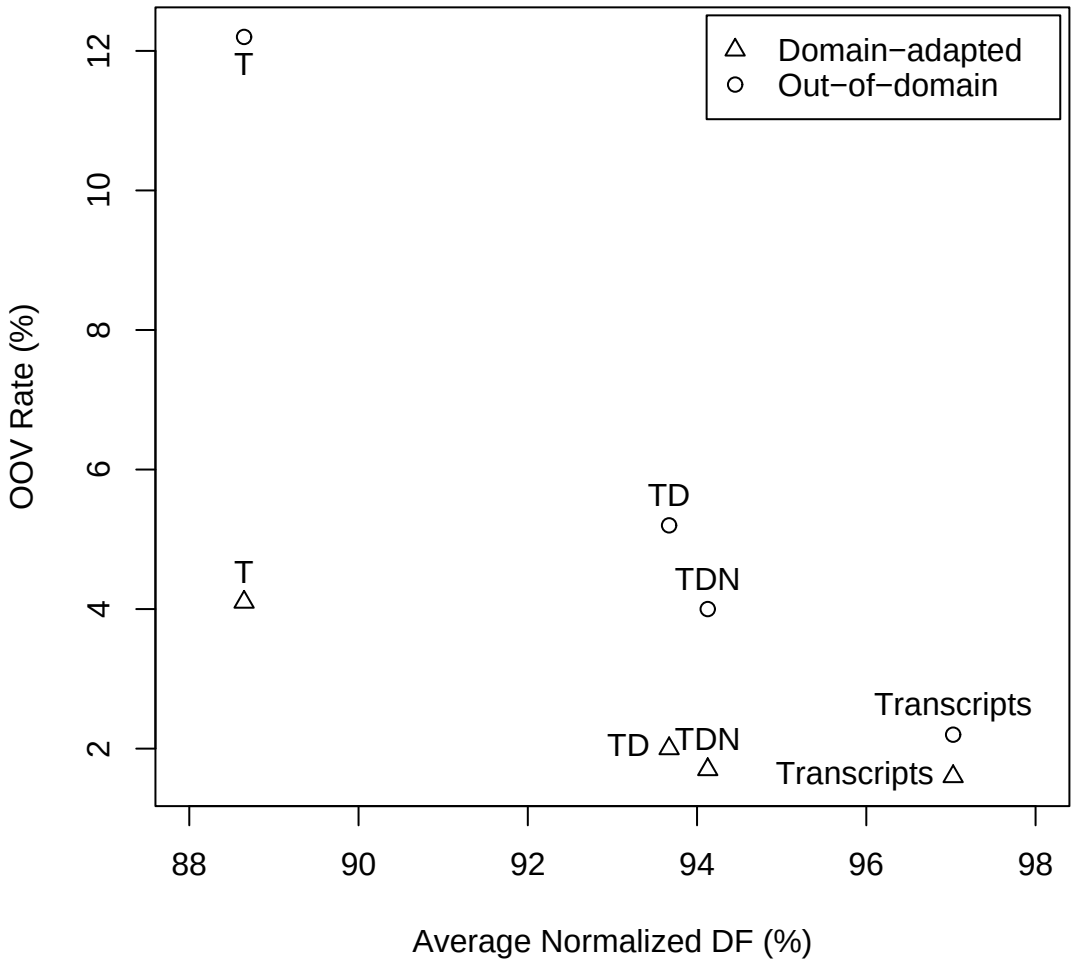


Figure 2.6: The OOV rates from Table 2.4 plotted against the average normalized document frequency for a term in the same set. Statistics computed using both the OOD (◯) and DA (△) dictionaries are shown.

pass	forward	backward	cross-word
UDEC	47.06	-	-
ADEC	37.02	32.85	-
lattice rescoring	-	-	31.63

Table 2.5: Word error rates for various passes of the word-level recognition system on MALACH development data, produced using the OOD recognition system.

pass	forward	backward	cross-word
UDEC	47.94	-	-
ADEC	37.86	33.71	-
lattice rescoring	-	-	32.40

Table 2.6: Word error rates for various passes of the word-level recognition system on MALACH development data, produced using the DA recognition system.

are often heavily accented and may have poorly matched dictionary pronunciations. In view of Table 2.4, this at least underscores how cautious we must be in interpreting a lower WER to imply a better retrieval system. A better WER does not mean you will do better at the words of interest (e.g., query terms).

These WER results are useful for system comparison, although we must use caution when extrapolating them to the untranscribed SR collection data. First, unlike our LVCSR test data, the SR data has no manually defined utterance markings and automatic segmentation may degrade recognition accuracy. Secondly, we don't know when speakers change in the SR data, and we expect recognition of cross-talk speech to be poor. We validate our LVCSR performance on the SR data, in Chapter 4, using an extrinsic evaluation. In that evaluation, we consider the downstream task of *ad hoc* speech retrieval using our LVCSR output.

2.3.3 Phone Error Rate

Like WER, Phoneme Error Rate (PER) is calculated by first producing an alignment of the hypothesis and reference transcripts, although now at the level of phonemes.

As before, the counts of each error type are used to compute

$$PER = 100 \cdot \frac{S + D + I}{N},$$

where S, D, I are the number of substitutions, insertions, and deletions respectively, while N is the length (in phonemes) of the reference.

Tables 2.7 and 2.8 show the phoneme error rates for each system, after various recognition passes, using both the OOD and DA system respectively. We show phoneme error rates from both the phoneme and phoneme-multigram level systems, as well as phoneme error rates computed from word transcripts (with words simply replaced by their pronunciations). First, we see in both tables that the best phoneme error rate for the multigram system is roughly half that of the phoneme-only system. Comparing the two tables shows that the non-word systems are both improved by the larger (DA) dictionary and the small increase in training data.[10] Finally, both tables show that the word systems achieve a considerably lower PER than the best multigram PER. These reductions in PER represent improvements over state-of-the-art methods for constructing inverted indices using phoneme recognition lattices [67].

We see in both Tables 2.7 and 2.8 that the PER for our word-based systems is significantly better than for the multigram systems. First, we must note that the word system has an important advantage in this evaluation because our reference phoneme transcripts were produced using the word level dictionaries (not the phonemes actually spoken). Nevertheless, we conducted several *a posteriori* experiments to investigate if other factors may account for this discrepancy in PER. One possible reason recognition improves for the word system is that a considerably larger language modeling corpus is available (since we use not only the acoustic training transcripts but also mixtures of newswire, conversational telephone transcriptions

[10]Note that the phoneme and multigram OOD systems also have slightly less training data because, just as for the word-level system, utterances containing words outside the OOD dictionary are not used for any training.

Recognition Unit	pass	forward	backward	cross-word
Phones	UDEC	71.19	-	-
	ADEC	60.22	69.16	-
	lattice rescoring	-	-	64.36
Phone multigrams	UDEC	42.58	-	-
	ADEC	33.60	32.79	-
	lattice rescoring	-	-	32.05
Phones from words	UDEC	33.60	-	-
	ADEC	24.70	22.09	-
	lattice rescoring	-	-	20.47

Table 2.7: Phone error rates for various passes of a phoneme-, multigram-, and word-level speech recognition system on MALACH test data, trained using the OOD system.

Recognition Unit	pass	forward	backward	cross-word
Phones	UDEC	64.56	-	-
	ADEC	59.30	67.28	-
	lattice rescoring	-	-	62.34
Phone multigrams	UDEC	42.40	-	-
	ADEC	33.65	32.37	-
	lattice rescoring	-	-	31.63
Phones from words	UDEC	34.48	-	-
	ADEC	25.31	22.69	-
	lattice rescoring	-	-	22.06

Table 2.8: Phone error rates for various passes of a phoneme-, multigram-, and word-level speech recognition system on MALACH test data, trained using the DA system.

and other sources). Our multigram implementation only uses the acoustic transcripts for language modeling. To test this, we reran word-level decoding after re-training the OOD word language model on only the acoustic word-level transcripts. However, this only slightly degraded the word system's PER to 22.08% (from 20.47%). Secondly, since the recognition system was tuned for words, we considered optimizing our decoding parameters on the multigram system for the multigram test data. While unfair for evaluation, this at least tells us something about the potential gain from better parameter selection. Tuning for the test data, our PER improved from 32.05%

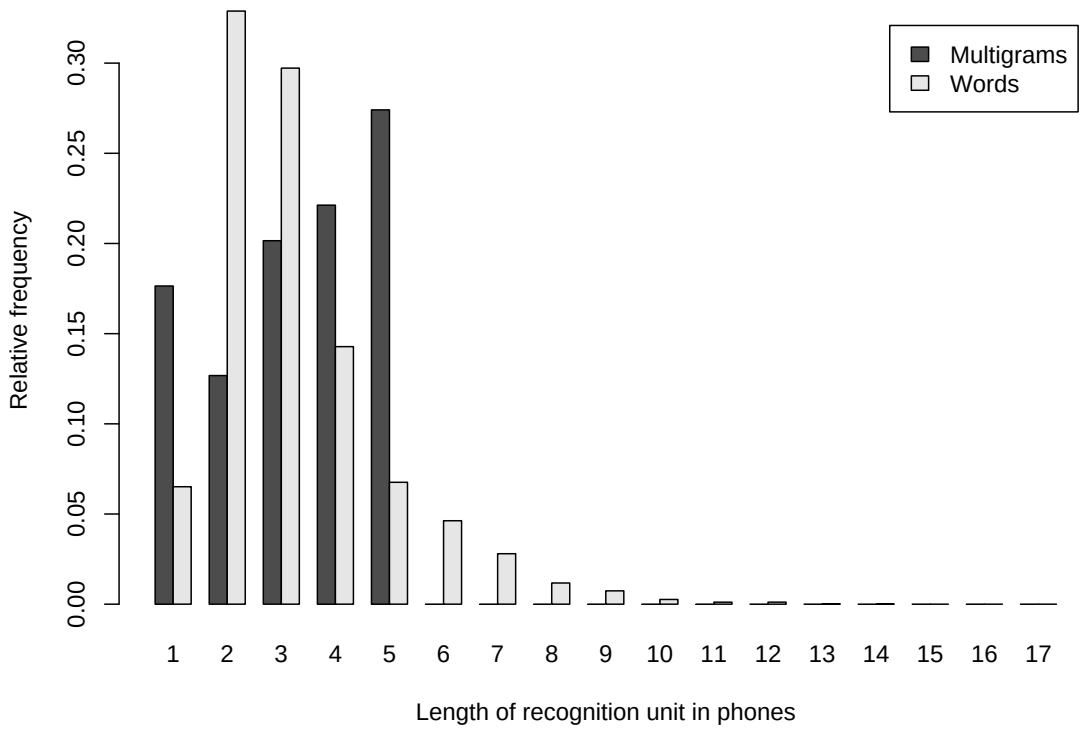


Figure 2.7: Histograms of the length (in phonemes) of all recognition units from the acoustic training transcripts. Data from words and multigrams are shown. Note that, of course, all recognition units for the phoneme-level system (not shown) have length of one.

to 29.65%. This still leaves a significant gap in PER between the multigram and word-level systems.

The larger effect appears then to be due simply to the context size available to the multigram and word-level systems. While we chose our multigrams to contain five or fewer phonemes, real words can be much longer (our longest word contains 17 phonemes) and longer recognition units provide strong constraints in recognition. Looking at our acoustic training transcripts (*not* simply the recognition unit dictionaries), we see that the mean phoneme length for words and multigrams is similar at 3.212 and 3.29 respectively (their median length is the same at 3 phonemes). Nevertheless, the distribution of observed phoneme lengths for recognition units is very

different. Figure 2.7 shows histograms, side by side, of the phoneme length of all observed recognition units for the word and multigram acoustic transcripts. Of course, the word system has a much longer right-hand tail; roughly 15% of the observed words are longer than the longest allowable multigram.

2.3.4 Indexing

Looking again at Figures 2.3 and 2.4, we can observe several benefits of indexing by expected phoneme sequence counts. First, we observe that many alternate paths through a lattice do not provide new phoneme sequences, but only alternate segmentations of the same phoneme sequence (e.g., the multigram `S-T-AA-R-T` in Figure 2.4). These alternate segmentations are not expected to be useful to us, in part because we focus in this work on detecting terms at the granularity of an *utterance*.[11] Moreover, we can significantly reduce the space and time costs for search by using this conflated representation. As an example, while the lattice in Figure 2.4 contains 248 arcs (each having start and end nodes, associated phonemes, and an arc probability), it contains only 201 unique phoneme sequences of length five or less. We consider this further in Section 2.3.5.

To illustrate our indexing approach, Table 2.9 gives the five largest phoneme sequence counts for each sequence length for the lattices in Figures 2.3 and 2.4. Note both contain the correct phoneme sequence `S-OW-IX-N` for the in-vocabulary words "so in" with high expected count. However, on the OOV term "Neustadt" (pronounced roughly "Noy-stot"), they disagree significantly—with the multigram system being phonemically much closer (compare the multigram `S-T-AA-R-T` with the correct `S-T-AA-T`). We see in both Figures 2.3 and 2.4 that the recognizers' uncertainty increases sharply with the onset of the OOV term "Neustadt" (i.e., the lattices' bushiness increases), so it is not surprising that the phoneme and multigram

[11] Detecting terms at a granularity finer than the utterance might be important for proximity-based indexing. For example, see [7].

	Phoneme System		Multigram System	
Sequence length	**Sequence**	**Count**	**Sequence**	**Count**
1	IX	1.38	S	2.33
	S	1.07	T	2.21
	N	1.07	IX	1.12
	OW	1.05	AA	1.00
	SH	1.05	N	1.00
2	S-OW	1.00	S-OW	1.00
	IX-N	0.70	T-AA	1.00
	N-IX	0.57	S-T	1.00
	IX-SH	0.57	IX-N	0.99
	SH-AX	0.39	AA-R	0.90
3	N-IX-SH	0.57	S-T-AA	1.00
	IX-N-IX	0.39	AA-R-T	0.90
	S-OW-IX	0.32	T-AA-R	0.90
	OW-IX-N	0.27	S-OW-IX	0.81
	S-OW-B	0.23	OW-IX-N	0.81
4	IX-N-IX-SH	0.39	S-T-AA-R	0.90
	S-OW-IX-N	0.27	T-AA-R-T	0.90
	IH-N-IX-SH	0.18	S-OW-IX-N	0.81
	N-IX-SH-AX	0.15	AX-S-T-AA	0.78
	OW-IX-N-IX	0.15	IX-N-DH-AX	0.78
5	OW-IX-N-IX-SH	0.16	S-T-AA-R-T	0.90
	S-OW-IX-N-IX	0.15	IX-N-DH-AX-S	0.78
	S-OW-B-IX-N	0.15	N-DH-AX-S-T	0.78
	S-OW-T-IX-N	0.14	DH-AX-S-T-AA	0.78
	S-OW-D-IX-N	0.14	AX-S-T-AA-R	0.71

Table 2.9: The five phoneme sequences having largest expected count for each phoneme sequence length, for the lattices from Figures 2.3 and 2.4.

expected counts disagree on this unseen term. The multigram system is able to utilize its larger context to do *better* than the phoneme system, but this context also hurts it—here, we see it hallucinate the phoneme `R`. This common phenomenon will motivate several of our term frequency estimation approaches (see Section 3.3).

We also see from Table 2.9 that multigram counts tend to be much higher for longer sequences (because there is no cost for additional phonemes within the same multigram). This suggests that, if we want to use these counts as features for detecting term occurrences, we may benefit from first learning a suitable transformation of the counts. We explore this in Section 3.4.

2.3.5 Index Size

Suppose we consider phoneme n-grams of order $n \leq 5$. Since our phoneme inventory contains 39 phonemes, in the worst case we would have to store $\sum_{i=1}^{5} 39^i = 92{,}598{,}519$ n-grams in our index. If we increase n by 1, we must index as many as 3.52 billion additional n-gram sequences. Increase n again, and we must consider an additional 137.23 billion sequences. Clearly, this could quickly become difficult to manage. Fortunately, many of these sequences will be phonotactically impossible and many more will simply never occur in our language of interest. With a sufficiently powerful language model, we ought to be able to prune away most of these sequences and retain manageable index sizes.

Figure 2.8 shows the number of n-gram sequences contained in our test utterances' lattices, using both the phoneme and multigram recognizers. We see that the phoneme lattices tend to have considerably more phoneme sequences per lattice (the slope of the least squares fit is 1.96). We expect this is simply due to the comparatively weak language models that the phoneme system uses. While the multigram system's language model is the same *order* as the phoneme system's, its effective context history is much larger, because each recognition unit may contain multiple phonemes.

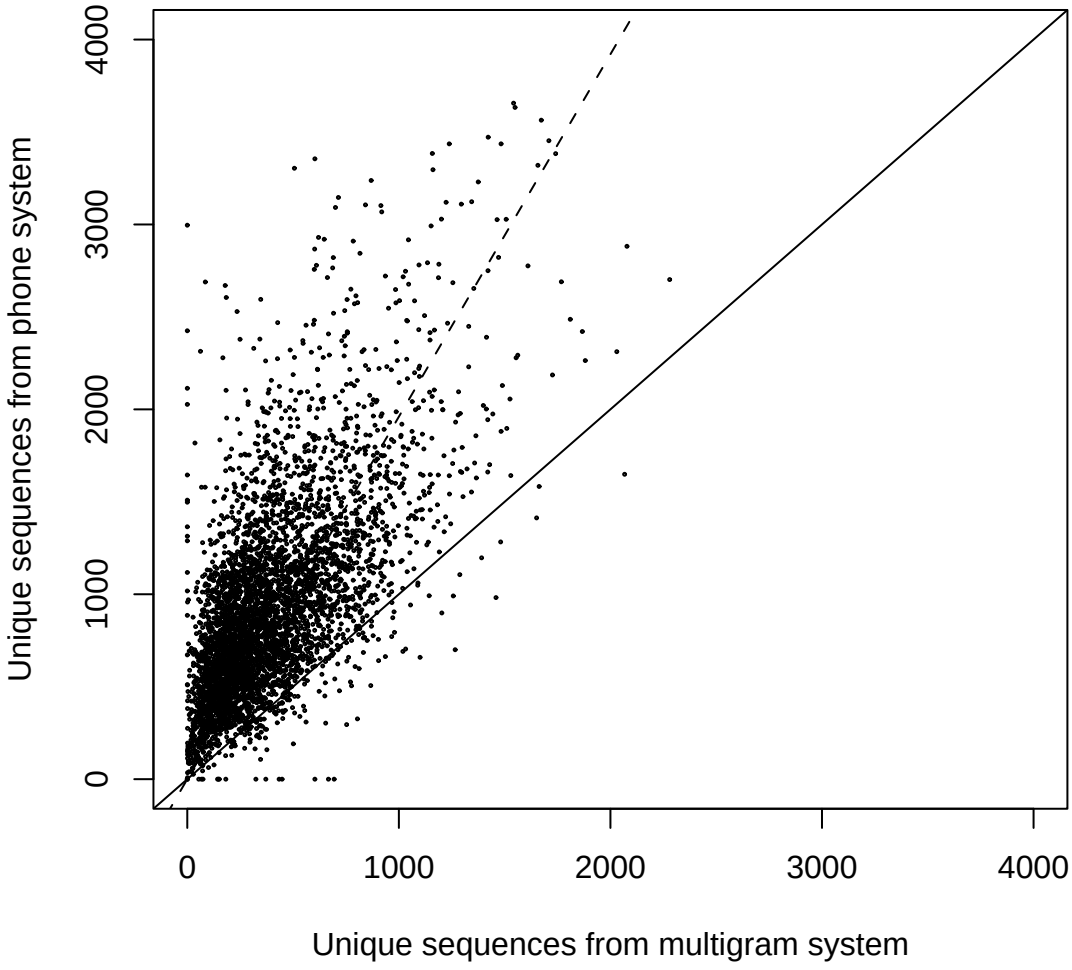


Figure 2.8: The number of unique phoneme n-grams, $n \leq 5$, for each test utterance's lattice, using both a phoneme and multigram recognizer. Along the solid line, lattices would have the same number of n-grams. The least squares fit of the data is shown as a dashed line, with a slope of 1.96.

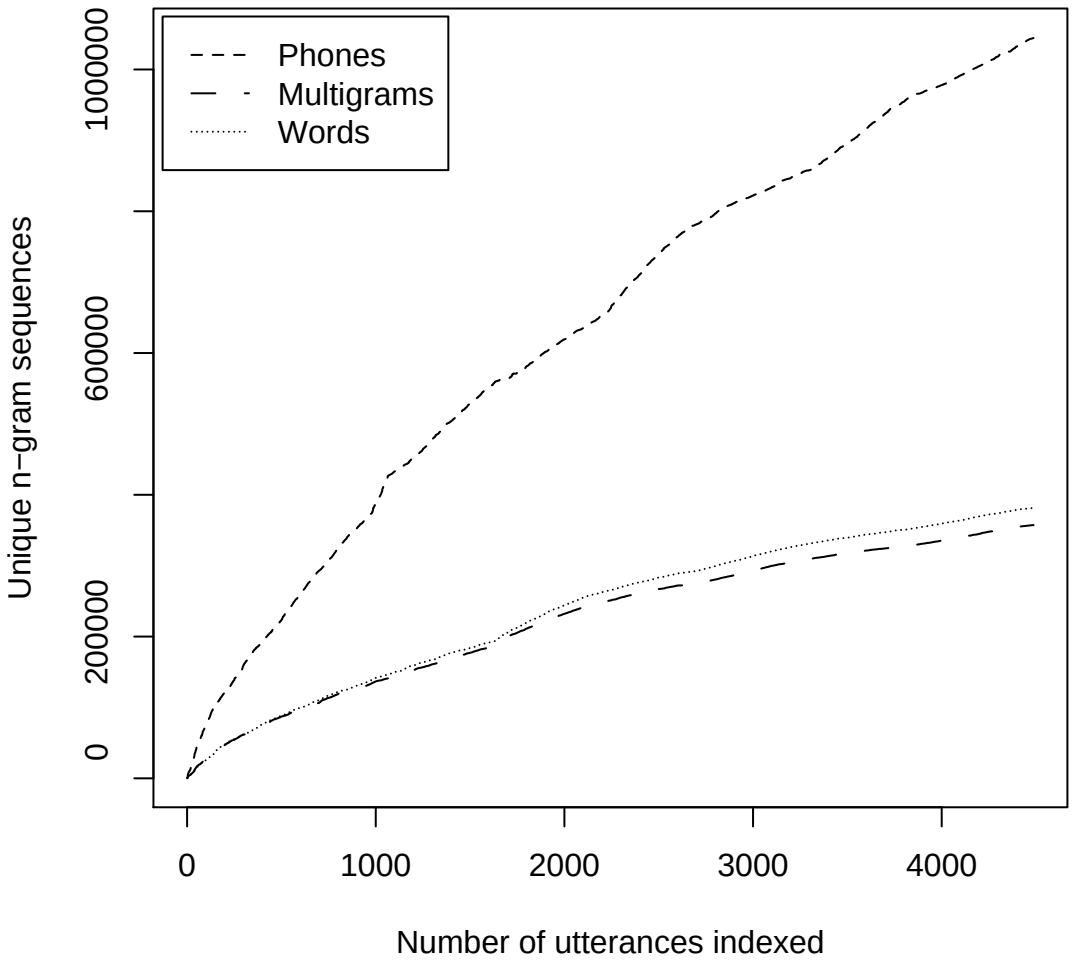


Figure 2.9: The total number of index keys (unique phoneme n-grams) vs. indexed utterances for the phoneme, multigram, and word systems.

Figure 2.9 shows that, for the same number of indexed utterances, the multigram system also has far fewer keys (unique phoneme n-grams) to index than it's phoneme-based counterpart (our baseline). This is so both because, as Figure 2.8 indicates, there are fewer unique phoneme sequences in each particular multigram lattice and because fewer phoneme sequences are *probable* under the multigram system. That is to say, the system has much more evidence (i.e., context) from which to reject phonotactically impossible or even phonemically improbable sequences. By comparison, obtaining phonemes from word lattices would completely prevent phonotactically impossible sequences—but would also only allow sequences obtainable from concatenations of pronunciations in the LVCSR dictionary. The multigram system does allow phonotactically impossible sequences (because phoneme unigrams are never removed from the multigram dictionary), which we hope to compensate for by more flexible handling of new phoneme sequences.

The reduction in index size afforded by longer recognition units represents an important advantage over our baseline index, constructed using phonemes. Indeed, the large number of phoneme sequences obtainable by traversing phoneme unigram lattices has inspired previous work in reducing the size of phoneme lattice indices. For example, in [67], a whitelist of phoneme sequences was created and only sequences within the list were indexed. At query time, sequences not in the whitelist were handled by backing off to phoneme sequences of shorter length. In [52], a small set of discriminative phoneme sequences were chosen for indexing using several heuristics. This greatly reduced the index size, but the method was only designed to retrieve a set of candidate lattices for a second-pass term detection algorithm. Our approach reduces the index size in a conceptually simpler way, by reducing the candidate recognition units *a priori*, rather than discarding putatively less useful sequences after recognition. This has the additional benefit of improving PER with respect to phoneme unigram recognition, as shown above.

2.4 Chapter Summary

In this chapter, we introduced and evaluated our approach to speech indexing. For vocabulary-independent indexing, we showed that, by considering longer recognition units such as phoneme multigrams or words, we could both significantly reduce our recognition error rate and the size of our inverted index. We constructed this index using the expected counts of phoneme sequences in the recognition lattices. For the multigram and word recognition cases, these counts were computed after first expanding the lattices' arcs to produce new lattices containing phonemes.

We found that our word-level system produced our best phoneme error rate, which we attribute simply to the greater context provided by longer words. On the other hand, multigram indexing slightly reduced our resulting index sizes, which may be useful for some very large applications. We found that indexing via phoneme-level recognition, while having the least annotation costs, produced considerably less accurate recognition and indices which were much larger than the similarly trained word or multigram systems. While we were able to train on nearly 200 hours of transcribed speech, it may be that phoneme systems would be more competitive if much less annotation were available.

To evaluate our recognition systems, we relied in this chapter primarily on intrinsic measures of the one-best hypothesis quality (e.g., phoneme error rate). We may also evaluate a system using an extrinsic measure, that is, its performance in a downstream task. As as extrinsic evaluation of these subword indexing systems, we consider in Chapter 3 the problem of ranking utterances by our confidence that they contain a term. We defer an extrinsic evaluation of our word-level indexing until Chapter 4, in which we utilize word frequency estimates for *ad hoc* speech retrieval.

Of course, our goal in this dissertation is not only to produce better phoneme transcripts or lattices, but to find topically relevant segments in an *ad hoc* SR task. We have focused on vocabulary-independent indexing because, as we have seen, queries

often contain words which are outside of an OOD LVCSR dictionary. Now that we have these improved indices, we can consider the problem of how best to find OOV words using them. We begin this discussion in Chapter 3.

Chapter 3

Vocabulary-Independent Ranked Utterance Retrieval

3.1 Introduction

In this chapter, we focus on finding the words that could not be anticipated by our OOD LVCSR dictionary. To do this, we use the vocabulary-independent indices that we produced using phoneme, multigram, and word recognition in Chapter 2. Our goal is to rank utterances by our confidence that they contain an arbitrary query word. This task, which we refer to as *ranked utterance retrieval*, sits between *Spoken Term Detection* (STD) and SR. The goal of STD is to detect the set of locations of a query term in a speech collection. Unlike STD, we do not attempt to identify the absolute location of the term at a granularity finer than that of a speech utterance (only it's presence or absence).[1] Unlike SR however, we aim here only to detect that an utterance contains a term—not that an utterance is relevant to an information need. Of course, we expect ranked utterance retrieval to be a useful component of a full SR system (i.e., better ranked utterance retrieval can reasonably be expected to render better SR), but we will consider ranked utterance retrieval and *ad hoc* SR separately for evaluation. This chapter's contribution is to present several novel methods for improving ranked utterance retrieval. In Chapter 4, we utilize our best ranked utterance system from this study to improve our performance on an *ad hoc* retrieval task.

Ranked utterance retrieval is difficult because our vocabulary-independent indices are imperfect. To find a word, we must account for the mismatch between the query's phonemes and an errorfully recognized phoneme sequence. This mismatch

[1] This is sufficient for our SR approach, and it allows us to use the compact indices from Chapter 2 that discarded within-lattice timing information.

may have many causes, on both the human speech production and the automatic speech recognition side, including pronunciation variability and various peculiarities of the recognizer (e.g., a particular phoneme may be systematically misrecognized). We focus on improving ranked utterance retrieval by accounting for this mismatch in several new ways. First, we introduce our baseline approach in Section 3.2. Then, in Section 3.3 we consider incorporating learned, alternate pronunciations, or *degradations*, of a term which simultaneously capture variation due to pronunciation and errorful recognition. In Section 3.4, we propose a new discriminative estimator for term frequency, using simple features extracted from the utterance representations. This discriminative model is able to significantly improve retrieval effectiveness when only one pronunciation is considered. We introduce our experimental validation in Section 3.5 and our results in Section 3.6. We show that each of our new methods can significantly improve upon a baseline generative approach, although the best approach will depend on the constraints of a particular task. Finally, in Section 3.7 we sum up and outline directions for future work.

3.2 Generative Baseline

Each method we present ranks the utterances by the term's expected frequency within the corresponding phoneme lattice. This general approach has previously been considered [67, 47], on the basis that it provides a minimum Bayes-risk ranking criterion [66, 45] for the utterances. What differs for each method is the particular estimator of term frequency which is used. We first outline our baseline approach, a Generative model for Term Frequency Estimation (**GTFE**). This represents a state-of-the-art approach for ranked utterance retrieval with OOV query words. If the precise location of an OOV word is also required (at a granularity finer than an utterance), this method can also be used as a high-recall first pass in a two-stage system before a second-pass linear scanning of the phoneme lattices confirms the word's

presence [67]. Again, our goal here is only to detect the word's presence within an utterance, although we note that our new methods can be incorporated into such a two-stage system if desired.

Recall that our vocabulary-independent indices from Chapter 2 contain the expected counts of phoneme sequences from our recognition lattices. Yu et al. [67] used these expected phoneme sequence counts to estimate term frequency in the following way. For a query term Q and lattice $\mathcal{L}$, term frequency $\widehat{tf}_G$ is estimated as

$$\widehat{tf}_G(Q, \mathcal{L}) = P(Q|\mathcal{L}) \cdot N_{\mathcal{L}},$$

where $N_{\mathcal{L}}$ is an estimate for the number of words in the utterance. The conditional $P(Q|\mathcal{L})$ is modeled as an order M phoneme level language model,

$$\hat{P}(Q|\mathcal{L}) = \prod_{i=1}^{l} \tilde{P}(q_i|q_{i-M+1}, \ldots, q_{i-1}, \mathcal{L}), \tag{3.1}$$

so that

$$\widehat{tf}_G(Q, \mathcal{L}) \approx \hat{P}(Q|\mathcal{L}) \cdot N_{\mathcal{L}}. \tag{3.2}$$

For this model, the probability of a query phoneme q_j being generated, given that the phoneme sequence $q_{j-M+1}, \ldots, q_{j-1}$ was observed, is estimated as

$$\tilde{P}(q_j|q_{j-M+1}, \ldots, q_{j-1}, \mathcal{L}) = \frac{E_{P_{\mathcal{L}}}[C(q_{j-M+1}, \ldots, q_j)]}{E_{P_{\mathcal{L}}}[C(q_{j-M+1}, \ldots, q_{j-1})]}.$$

Recall, $E_{P_{\mathcal{L}}}[C(q_{j-M+1}, \ldots, q_{j-1})]$ denotes the expected count in lattice $\mathcal{L}$ of the phoneme sequence $q_{j-M+1}, \ldots, q_{j-1}$ (see Equation 2.4).

In practice, because of data sparsity, the language model in Equation 3.1 must be modified to include smoothing for unseen phoneme sequences. Accordingly, we use a backoff M-gram model with Witten-Bell discounting [61]. We set the phoneme language model's order to $M = 5$, which gave good results in previous work [67]. The

core of our implementation is built using the SRI language modeling toolkit [53].

3.3 Incorporating Query Degradations

One problem with the generative approach of Section 3.2 is that recognition error is not modeled (apart from the uncertainty captured in the phoneme lattice). The essential problem is that while the method hopes to model $P(Q|\mathcal{L})$, it is in fact only able to model the probability of one *degradation* H in the lattice, that is $P(H|\mathcal{L})$. We define a query degradation as any phoneme sequence (including the lexical sequence) which may, with some estimated probability, occur in an errorful phonemic representation of the audio (either a one-best or lattice hypothesis). Because of speaker variation and because recognition is errorful, we ought to also consider non-lexical degradations of the query phoneme sequence. That is, we should incorporate $P(H|Q)$ in our ranking function.

It has previously been demonstrated that allowing for phoneme confusability can significantly increase spoken term detection performance on one-best phoneme transcripts [6, 48] and in phonemic lattices [15]. These methods work by allowing weighted substitution costs in minimum-edit-distance matching. Previously, these substitution costs have been maximum-likelihood estimates of $P(H|Q)$ for each phoneme, where $P(H|Q)$ is easily computed from a phoneme confusion matrix after aligning the reference and one-best hypothesis transcript under a minimum edit distance criterion. Similar methods have also been used in other language processing applications. For example, in [25], one-for-one character substitutions, insertions and deletions were considered in a generative model of errors in OCR.

In this work, because we are focused on constructing inverted indices of audio files (for speed and to conserve space), we must generalize our method of accounting for query degradations. Given a degradation model $P(H|Q)$, we take as our ranking function the expectation of the estimate $N_{\mathcal{L}} \cdot \hat{P}(H|\mathcal{L})$ (Equation 3.2's right hand side)

with respect to $P(H|Q)$,

$$\widehat{tf}_G(Q, \mathcal{L}) = \sum_{H \in \mathcal{H}} \left[\hat{P}(H|\mathcal{L}) \cdot N_{\mathcal{L}} \right] \cdot P(H|Q), \tag{3.3}$$

where $\mathcal{H}$ is the set of degradations.[2] Note that, while we consider the expected value of our baseline term frequency estimator with respect to $P(H|Q)$, this general approach could be used with any other term frequency estimator. In our experiments and analysis, we abbreviate this generative term frequency estimator with query degradation as **GTFE-QD**.

Our GTFE-QD formulation is similar to approaches taken in OCR document retrieval, using degradations of character sequences [9, 10]. For vocabulary-independent spoken term detection, perhaps the most closely related formulation is provided by [31]. In that work, they ranked utterances by the weighted average of their matching score, where the weights were confidences from a grapheme to phoneme system's first several hypotheses for a word's pronunciation.[3] The scores were edit distances, where substitution costs were weighted using phoneme confusability. Accordingly, their formulation was not aimed at accounting for errors in recognition, but rather for errors in hypothesizing lexical pronunciations. We expect this accounts for their lack of significant improvement using the method.

Since we don't want to sum over all possible recognition hypotheses H, we might instead sum over the smallest set $\mathcal{H}$ such that $\sum_{H \in \mathcal{H}} P(H|Q) \geq \gamma$. That is, we could take the most probable degradations until their cumulative probability exceeds some threshold γ. In practice, however, because degradation probabilities can be poorly scaled, we instead take a fixed number of degradations and normalize their scores. When a query is issued, we apply a degradation model to learn the top few phoneme

[2] Equation 3.3 is an abuse of notation since we are not in fact estimating the term frequency, but rather the quantity $\hat{P}(Q|\mathcal{L}) \cdot N_{\mathcal{L}}$ which is useful for ranking, since $P(\mathcal{L}|Q) \propto \hat{P}(Q|\mathcal{L}) \cdot N_{\mathcal{L}}$.

[3] A grapheme to phoneme system converts a word's orthographic representation into one or more hypotheses of its phonemic pronunciation.

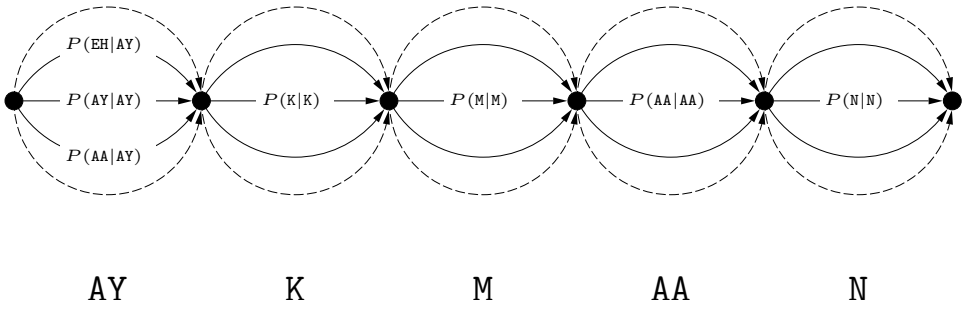


Figure 3.1: A degraded query lattice produced in CMQD query degradation, for the phonemes from query *Eichmann*. Parameter $P(\texttt{AY}|\texttt{AY})$ is the phoneme confusion probability and gives the conditional probability of phoneme AY being hypothesized by the recognizer when the true (reference) phoneme is AY. Dashed lines are place holders for the many possible arcs not shown.

sequences $\mathcal{H}$ that are most likely to have been recognized, under the model. In the machine translation literature, this process is commonly referred to as *decoding*.

We now turn to the modeling of query degradations H given a phoneme sequence Q, $P(H|Q)$. First, we consider a simple baseline approach in Section 3.3.1, borrowing from the ideas in [48, 15]. Then, in Section 3.3.2, we propose a more powerful technique, using state-of-the-art machine translation methods to hypothesize our degradations.

3.3.1 Query Degradation by Phone Confusion Matrices

In [48], phoneme confusion matrices created by aligning hypothesized and reference phoneme transcripts were used to weight edit costs for a minimum-edit distance based search in a one-best phoneme transcript. In [15], phoneme lattices were used, although with *ad hoc* edit costs and without efficient indexing. In this work, we do not want to linearly scan each phoneme lattice for our query's phoneme sequence, preferring instead to look up sequences in the inverted indices from Chapter 2.

Our baseline approach is similar to the edit-cost approach taken by [48], although we generalize it so that it may be applied within Equation 3.3 and we consider speech recognition hypotheses beyond the one-best hypothesis. First, we randomly generate

AY	K	M	AA	N
Vowel	Consonant	Semi-vowel	Vowel	Semi-vowel
Dipthong	Voiceless plosive	Nasal	Back vowel	Nasal

Figure 3.2: Three levels of annotation used by the factored phrase-based query degradation model.

N traversals of each phonemic recognition lattice. These traversals are random paths through the lattice (i.e., we start at the beginning of the lattice and move to the next node, where our choice is weighted by the outgoing arcs' probabilities). Then, we align each of these traversals with its reference transcript using a minimum-edit distance criterion. Phone confusion matrices are then tabulated from the aggregated insertion, substitution, and deletion counts across all traversals of all lattices.[4] From these confusion matrices, we compute unsmoothed estimates of $P(h|r)$, the probability of a phoneme h being hypothesized given a reference phoneme r.

Making an independence assumption, our degradation model for a query with m phonemes is then $P(H|Q) = \prod_{i=1}^{m} P(h_i|r_i)$. We efficiently compute the most probable degradations for a query Q using a lattice of possible degradations and the forward backward algorithm. Figure 3.1 shows the lattice of possible degradations for an example query. We call this approach **CMQD** (Confusion Matrix based Query Degradation).

3.3.2 Phrase-Based Statistical Query Degradation

One problem with CMQD is that we only allow insertions, deletions, and one-for-one substitutions. It may be, however, that certain pairs of phonemes are commonly hypothesized for a particular reference phoneme (in the language of statistical machine translation, we might say that we should allow some non-zero *fertility*). Secondly, there is nothing to discourage query degradations which are unlikely under an (errorful) language model—that is, degradations that are not observed in the speech hy-

[4] Alternatively, we could directly align to the hypothesis lattices.

potheses. Finally, CMQD doesn't account for similarities between phoneme classes. While some of these deficiencies could be addressed with an extension to CMQD (e.g., by expanding the degradation lattices to include language model scores), we can do better using a more powerful modeling framework. In particular, we adopt the approach of phrase-based statistical machine translation [24, 22]. This approach allows for multiple-phoneme to multiple-phoneme substitutions, as well as the soft incorporation of additional linguistic knowledge (e.g., phoneme classes). This is related to previous work allowing higher order phoneme confusions in bigram or trigram contexts [6], although they used a fuzzy edit distance measure and did not incorporate other evidence in their model (e.g., the phoneme language model score). The reader is referred to [22, 23] for detailed information about phrase-based statistical machine translation. We give a brief outline here, sufficient only to provide background for our query degradation application.

Statistical machine translation systems work by converting a source-language sentence into the most probable target-language sentence, under a model whose parameters are estimated using example sentence pairs. Phrase-based machine translation is one variant of this statistical approach, wherein multiple-word *phrases* rather than isolated words are the basic translation unit. These phrases are generally not linguistically motivated, but rather learned from co-occurrences in the paired example translation sentences. We apply the same machinery to hypothesize our pronunciation degradations, where we now translate from the "source-language" reference phoneme sequence Q to the hypothesized "target-language" phoneme sequence H.

Phrase-based translation is based on the noisy channel model, where Bayes rule is used to reformulate the translation probability for translating a reference query Q into a hypothesized phoneme sequence H as

$$\arg\max_H P(H|Q) = \arg\max_H P(Q|H)P(H).$$

Here, for example, $P(H)$ is the language model probability of a degradation H and $P(Q|H)$ is the conditional probability of the reference sequence Q given H. More generally however, we can incorporate other *feature functions* of H and Q, $h_i(H, Q)$, and with varying weights. This is implemented using a log-linear model for $P(H|Q)$, where the model covariates are the functions $h_i(H, Q)$, so that

$$P(H|Q) = \frac{1}{Z} \exp \sum_{i=1}^{n} \lambda_i h_i(H, Q)$$

The parameters λ_i are estimated by MLE and the normalizing Z need not be computed (because we will take the argmax). Example feature functions include the language model probability of the hypothesis and a hypothesis length penalty.

In addition to feature functions being defined on the surface level of the phonemes, they may also be defined on non-surface annotation levels, called *factors*.[5] In a word translation setting, the intuition is that statistics from morphological variants of a lexical form ought to contribute to statistics for other variants. For example, if we have never seen the word *houses* in language model training, but have examples of *house*, we still can expect *houses are* to be more probable than *houses fly*. In other words, factors allow us to collect improved statistics on sparse data. While sparsity might appear to be less of a problem for phoneme degradation modeling (because the token inventory is comparatively very small), we nevertheless may benefit from this approach, particularly because we expect to rely on higher order language models and because we have rather little training data: only 22,810 transcribed utterances (about 600k reference phonemes).

In our case, we use two additional annotation layers, based on a simple grouping of phonemes into broad classes. We consider the phoneme itself, the broad distinction of vowel and consonant, and a finer grained set of classes (e.g., front vowels, central vowels, voiceless and voiced fricatives). Figure 3.2 shows the three annotation layers

[5]These should not be confused with the factors from linear models, as used in Section 3.4.

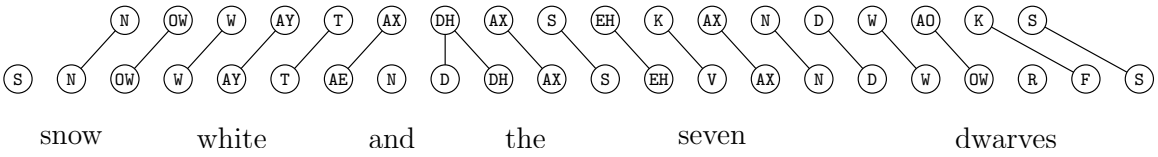


Figure 3.3: An alignment of hypothesized and reference phoneme transcripts from the multigram phoneme recognizer, for the phrase-based query degradation model.

we consider for an example reference phoneme sequence. A complete enumeration of the factors is given in Table 3.1. After mapping the reference and hypothesized phonemes to each of these additional factor levels, we train language models on each of the three factor levels of the hypothesized phonemes which are incorporated as features in the translation model.

We use the open source toolkit `Moses` [23] as our phrase-based machine translation system. We used the SRI language modeling toolkit to estimate interpolated 5-gram language models (for each factor level), and smoothed our estimates with Witten Bell discounting [61]. We used the default parameter settings for `Moses`'s training, with the exception of modifying `GIZA++`'s default maximum fertility from 10 to 4 (since we don't expect one reference phoneme to align to 10 degraded phonemes). We used default decoding settings, apart from setting the distortion penalty to prevent any reorderings (since alignments are logically constrained to never cross). For the rest of this chapter, we refer to our phrase-based query degradation model as **PBQD**. We denote the phrase-based model using factors as **PBQD-Fac**.

Figure 3.3 shows an example alignment learned for a reference and one-best phonemic transcript. The reference utterance "snow white and the seven dwarves" is recognized (approximately) as "no white a the second walks". Note that the phrase-based system is learning not only acoustically plausible confusions, but critically, also confusions arising from the phonemic recognition system's peculiar construction. For example, while `V` and `K` may not be acoustically similar, they are still confusable—within the context of `S EH`—because multigram language model data has many examples of

Level	Class	Members
1	One class per phoneme	IY IH EH AE IX UX AX UW UH AH AO AA EY AY OY AW OW P T K B D G M N NX F TH S SH HH V DH Z ZH CH JH L R Y W
2	Vowels	IY IH EH AE IX UX AX UW UH AH AO AA EY AY OY AW OW
	Consonants	P T K B D G F TH S SH HH V DH Z ZH CH JH
	Semi-vowels	M N NX L R Y W
3	Front vowels	IY IH EH AE
	Central vowels	IX UX AX
	Back vowels	UW UH AH AO AA
	Retroflexes	ER
	Dipthongs	EY AY OY AW OW
	Voiceless plosives	P T K
	Voiced plosives	B D G
	Nasals	M N NX
	Voiceless fricatives	F TH S SH HH
	Voiced fricatives	V DH Z ZH
	Affricates	CH JH
	Glides	L R Y W

Table 3.1: The inventory of phonemes broken into three factor levels for the phrase-based query degradation model from Section 3.3.2.

the word *second*. Moreover, while the word *dwarves* (D-W-OW-R-F-S) is not present in the OOD dictionary, the words *dwarf* (D-W-AO-R-F) and *dwarfed* (D-W-AO-R-F-T) are present (*N.B.*, the change of vowel from AO to OW between the OOV and in vocabulary pronunciations[6]). While CMQD would have to allow a deletion and two substitutions (without any context) to obtain the correct degradation, the phrase-based system can align the complete phrase pair from training and exploit context. Here, for example, it is highly probable that the errorfully hypothesized phonemes W AO will be followed by *K*, because of the prevalence of *walk* in language model data.

[6]This is essentially caused by difficulties in pronunciation generation. Note however, the system can not simply force *dwarves* to share common phonemes with nearby words already present in the dictionary (e.g., *dwarf, dwarfed*). Consider the various pronunciations of the 'o' in *telephone, telephonic, telephony.*

3.4 A Discriminative Approach

Our query degradation approach was principally motivated by the mismatch between a query word's lexical phoneme sequence and the speech recognition system's errorful recognition hypothesis. This mismatch was addressed by modifying Equation 3.2 to incorporate many alternative phoneme sequences, thus requiring a significant increase in the time required for each search. If however we desire faster search times, we may alternatively account for the sequence mismatch problem by making better use of the features we have for our one lexical pronunciation.

There are several reasons why we might expect Equation 3.2 to be a suboptimal estimator for term frequency in a speech utterance. First, we are severely limited in how available information may be used for prediction. We can not easily extend the model to incorporate additional information (e.g., linguistic or signal knowledge). Second, generative models are not designed to be good *discriminators* between responses. Consider, for example, that if a phoneme is systematically misrecognized, Equation 3.2 will underestimate the frequency of terms which contain it. By comparison, a discriminative model can in principle simply learn that smaller feature values may still be predictive of the term's presence. Or, we may have intuitively derived features which we expect to be reasonable frequency predictors, *provided* we can learn a suitable but unknown transformation of them. Just as a linear model might be improved by taking the logarithm of a covariate, we'd like to transform our covariates, but to also *learn* what the appropriate transformations are. A suitable framework for this type of learning, greatly simplifying the task of incorporating additional knowledge sources, is provided by Generalized Additive Models.

Generalized Additive Models [62, 17] (GAMs) are a generalization of *Generalized Linear Models* (GLMs), while GLMs are a generalization of the well known linear model. In a GLM, the distribution of an observed random variable Y_i is related to

the linear predictor η_i through a smooth monotonic *link function* g,

$$g(\mu_i) = \eta_i = \boldsymbol{X}_i\boldsymbol{\beta}.$$

Here, $\mu_i \equiv \mathbb{E}(Y_i)$, $\boldsymbol{X}_i$ is the i^{th} row of the $n \times m$ model matrix $\boldsymbol{X}$ (one set of observations corresponding to one observed y_i) and β is a vector of unknown parameters to be learned from the data. If we constrain our link function g to be the identity transformation, and assume Y_i is Normal, then our GLM reduces to a simple linear model.

Generalized additive models allow for additional model flexibility by allowing the linear predictor to now also contain learned smooth functions f_j of the covariates x_k. For example,

$$g(\mu_i) = \boldsymbol{X}_i^*\boldsymbol{\theta} + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}, x_{4i}).$$

As in a GLM, $\mu_i \equiv \mathbb{E}(Y_i)$ and Y_i belongs to the exponential family. Strictly parametric model components are still permitted, which we represent as a row of the model matrix $\boldsymbol{X}_i^*$ (with associated parameters $\boldsymbol{\theta}$).

GAMs may be thought of as GLMs where one or more covariate has been transformed by a basis expansion, $f(x) = \sum_{j=1}^{q} b_j(x)\beta_j$. Given a set of q basis functions b_j spanning a q-dimensional space of smooth transformations, we are back to the linear problem of learning coefficients β_j which "optimally" fit the data. If we knew the appropriate transformation of our covariates (say the logarithm), we could simply apply it ourselves. GAMs allow us to learn these transformations from the data, when we expect some transformation to be useful but don't know it's form *a priori*. In practice, these smooth functions may be represented in various ways (i.e., bases). In this work, we represent our smooths as thin plate regression splines [65][7]

[7]We do not expect the choice of basis to significantly effect performance, although thin plate regression splines have some attractive properties (e.g., they do not require we choose a set of knot locations for the spline).

Let Q_n be the set of all length n phoneme n-grams in a query word's phoneme sequence. We take as our basic feature $\phi(Q_n, \mathcal{L})$, the average expected count of the n-grams from Q_n in lattice $\mathcal{L}$. That is, we use Equation 2.4 to compute the expected number of occurrences of a query's subsequences in the lattice, then average this count over all n-grams of the same length.

$$\phi(Q_n, \mathcal{L}) = \frac{1}{|Q_n|} \sum_{\mathbf{q} \in Q_n} \mathbb{E}_{P_\mathcal{L}}[C(\mathbf{q})].$$

This is motivated by our desire to extract features which are independent of the phonemic length of a query word (so that we may construct length-independent models), and because we expect the distribution of expected counts for a query's phoneme subsequences to be a strong predictor of term occurrence. The mean is simply one statistic we may consider on this distribution. We may, for example, also consider the minimum expected count,

$$min.counts(Q_n) = \min_{\mathbf{q} \in Q_n} \mathbb{E}_{P_\mathcal{L}}[C(\mathbf{q})].$$

In addition to continuous valued features, we may also expect some benefit from discrete *factor* variables. Factors can be thought of as "learned intercepts" for different subsets of the response. In this work, we use this as a small-dimensional way to mitigate false alarms, although they could also be used to learn about channel, source, or other discrete features. For example, we might construct a simple model $y_i = \alpha_i + \epsilon_i$, where

$$\alpha_i = \begin{cases} a & \text{if } \phi(Q_1, \mathcal{L}) \leq \tau \\ b & \text{otherwise} \end{cases}$$

The intuition is that we may want to penalize lattices if their average expected count of phoneme unigrams is particularly small. One strategy is to choose τ to give an equal miss and false alarm probability on our training data, i.e., an Equal Error Rate

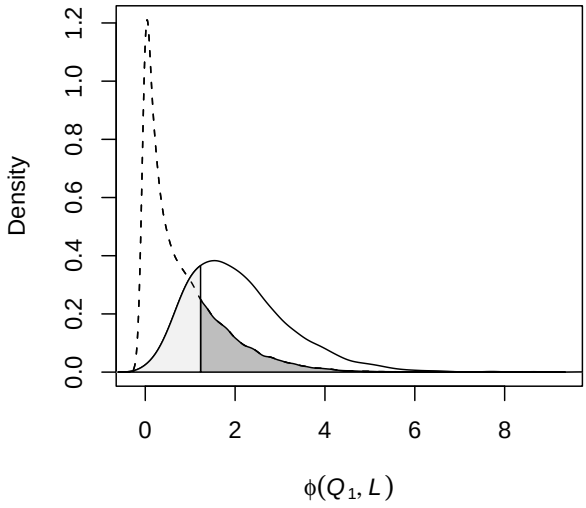


Figure 3.4: Thresholding for EER. The distribution of $\phi(Q_1, \mathcal{L})$ for absent (dotted) and present (solid) terms is shown.

(EER). Alternatively, we may choose τ such that we let in the same *proportion* of utterances as the true proportion from training. We call this thresholding P-cut.

The complete form of our new term frequency model is then

$$\begin{aligned} \mathbb{E}(tf_i) &= \beta_0 + \underbrace{\widehat{tf}_G(Q, \mathcal{L})}_{\text{offset}} + \underbrace{\alpha_i + \lambda_i + \delta_i + \gamma_i}_{\text{factors}} \\ &\quad + \underbrace{f_1(\phi(Q_3, \mathcal{L})) + f_2(\phi(Q_4, \mathcal{L})) + f_3(\phi(Q_5, \mathcal{L}))}_{\text{smooth terms}} \end{aligned} \tag{3.4}$$

where $tf \sim$ Gaussian,[8] and the model factors correspond to the presence or absence of $\phi(Q_1, \mathcal{L})$ and $min.count(Q_1)$, using both EER and P-cut thresholding. To train the model, we first get actual truth values of term frequencies from counting training occurrences in manual speech transcripts. We then compute the offset values, factor levels, $\phi(Q_3, \mathcal{L})$, $\phi(Q_4, \mathcal{L})$, and $\phi(Q_5, \mathcal{L})$ for each Q in a set of training words and each $\mathcal{L}$ in the set of training lattices. Each word Q can occur in any lattice $\mathcal{L}$, and so the vast majority of training examples have a term frequency of zero. From this

[8]Another distribution may be more suitable for the response tf (e.g., Poisson or perhaps binomial), in which case a link function other than the identity transformation would be used. We found that a Gaussian response simplified computation, although this should be revisited in future work.

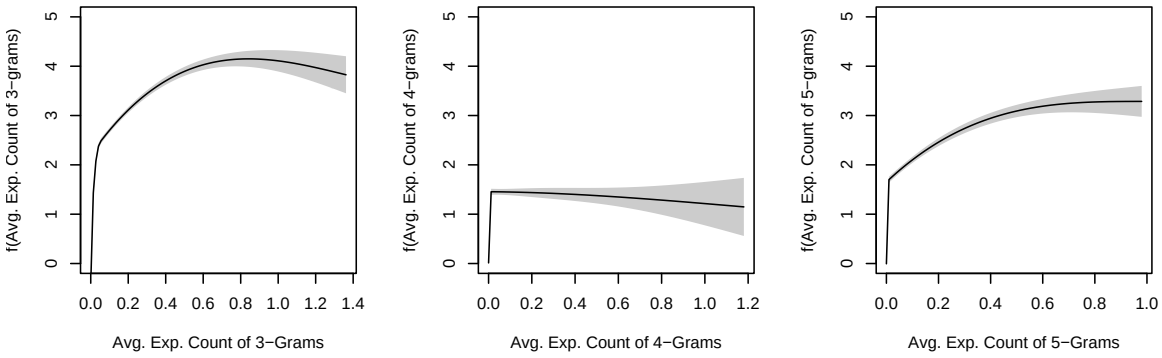


Figure 3.5: Three learned smooth functions for the generalized additive model in Equation 3.4. Shaded regions denote ±2 standard error for the smooth estimation procedure.

collected data, the model parameters are learned (see Section 3.5 for some additional information about training). At runtime, the predictors are again collected and the model is applied to produce our discriminative estimate of term frequency $\widehat{tf}_D(Q, \mathcal{L})$ for each term Q in lattice $\mathcal{L}$. We refer to this Discriminative Term Frequency Estimation approach as **DTFE** for the remainder of this chapter. We originally introduced this approach in [39].

Figure 3.5 shows the smooths f from Equation 3.4 learned in our experimental validation. We must cautiously interpret these smooths, particularly since the features being modeled are strongly dependent. Generally speaking, however, we see that larger average expected counts increase the model probability of a term's occurrence. It is not clear whether the flat shape of $f(\phi(Q_4, \mathcal{L})$ is an artifact of training with co-varying predictors.

3.5 Experiments

Our speech collection is a set of oral history interviews from the MALACH collection, which has been previously used for SR evaluations using one or more one-best word level transcripts [41, 38]. We ran phoneme-, multigram-, and word-based speech recognition to produce our vocabulary-independent indices, using the OOD dictionary

as discussed in Chapter 2.

Our task may be thought of as a trivial, if unrealistic, information retrieval problem (where a query is expressed as a single term and an utterance is "relevant" if it contains the term). That is, a user is looking for utterances containing the term, and we reward the system for putting the appropriate utterances at the top of a ranked list. An appropriate and commonly used measure for ranked utterance retrieval is Mean Average Precision (MAP) [40], as defined in Chapter 1. We consider this preferable to standard STD measures (such as NIST's *actual term weighted value* [14]) for two reasons. First, STD measures require locating a term with finer granularity than is useful in our SR system. Secondly, standard STD measures are computed using a fixed detection threshold, which is unnecessary and unlikely to be helpful for downstream evidence combination.

For our evaluation, we consider retrieving short utterances from seventeen fully transcribed MALACH interviews. Note, these interviews are disjoint from the *ad hoc* SR collection interviews; this is necessary so that we can fairly use our best system from these experiments for the *ad hoc* experiments reported in Chapter 4. Our query set contains all single words occurring in these interviews that are OOV with respect to our base, OOD dictionary. This gives us a total of 261 query terms for evaluation, which are listed in Table 3.2. Note, query words are also not present in the multigram training transcripts, in any language model training data, or in any transcripts used for degradation modeling.

To train each of our degradation models, we used a held out set of 22,810 utterances. For CMQD, we computed 100 random traversals on each lattice, giving us a total of 2,281,000 hypothesis and reference pairs to align for our confusion matrices. We train the DTFE model using leave-one-out cross validation (leaving out entire interviews). We used the `mgcv` package available for `R` [62, 56], which fits the model using penalized likelihood maximization (by Penalized Iteratively Reweighted Least

ACCOMONDO ANTI-SEMITIC ANTI-SEMITISM APPEL APPELL ARMBAND ARMBANDS AULTSELT AUSSENKOMMANDO BARONETS BARRACK BERGEN-BELSEN BERTA BIRKENAU BLOCKAVA BLOCKOVA BLOTCHIK BRADER BRESHINKA BRIBING BROATKAMER BROMIDE BROTHKAMER BUTCHERS CAMOUFLAGED CAREFREENESS CARPATHIAN CARPATOLIO CARPESIAN CHAVIVA CHERBOURG CHIMNEYS CHOLENT CHUBBIER CLOWNISH COLUMAY CORIAS CORSO CREMATORIA CREMATORIAS CREMATORIUM CREMATORIUMS CUTLERY CYRILLICA CZECHLAGER CZECHS DAMASK DEBRUTSUN DECONSGRABEN DEHUMANIZE DEHUMANIZED DELOUSE DEMORALIZING DOTING EICHLER EINZATSGROUPEN ELSNIG ENVIED EXCACTLY EXCURSIONS FATALIST FAVALEE FELICHTENSWAGER FLESHER FOREMEN FRECKLES FRIEDMANS FURTH GALICIA GENDARES GENTILE GENTILES GRANDCHILDREN'S GRIES GROSS-ROSEN HAGGADAH HAGGANAH HALUTZIM HANDMADE HANDKERCHIEF HASIDIM HAVERAH HEDY HEITCHU HOMOTOV HONOREES HUGARIANS INTERMARRIED ISSAC JANCOV JEANIE'S JUDA JUDENLAGER KAMPF KAPO KARISHKADANEE KARP KARPATOLIO KERCHIEFAS KIBBLES KIPPUR KISHTACHA KLIDUKESCOMER KNOWLEDGABLE KOMMANDO KORMIBISTOSH KOSHA KOSHEITSA KRAKOW KRYMANESURE LAGER LAGERS LAJA LAMPSHADES LATTA LATVIAN LEGACIES LEWIN LIBERITZ LIFESAVING LIQUIDATED MACHLINBERG MAJDANEK MALLET MALT MARLINKA MARTANG MAUTHAUSEN MCGILL MENGELE MENSTRUATE MICROPHONE'S MILDAOVA MITTEN MIZRAHI MODERNISM MOONCATCH MOSHE MOSHTAG MOSHTAGLAVER MOSHTAGRAVER MOULDING MULAMAD MUNKASOLGATA MUNKATABER MUNKATCH MUSELMANN MUZITSAH MYER NAKEDNESS NEUSTADT NIGHTGOWNS NON-JEW NON-STOP NULLUS NUMERUS OLDFASHIONED OLD-TIMERS OPTIMISTS ORTHODOXY OSETTA OSLOVAKIA OUTOF-TOWN OUTCASTS OUTERWEAR OVER-TIME PAIS PARLORS PEELINGS PERINI PIC PIOUS PLASZOW PLEASANTS POTATOS PRENSLAU PULGARIE PULGARIEI QUAGREE RABBINICAL RAKOV RAVENSBRUCK REFUGEE'S REMORTGAGED REPATRIATE RESENTS RHINESTONES ROMANIANS ROSIKA RUCHEL S-S- SAPINKA SATMAR SAVRUSH SAYLAPER SAYLAPPEL SCHLOMO SCHLUCK SCHOOLMATE SCHUL SCHWAMBURGER SCROUNGE SEBRUSH SELAPELL SEVINOBERGAMMO SHAMOS SHEIN SHOEMAKING SIDONIA SILESIA SLOVAKS SMUTCHURISH SOLED SPAGEN STECKIN SUDETEN SURLACH SURLICH SURLUSH SWALLOWS SWEETENED SZMALCONIKI TALMUDIC THERESIENSTADT TIBOR TOOTHACHE TYPHOID UKRAIN UKRAINES UKRAINIA UKRAINS UMBRAU UNEVENTFUL UNLAOUSEN UPRISAL VENICHTENLAGER WEHRMACHT WEINBERG WESTERNIZED YEARSHAVA YESHIVAS YOM ZELA ZIONISM ZIONISTIC ZIONISTS ZOMBIES

Table 3.2: The OOV terms used for the ranked utterance retrieval evaluation. Not shown are 13 personal names.

Squares [64]).

Method	Phone Source	QD Model	Query Degradations				
			0	1	5	50	500
GTFE	Phonemes	—	0.0387	—	—	—	—
	Multigrams	—	0.1258	—	—	—	—
	Words	—	0.1255	—	—	—	—
GTFE-QD	Phonemes	PBQD-Fac	—	0.0479	0.0581	0.0614	0.0612
	Multigrams	CMQD	—	0.1258	0.1272	0.1158	0.0991
	Multigrams	PBQD	—	0.1160	0.1283	0.1347	0.1317
	Multigrams	PBQD-Fac	—	0.1238	0.1399	0.1510	0.1527
	Words	PBQD-Fac	—	0.1162	0.1509	0.1787	0.1753
DTFE	Phonemes	—	0.0793	—	—	—	—
	Multigrams	—	0.1393	—	—	—	—
	Words	—	0.1637	—	—	—	—

Table 3.3: MAP results for all conditions in the ranked utterance retrieval evaluation.

3.6 Results

Table 3.3 shows results from the ranked utterance retrieval evaluation. First, we see that GTFE yields considerably higher MAP using words or multigrams than phonemes. This is almost certainly due to the considerably improved phoneme recognition afforded by longer recognition units (as summarized in Table 2.7). Secondly, as we saw in Figure 2.8, many more unique phoneme sequences typically occur in phoneme lattices than in their word or multigram counterparts. We expect this will increase the false alarm rate for the phoneme system, thus decreasing MAP. This represents a significant improvement over our state-of-the-art baseline approach, brought about by the improved indexing of Chapter 2.

Surprisingly, while the word-based recognition system achieved considerably lower phoneme error rates than the multigram system (Cf. Table 2.7), word-based GTFE was in fact indistinguishable from GTFE using multigrams. We speculate that this is because the GTFE method, as it is essentially a language modeling approach, is sensitive to data sparsity and requires appropriate smoothing. Because multigram

lattices incorporate smaller recognition units, which are not constrained to be English words, they naturally produce smoother phoneme language models than a word-based system. On the other hand, the multigram system is also not statistically significantly better than the word-based GTFE, suggesting this may be a promising area for future investigation.

Query degradation (GTFE-QD) appears to help all systems with respect to their generative baseline (GTFE). This agrees with our intuition that, for ranked utterance retrieval, low MAP on OOV terms is predominately driven by low recall.[9] Note that, at one degradation, CMQD has the same MAP as GTFE, since the most probable degradation under CMQD is almost always the reference phoneme sequence. Because the CMQD model can easily hypothesize implausible degradations, we see the MAP increases modestly with a few degradations, but then MAP decreases. In contrast, the MAP of the phrase-based system (PBQD) increases through to 500 query degradations using multigrams. The phonemic system appears to achieve its peak MAP with fewer degradations, but also has a considerably lower best value.

The non-factored phrase-based system PBQD achieves a peak MAP considerably larger than the peak CMQD approach. And, likewise, using additional factor levels (PBQD-Fac) also considerably improves performance. Note especially that, using multiple factor levels, we not only achieve a higher MAP, but also a higher MAP when only a few degradations are possible. This represents a significant improvement over the state-of-the-art generative baseline, both when the index is constructed using conventional phoneme unigram recognition and when using our improved recognition units from Chapter 2.

Regarding the discriminative method (DTFE), we see an appreciable improvement over the generative baseline, and approximately equal performance to using just a few

[9]Naturally, the motivation for vocabulary-independent term detection is that it will improve word recall. We note however that the preferred operating point in the tradeoff between precision and recall will be task specific. For example, it is known that precision errors become increasingly important as collection size grows [49].

	CMQD		Phrase-based	
Depth	Pronunciation	Weight	Pronunciation	Weight
1	M-EH-NX-EY-L-EH	60.45	M-EH-N-T-AX-L	44.07
2	M-EH-NX-EY-L	12.07	M-EH-N-T-AX-L-AA-T	17.86
3	M-NX-EY-L-EH	12.07	AH-AH-AH-AH-M-EH-N-T-AX-L	14.31
4	M-EH-NX-EY-EH	8.78	M-EH-N-DH-EY-L-EH	13.98
5	M-EH-NX-L-EH	6.63	M-EH-N-T-AX-L-IY	9.77

Table 3.4: The top five degradations and associated probabilities using the CMQD and PBQD-Fac models, for the term *Mengele* using multigram indexing.

query degradations from our best query degradation model (GTFE-QD/PBQD-Fac). This suggests that, in applications where disk access times are of concern, DTFE has a particularly strong advantage: it is able to significantly improve ranking with respect to the generative baseline, without requiring additional index lookups from pronunciation variants. This suggests a combined approach, ranking by the expected value of our discriminative estimator with respect to the degradation distribution. We leave this for future work.

Table 3.4 shows example degradations using both the CMQD and PBQD-Fac degradation models for multigrams. The query word is *Mengele*. Notice that CMQD hypothesizes degradations that are near (in an edit distance sense) to the reference pronunciation (M-EH-NX-EY-L-EH), while the phrase-based system tends to hypothesize degradations that sound like commonly occurring words (*mental, meant a lot, men they. . . , mentally*). In this case, the lexical phoneme sequence does not occur in the PBQD-Fac degradations until degradation nineteen. Also note, that because deleting EH has the same cost irrespective of context for CMQD, both CMQD degradations 2 and 3 are given the same pronunciation weight. In this case, CMQD performs considerably better, achieving an average precision of 17.07, while the phrase-based system obtains only 3.00. This suggests that, for this example, the phrase-based language model is exerting too much influence on the degradations, which is likely to increase the incidence of false alarms. One solution, for future work, might be

to incorporate a false alarm model (e.g., down-weighting putative occurrences which look suspiciously like non-query words). Secondly, we might consider training the degradation model in a discriminative framework (e.g., training to optimize a measure that will penalize degradations which cause false alarms, even if they are good candidates from the perspective of MLE).

3.7 Chapter Summary

Our goal in this chapter was to rank utterances by our confidence that they contained a previously unseen query word. We introduced several new approaches to this problem, which were principally motivated by the mismatch between the query's phonemes and the recognition phoneme sequences due to errorful speech indices and human variability. These systems were constructed and evaluated using phoneme-, multigram-, and word-based indexing from Chapter 2, and significant improvements in MAP for each indexing system were achieved over a state-of-the-art baseline technique.

The main goal of this work, however, is not to find isolated occurrences of words, but rather to find segments of speech which satisfy an information need—even when the topics contain OOV words. Now that we have improved methods for finding utterances containing OOV words, we are closer to this goal. In Chapter 4, we use our improved ranked utterance retrieval systems for this purpose. Namely, we combine our term frequency estimates from LVCSR and our vocabulary-independent methods for improved *ad hoc* speech retrieval.

In the following chapter, for ranked utterance retrieval we use the generative model with fifty query degradations hypothesized using the factored phrase-based model. We consider this approach using indices constructed with both word and multigram LVCSR and refer to this as simply the Ranked Utterance Retrieval (**RUR**) system.

Chapter 4

Combining Evidence for Ad Hoc Speech Retrieval

4.1 Introduction

In Chapter 3, we used our subword indices from Chapter 2 to rank utterances by our confidence that they contained an out-of-vocabulary word. However, a word's presence can not guarantee a segment is relevant to an information *topic*—even if the topic may be expressed using that term.

For words that can be anticipated by a word-based LVCSR dictionary, there is little reason to use subword representations alone for retrieval. First, we can benefit from a great deal of research which has already been invested in word-based SR systems [16, 47, 41, 4, 30] and word-based LVCSR [33, 43].[1] Secondly, for words *within* an LVCSR system's dictionary, recent research suggests that LVCSR systems are considerably better than vocabulary-independent systems at detecting spoken terms [14]. However, because OOV terms tend to be among the most informative terms in a topic's query specification, vocabulary-independent systems have also been considered for *ad hoc* SR [35, 52, 67]. Unfortunately, these vocabulary-independent SR results have often been difficult to interpret because of small, synthetic or proprietary test collections or because they do not incorporate human assessments of relevance in their evaluation (e.g., a segment is deemed "relevant" if it contains a word or, more often, if it was staged using a prompt corresponding to the topic). The lack of a substantial and realistic test collection has tended to focus speech retrieval research on detecting term occurrences rather than retrieving *informative* speech segments. Previous work in combining LVCSR and vocabulary-independent SR systems has focused

[1] These few references pertain to the LVCSR system we use in this work. See [21] for a general introduction to LVCSR.

primarily on this term detection task [27, 29]. Very little previous work has attempted to combine LVCSR and vocabulary-independent techniques for *ad hoc* SR [20]. The state-of-the-art approach to *ad hoc* SR is to adapt the decoding dictionary first (what we call the DA approach). When costs prohibit topic adaptation or when the topic is not known (e.g., with open domain problems such as podcasts), the state-of-the-art approach is to simply use a large but non-adapted dictionary (what we call the OOD approach). If the system is specifically geared towards handling the OOV problem, the state of the art is to either fall back to the vocabulary-independent system when a query word is OOV (what we call the *backoff* approach) or to combine evidence by linearly combining normalized scores or augmenting the word level index, as in [20]. Alternatively, a state-of-the-art approach to SR which may also address OOV words is to expand the documents or queries using blind relevance feedback [51]. For example, we might expand a query with words that collocate strongly with query words on a side text collection. We do not focus on this approach however, both because of the attendant difficulties with document expansion (e.g., determining a suitable side collection) and because we expect expansion techniques and the methods of this dissertation to work well in combination. Accordingly, we do not consider expansion techniques further in this dissertation, greatly simplifying our evaluation and analysis.

Naturally, we'd like to instead combine the strengths of both OOD LVCSR and vocabulary-independent term detection for SR. We present a simple model for this evidence combination, in which we learn monotonically increasing transformations of each system's retrieval scores which may then be easily combined for segment ranking. This work differs from previous SR combination efforts [20, 18, 60] in several important ways. First, we validate our method on a comparatively large collection of spontaneous speech. Second, we utilize one of our improved ranked utterance retrieval methods from Chapter 3. Third, we learn a transformation of our retrieval scores to predict relevance, rather than simply thresholding confidence values for

augmenting an index or combining scores via arbitrary normalizations. Finally, and most importantly, we find that our combination of evidence produces a new ranking which is significantly better than either ranking alone.

This chapter is organized as follows. First, in Section 4.2 we introduce the collection and task we use for our experiments. In Section 4.3, we introduce our word-based SR systems. In Section 4.4, we present the ranked utterance retrieval systems which we utilize for *ad hoc* SR. Then, we discuss in Section 4.5 how to combine our LVCSR-based results with results from our vocabulary-independent systems. We present our results in Section 4.6 and conclude with remarks in Section 4.7.

4.2 Speech Collection and Task

Our SR collection is a collection of 272 MALACH interviews, used previously by the Cross Language Evaluation Forum's cross-language speech retrieval (CLEF CL-SR) track [41, 37, 59]. We present a brief overview of the collection here, while the reader is referred to [36] for further information. Note that the interviews used for LVCSR training and testing are disjoint from the SR interview collection. Likewise, there are no interviews in common between the testing collection from Chapter 3 and the SR interview collection.

As previously noted, the speech audio was automatically segmented into short utterances for the purpose of running automatic speech recognition. Longer, topically coherent *segments* of the speech were also defined by professional indexers. For comparison, an average utterance is 6.75 seconds (with a standard deviation of 4.16), while segments average 3.45 minutes (with a standard deviation of 137.9 seconds). This distinction is important because it is these *segments* that we retrieve in our SR experiments, not the utterances as in Chapter 3. There are 8,104 such segments (corresponding to roughly 589 hours of conversational speech) and 96 assessed topics.

Following standard TREC conventions, the MALACH queries are fully specified

as a title, description, and narrative. Several examples are given in Table 2.3. Recall, as we saw in Section 2.3.1, the title field may be thought of as a short representation of the query, akin to what a user might enter into a web search engine. Naturally, then, the title field is expected to contain the most discriminative terms. The description field provides a further specification of the topic, akin to what a user might tell a reference librarian when first seeking help in retrieving topic information.

We evaluate on multiple topic sets. To allow comparison with previously published results, we run on 33 evaluation topics used in CLEF's 2006 and 2007 CL-SR track [37, 41]. In those 33 topics however, there are only 10 and 12 topics having OOV terms in their title and description fields respectively. When reporting on this topic set, we average across the complete CLEF topic set—including the topics without OOV terms. This indicates roughly how much MAP may be lost due to OOV query words in a random selection of topics. For the remainder of this chapter, we refer to this topic set as the **CLEF Topics**.

We also run on the 38 topics from the complete topic set having at least one OOV word in their title and the 49 topics containing at least one OOV word in their title or description. This topic set is denoted as **OOV Topics** for the remainder of this chapter. Table 4.1 lists the title, description, and narrative words which are OOV in the complete set of topics with respect to the OOD dictionary. Note that words in the topics may also be OOV with respect to the DA dictionary. That is to say, expanding a decoding dictionary with words from 200 hours of speech transcripts will of course not guarantee that every topic word is within the vocabulary. Table 4.2 lists the smaller set of words which are OOV with respect to the DA dictionary.

4.2.1 Evaluation Measures

As in Section 3.5, we evaluate our system using mean average precision, although now a speech segment is only considered relevant if it has been judged *topically relevant*

Field	OOV Terms
T	AEG AFFILIATIONS ANTISEMTISM (sic) BIRKENAU BUCHENWALD COUNTERFEITING COURIERS DEHUMANIZATION DP EICHMANN EUGENICS FARBEN GENTILE GIS HASIDISM IG INTERNMENT JOSEF KAPOS KINDERTRANSPORT LIBERATORS MISCHLINGE MUSELMAN MUSELMANNER NEUENGAMME OBSERVANCE OBSERVANCES POSTWAR SACHSENHAUSEN SCHLEICH SHOAH SINTI SOBIBOR SONDERKOMMANDO STRACHOWICE TELEFUNKEN TEREZIN VARIAN WALLENBERG ZIONISM
D	AEG AGALSTERHAUSEN ANTISEMITISM APPROACHABILITY ARANDORA BIRKENAU BUCHENWALD BURIALS CHAPLAINS COUNTERFEITING COURIERS CREMATORIUM DP DUNERA EICHMANN EMIGRATION ESP ETC EUGENICS EXPROPRIATED EXPROPRIATION FARBEN FOEHRENWALD GENTILE HASHOMER HASIDIM HATZAIR IG III INTERNED INTERNMENT INTERWAR JOSEF KAPOS KINDERTRANSPORT LIBERATORS MARSEILLE MINISTERSHIP MISCHLINGE MUSELMAN NEUENGAMME OBSERVANCES OSWEGO POSTWAR REBUFF SACHSENHAUSEN SCHLEICH SHOAH SINTI SOBIBOR SONDERKOMMANDO SS STARACHOWICE TELEFUNKEN TEREZIN UNQUESTIONING VARIAN WALLENBERG WINDSHEIM WWII
N	AEG AGLASTERHAUSEN ALLOCATIONS ATZERET AUSSENLAGER BIALA BIELSKO BITTERFIELD BORBEK BRAUNSCHWEIG BRETON BRICKWORKS BUCHENWALD BULGARIAN BUNA BUSSERASCH CATEGORIZATION CHACHMEI CHAGALL CHAPLAINS CHEVRA COMMEMORATIONS COUNTERFEITING COURTSHIPS CREMATORIA CREMATORIUM DEHUMANIZATION DENIGRATION DISROBING DP EICHMANN EMIGRATE ERFURT ETC EUGENICS FOEHRENWALD GALICIAN GENTILE GERMANIA GETTO GI GIS HASIDIC HASIDISM HEIL HEINKEL HEVRA HEYDEBRECK HIDER HIDERS HOECHST HUELS HYDRAWERK IG INFANTICIDE INTERNED INTERNEE INTERNMENT INTERVIEWEE JEWRY KADDISHA KADISHA KAIZERWALD KAPOS KINDERTRANSPORT KIPPUR KLINKERWERK KZ LEUNA LEVERKUSEN LIBERATORS LITZMANNSTADT LUBLIN LUDWIGSHAFEN MAUTHAUSEN MENGELE MESSERSCHMITT MONOWITZ MUSELMAN OBSERVANCE OBSERVANCES OHRDRUF OPPAU ORANIENBURG OSCHERSLEBEN OSRAM PLUNDERING POSTWAR REICHENBACH REPRESSIONS REPRISAL RIGA SACHSENHAUSEN SCHKOPAU SHABBOS SHAPIRA SHAVUOT SHEMINI SIMCHAT SINTI SOBIBOR SONDERKOMMANDO SPOTTING SS STARACHOWICE SUBSIDIARIES SUKKOT TELEFUNKEN VARIAN VICHY WALLENBERG WHOLENESS WINDSHEIM WOLFEN YOM ZIONISM

Table 4.1: Words in the CLEF CL-SR collection topics that are not contained in the OOD dictionary.

Field	OOV Terms
T	AEG AFFILIATIONS ANTISEMTISM (sic) COURIERS EUGENICS GIS IG MISCHLINGE MUSELMAN MUSELMANNER NEUENGAMME POSTWAR SCHLEICH SINTI STRACHOWICE TELEFUNKEN VARIAN
D	AEG AGALSTERHAUSEN APPROACHABILITY ARANDORA COURIERS DUNERA ESP ETC EUGENICS EXPROPRIATED EXPROPRIATION FOEHRENWALD HASHOMER HATZAIR IG III INTERWAR MINISTERSHIP MISCHLINGE MUSELMAN NEUENGAMME OSWEGO POSTWAR REBUFF SCHLEICH SINTI STARACHOWICE TELEFUNKEN UNQUESTIONING VARIAN WINDSHEIM WWII
N	AEG AGLASTERHAUSEN ALLOCATIONS ATZERET AUSSENLAGER BIELSKO BITTERFIELD BORBEK BRAUNSCHWEIG BRETON BRICKWORKS BUSSERASCH CATEGORIZATION CHACHMEI CHAGALL CHEVRA COMMEMORATIONS COURTSHIPS DENIGRATION DISROBING ERFURT ETC EUGENICS FOEHRENWALD GERMANIA GETTO GI GIS HEVRA HEYDEBRECK HIDER HIDERS HOECHST HUELS HYDRAWERK IG INFANTICIDE INTERNEE KADDISHA KADISHA KAIZERWALD KLINKERWERK KZ LEUNA LEVERKUSEN LITZMANNSTADT LUDWIGSHAFEN MESSERSCHMITT MUSELMAN OHRDRUF OPPAU ORANIENBURG OSCHERSLEBEN POSTWAR REICHENBACH REPRESSIONS SCHKOPAU SHAPIRA SHEMINI SIMCHAT SINTI SPOTTING STARACHOWICE SUBSIDIARIES SUKKOT TELEFUNKEN VARIAN WHOLENESS WINDSHEIM WOLFEN

Table 4.2: Words in the CLEF CL-SR collection topics that are not contained in the DA dictionary.

by a human assessor.

Secondly, we report the *Fraction of Recovered Mean average precision* (**FRM**), which we define as

$$FRM = \frac{MAP - MAP_{OOD}}{MAP_{DA} - MAP_{OOD}},$$

where MAP_{DA} and MAP_{OOD} are the MAPs associated with the DA and OOD word-based systems, respectively. The FRM indicates the proportion of MAP (lost because the dictionary was not adapted) which is recovered by combining the OOD word system with the vocabulary independent system's output. Note that, by definition, the DA SR system achieves an FRM of 100%, while the OOD system has an FRM of 0%.

Throughout this chapter, when we report statistically significant improvements in

MAP, we are comparing AP for paired topics using a Wilcoxon signed rank test at $\alpha = 0.05$.

4.3 Word-level SR Systems

We now present our SR approach using only hypotheses from a fixed-vocabulary LVCSR system. We construct systems using both OOD and DA LVCSR, which give lower and upper bounds respectively on the MAP attainable for each topic set. It is the results from this OOD SR system which we combine with our vocabulary-independent results. Throughout, we use the word indices presented in Chapter 2.

To rank documents using only the word counts from LVCSR, we use a vector-space model with Okapi BM25 weighting [46]. The approach defines a segment d's retrieval score (or *retrieval status value*, **RSV**) for query q as

$$s_{d,q} = \sum_{i=1}^{n} idf(q_i) \frac{(\frac{k_3+1)qf_i}{k_3+qf_i})f(q_i,d)(k_1+1)}{f(q_i,d) + k_1(1-b+b\frac{|d|}{avgdl})},$$

where the inverse document frequency (idf) is defined as

$$idf(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

N is the size of the collection, $n(q_i)$ is the document frequency for term q_i, qf_i is the frequency of term q_i in query q, $f(q_i, d)$ is the term frequency of query term q_i in document d, $|d|$ is the length of the matching document, and $avgdl$ is the average length of a document in the collection. As in previous work [38], we set the parameters to $k_1 = 1$, $k_3 = 1$, $b = 0.5$. We take as a word's term frequency, $f(q_i, d)$, the sum of the word's expected counts from all lattices within the segment. Because utterances can cross segments boundaries, we place word counts from an utterance in the segment containing the largest fraction of the utterance. For the purpose of

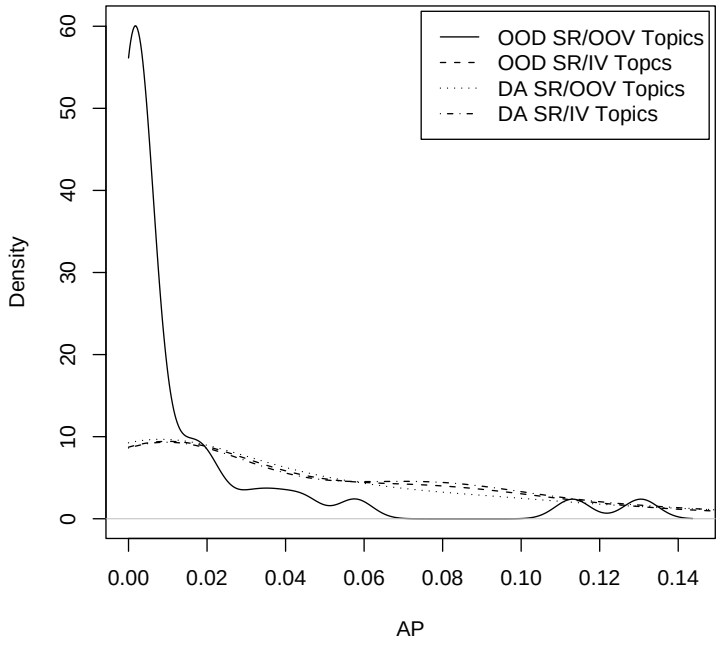


Figure 4.1: Density of AP for both the OOD and DA SR systems on 38 title queries with one or more OOV words and 58 title queries having only IV words.

computing document frequency, we define a word to be present within a segment if $f(q_i, d) \geq 0.5$.

In the results section, we refer to the word-based SR systems as **OOD BM25** or **DA BM25**, if they were constructed using OOD or DA LVCSR, respectively.

Now that we have both OOD and DA SR systems (using BM25), we can investigate how each is affected by the presence of OOV query words. To illustrate how the systems are affected differently, we ran both the OOD and DA SR systems on the complete set of 96 CLEF CL-SR topics. This complete set includes 58 completely IV title queries and 38 title queries having one or more OOV words. Figure 4.1 shows the estimated density of AP for each system on each of the OOV and IV topics sets. First, we see that the density of AP is similar for the DA system on both IV and OOV queries. Second, we see that the density of AP is similar for DA and OOD systems on IV queries. This is not surprising because the underlying LVCSR systems

are very similar. Finally, we see that the density of AP for the OOD system on OOV queries is sharply peaked near A$P = 0$ and noticeably differs from the AP densities on the other conditions. As we would expect, this confirms that the MAP loss between the DA and OOD SR systems is primarily due to queries with OOV words. To improve on these OOV queries, we incorporate additional evidence from a vocabulary-independent system.

4.4 Vocabulary-Independent Systems

We use our best multigram- and word-based ranked utterance retrieval approaches from Chapter 3. Both use the phrase-based query degradation model with factors (GTFE-QD/PBQD-Fac). We use 50 query degradations. As noted previously, we refer to this system in this chapter as simply the Ranked Utterance Retrieval (**RUR**) system.

In Chapter 3, we retrieved utterances. Our goal now, however, is to retrieve segments. Therefore, we modify GTFE-QD (Equation 3.3) to estimate term frequencies for entire segments,

$$\widehat{tf}_G(Q) = \sum_{\mathcal{L} \in \mathcal{D}} \sum_{H \in \mathcal{H}} \left[N_{\mathcal{L}} \cdot \hat{P}(H|\mathcal{L}) \right] \cdot P(H|Q), \tag{4.1}$$

where we sum the utterance-level term frequency estimates from each lattice $\mathcal{L}$ in the segment $\mathcal{D}$. As in the word-based SR systems, we consider an utterance to be part of a segment if the majority of the utterance is within the segment.

4.5 Combination Methods

There are two types of approaches for combining ranked retrieval results, data-fusion and data-merging. In *data-merging*, indices are combined first, and afterwords a single RSV is computed using the combined index. The difficulty with this approach

is how to transform the term frequency estimates from each system such that they are commensurate and, thus, combinable. As an example, in [20], scores above a threshold from a phonetic lattice scanner[2] were simply added to the index as being present words. This allowed then-state-of-the-art IR methods to be used, although the combined performance was not better than either system alone.

In *data-fusion*, separate RSVs are computed from each index before the RSVs are combined. This allows us to use strong retrieval systems as inputs, but also forces us to make simplifying assumptions for their combination (e.g., that a linear combination of RSVs is sensible after normalization). In this work, we focus on data-fusion approaches. To address the RSV transformation problem, we consider a new method which learns an appropriate normalization of the scores. First, we present several data-fusion techniques that have previously been considered.

One approach for combining ranked retrieval results is to simply linearly combine the multiple system scores for each topic and document. This approach has been extensively applied in the literature [2, 5, 42, 57] for text IR, with varying degrees of success, owing in part to the potential difficulty of normalizing scores across retrieval systems. In [20], this approach was used to combine results from a now small (20k word) LVCSR system with scores from a phone lattice scanner. Scores were normalized by the largest score for the input type. However, the combinations did not improve upon the best of the non-combined results.

More advanced score normalization methods have also been proposed for data-fusion, as in [50]. Perhaps the most successful of these is known as CombMNZ. CombMNZ has been shown to achieve strong performance and has been used in many subsequent studies [26, 34, 3, 28]. In this study, we use CombMNZ as a baseline for comparison, and following [28] and [26], compute it in the following way. First, we

[2]A phone lattice scanner considers, in turn, each phone lattice in a collection, searching for the query's phone sequence (or a nearby phone sequence). Under this approach, both space and time costs are considerable, as lattices must be stored and scanned sequentially.

normalize each score $s_{d,r}$ for segment d in ranked list r as

$$N_{d,r} = \frac{s_{d,r} - min(s_r)}{max(s_r) - min(s_r)},$$

where $max(s_r)$ and $min(s_r)$ are the maximum and minimum scores seen in the ranked list r. After normalization, the CombMNZ score for a document d is computed as

$$CombMNZ_d = \sum_{r \in \mathcal{R}} N_{d,r} \times |N_d > 0|.$$

Here, $\mathcal{R}$ is the set of ranked lists to be combined, $N_{r,d}$ is the normalized score of segment d in ranked list r, and $|N_d > 0|$ is the number of non-zero normalized scores given to d in any ranked list.

Manmatha et al. [32] showed that retrieval scores from IR systems could be modeled using a Normal distribution for relevant documents and exponential distribution for non-relevant documents. However, in their study, fusion results using this comparatively complex normalization approach achieved performance no better than the much simpler CombMNZ.

A simple rank-based fusion technique is *interleaving* [58]. In this approach, the highest ranked document from each list is taken in turn (ignoring duplicates) and placed at the top of the new, combined list. We use this as a second baseline for comparison.

4.5.1 Combining by Monotonic Transformation of RSV

We now present our combination approach. Recall, we aim to combine an OOD SR RSV from Section 4.3 and an RUR RSV from Section 4.4 to predict a new segment's probability of relevance. Suppose we had estimates for both the conditional probability of a segment's relevance given its word-based score, $P(rel|W)$, and its probability of relevance given a vocabulary independent system's score for an OOV title term

T, $P(rel|T)$. Assuming independence between W and T, we could then compute the probability of a speech segment's relevance given both W and T as

$$\begin{aligned} P(rel|W,T) &= \frac{P(W,T|rel)P(rel)}{P(W,T)} \approx \frac{P(W|rel)P(T|rel)P(rel)}{P(W,T)} \\ &= \frac{P(rel|W)P(W)}{P(rel)} \cdot \frac{P(rel|T)P(T)}{P(rel)} \cdot \frac{P(rel)}{P(W,T)} \\ &\approx \frac{P(rel|W)P(rel|T)}{P(rel)} \propto P(rel|W)P(rel|T), \end{aligned} \tag{4.2}$$

where the last relation is proportionality since we are only interested in ranking the segments and approximate equality indicates that independence was assumed.

Unfortunately, the retrieval status values obtained from our word-based SR system are not in fact probabilities of relevance. At most, we can say that, in general, a larger RSV ought to mean that a segment is more likely to be relevant. As a solution to this problem, we propose learning a smooth and monotonically increasing transformation f of the RSVs to map us from W to $P(rel|W)$. Specifically, our model is

$$\mathbb{E}(rel) = \beta_0 + f(W), \tag{4.3}$$

where f is constrained to be a smooth, monotonically increasing function and rel is binomial. As with our discriminative term frequency estimator (Equation 3.4 from Section 3.4), Equation 4.3 is an example of a generalized additive model. Note, separate models are learned for OOD SR RSVs and RUR RSVs. We represent the smooth f using a cubic smoothing spline, and monotonicity is ensured by modifying the standard quadratic programming problem for cubic smoothing splines with a set of linear constraints, as described in [63].

In general, our queries may have multiple title and description terms which are

OOV. Accordingly, we extend Equation 4.2 to

$$P(rel|W,T,D) \propto P(rel|W)^{\lambda} \left[\prod_{i=1}^{t} P(rel|T_i)\right]^{\gamma} \left[\prod_{j=1}^{d} P(rel|D_j)\right], \tag{4.4}$$

where λ, γ parametrize the contribution from each evidence source and t, d denotes the number of OOV terms from each field type (possibly zero). We refer to this approach of Combining by Monotonic Normalizing Transformations as **CMNT** for the remainder of this chapter.

Figure 4.2 shows the transformations $f(W)$ and $f(T)$ learned using Equation 4.3 on one fold of a leave-one-out cross-fold validation. On bottom, the estimated density of normalized RSVs[3] for both relevant and non-relevant segments are shown, for RSVs both from OOD SR and RUR (the density estimates are heavily smoothed for visualization purposes). As we expect, the relevant and non-relevant segments are strongly mixed while relevant segments tend to have modestly larger RSVs. We note, however, that the OOD and RUR RSVs have very different RSV distributions. On top, the probability of relevance given the RSVs (e.g., $P(rel|W)$) is shown for both the OOD SR and RUR system. Probability of relevance is constrained to increase monotonically with RSV. Note that both transformations have very different shapes. For example, the transformation learned on RUR RSVs flattens out for large RSVs (where the ratio of relevant to non-relevant RSV densities is small). For the largest normalized RSVs, we see that the probability of relevance given the OOD system's largest RSV is about twice as large as the probability of relevance given the RUR system's largest RSV. This is to be expected. First, because some topics still contain discriminative OOD words (in addition to their OOV words), the largest OOD RSVs are likely to be good discriminators for relevant segments. On the other hand, RUR is a harder task so that we would expect CMNT to have less confidence about the

[3]Mainly for plotting purposes, we normalize the RSVs by the largest RSV obtained by any segment.

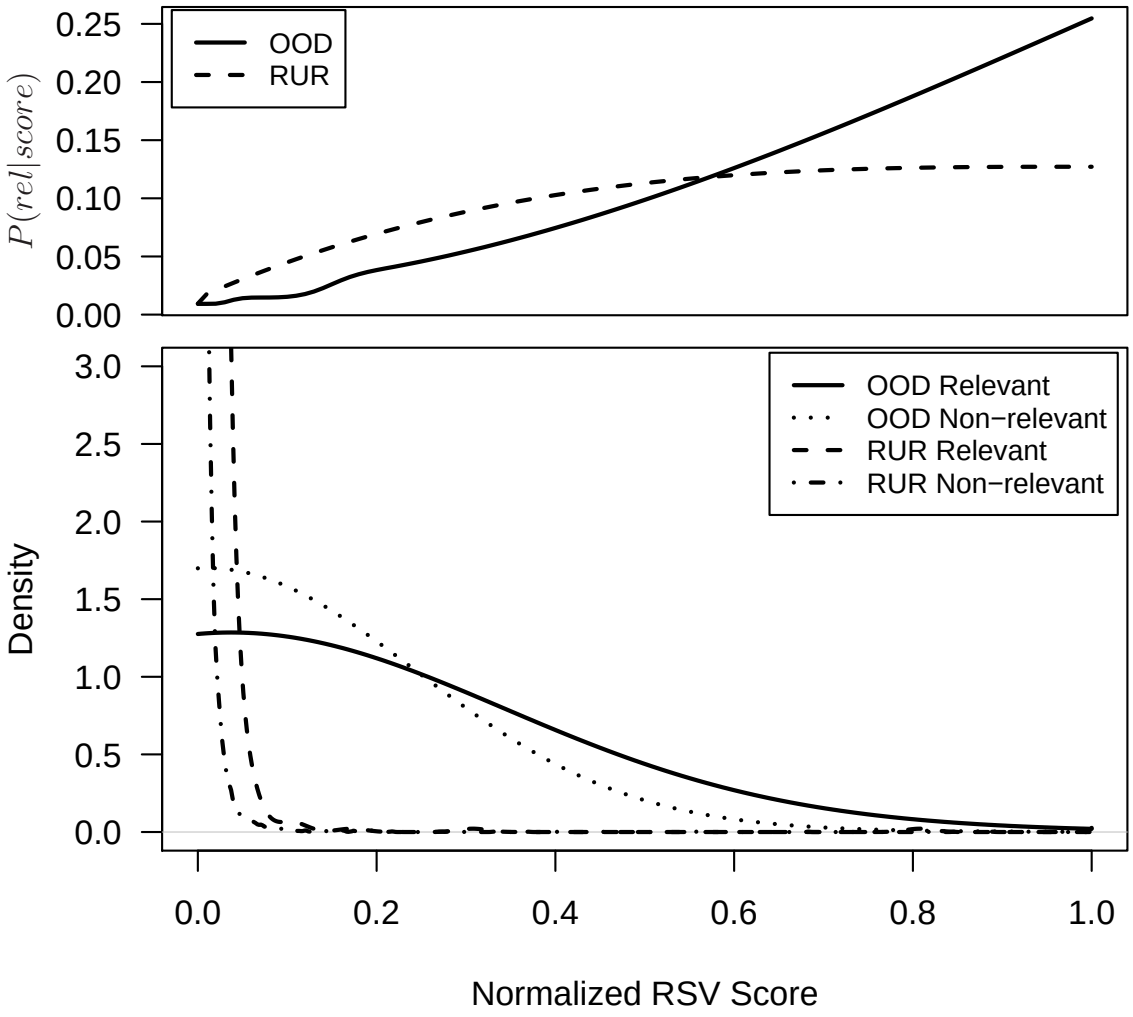


Figure 4.2: Bottom: the distribution of normalized RSVs for relevant and non-relevant training segments for both the OOD SR and RUR systems. Top: the smooth, monotonic transformations f learned via Equation 4.3.

predictive strength of RUR RSVs.

To apply Equation 4.4 we must choose values of λ, γ. Our approach is simply to choose the parameters that give the best MAP in a leave-one-out cross-validation on the training queries. We sweep over λ, γ, on the intervals $0 \leq \lambda \leq 100$ and $0 \leq \gamma \leq 100$.

4.6 Results

For our combined system, we consider as baselines both CombMNZ and interleaving, as discussed in Section 4.5. We also consider our new approach, defined in Section 4.5.1. The smooth transformations were learned using the `mgcv` package available for `R` [62], which fits the model using penalized likelihood maximization [64]. We use leave-one-out cross-validation (leaving out queries).

4.6.1 Title-Only Runs

Table 4.3 shows the title-only results from our experiments. We report on both the CLEF 2006/2007 set (having only 10 OOV queries) and the complete set of 38 topics having one or more OOV title word. For comparison, the best title-only submission at CLEF CL-SR 2006 achieved a MAP of 0.0495 using the provided, DA ASR word transcripts [37].[4] Our DA system, on the same topic set, achieves roughly the same MAP (0.0494).

First, we observe that neither CombMNZ nor interleaving is able to improve upon the best of the systems used alone (recall, the systems alone are the OOD SR and RUR systems). We suspect this is most likely because the RSVs from each system have very different distributions, so that more principled score normalization is necessary. This motivates our combination approach using monotonic normalizing transformations of the RSVs.

Suppose we rank all queries using the OOD SR system unless they have at least one OOV term, in which case we backoff and rank them only by their RUR score. Using this trivial combination approach, we see from Table 4.3 that our RUR system achieves a statistically significantly higher MAP than the OOD LVCSR system alone. We also see an improvement using the same combination approach for the CLEF

[4]The same set of topics was also used in the 2007 CLEF CL-SR, although no comparable scores (i.e., using only ASR transcripts and title queries) were reported [41].

Ranked List(s)			Combination		OOV Topics		CLEF Topics	
OOD	**RUR**	**DA**	**Rec. Unit**	**Method**	**MAP**	**FRM**	**MAP**	**FRM**
✓			N/A	no comb.	0.0158	0.0	0.0439	0.0
	✓		multigram	no comb.	0.0278	30.7	—	—
	✓		word	no comb.	0.0240	21.1	—	—
✓	✓		multigram	CombMNZ	0.0151	-1.8	0.0454	27.3
✓	✓		multigram	interleaving	0.0250	23.7	0.0464	45.5
✓	✓		multigram	backoff	—	—	0.0480	75.1
✓	✓		word	backoff	—	—	0.0492	96.3
✓	✓		multigram	CMNT	0.0382	57.5	0.0490	93.6
✓	✓		word	CMNT	0.0325	42.9	0.0503	116.1
		✓	N/A	no comb.	0.0547	100.0	0.0494	100.0

Table 4.3: Title run results from 38 topics having at least one OOV word and the results on the CLEF 2006/2007 test collection. Rec. unit indicates the recognition unit used to produce the vocabulary-independent index for RUR.

Topics set. We expect this simple approach works here because title queries tend to be short, so that an OOV query word often means the OOD SR RSV will not provide much information for ranking the segments.

Using CMNT to combine our OOD LVCSR and multigram RUR systems, we achieve an FRM of 57.5 on title queries with OOV terms. Using the word RUR system, the FRM is slightly smaller at 42.9. Both these improvements are statistically significant with respect to the OOD SR system alone (a state-of-the-art baseline which does not address the OOV problem). There is no statistically significant difference between the combined scores using multigram vs. word RUR. We also find that, on the OOV Topics set, CMNT significantly improves upon using RUR alone or combining evidence by simple normalizations (e.g., CombMNZ). These baselines are a sample of previous state-of-the-art methods for systems that do specifically address the OOV query word problem. We note, however, that the baseline combination approaches presented here benefit from the improved RUR methods of Chapter 3. Accordingly, we expect that our complete system would achieve yet larger improvements over

baseline combinations using previous state-of-the-art RUR methods.

Figure 4.3 shows the improvements obtained for each topic. On top, we see the difference in AP between the DA and OOD SR systems for each topic, sorted by the difference in AP. In the middle, with the topics in the same sort order, the difference in AP between the CMNT system (using multigram RUR) and the OOD SR system is shown. We see that the largest improvements for the CMNT system are predominately in topics with larger differences between DA and OOD MAP. On bottom, the difference in AP between the CMNT system using word RUR and the OOD SR system is shown. Again, the largest improvements are on topics with the largest loss in MAP between DA and OOD systems. This is as we would expect.

In Figure 4.3, we also see that, for a very few topics, the OOD system obtains a higher MAP than the DA system. In the most extreme case, OOD SR improved over DA SR by 0.0323 MAP, for the title query *The liberation of Buchenwald and Dachau*. One possible explanation for this may be that the terms *Buchenwald* and *Dachau* are rare—and therefore highly weighted by BM25, but they are not good discriminators for segments dealing specifically with the camps' *liberation*.

Figure 4.4 plots mean interpolated precision[5] vs. recall for title-only OOV queries, using the OOD and DA word systems, and the multigram- and word-based RUR systems alone. We see that, at low recall, the multigram RUR system yields higher precision than the word RUR system. One possibility is that this is because the degradation model for word RUR will hypothesize phoneme sequences occurring in complete words, thus increasing the false alarm rate and lowering the precision.

[5] *Interpolated precision* is commonly used when plotting precision vs. recall and is defined as the highest precision obtained for any recall level greater than or equal to the current recall level. Because moving down a ranked list and adding a non-relevant document always decreases precision while leaving recall unchanged, this interpolation is used to improve the smoothness of plots. *Mean interpolated precision* is the interpolated precision averaged across topics.

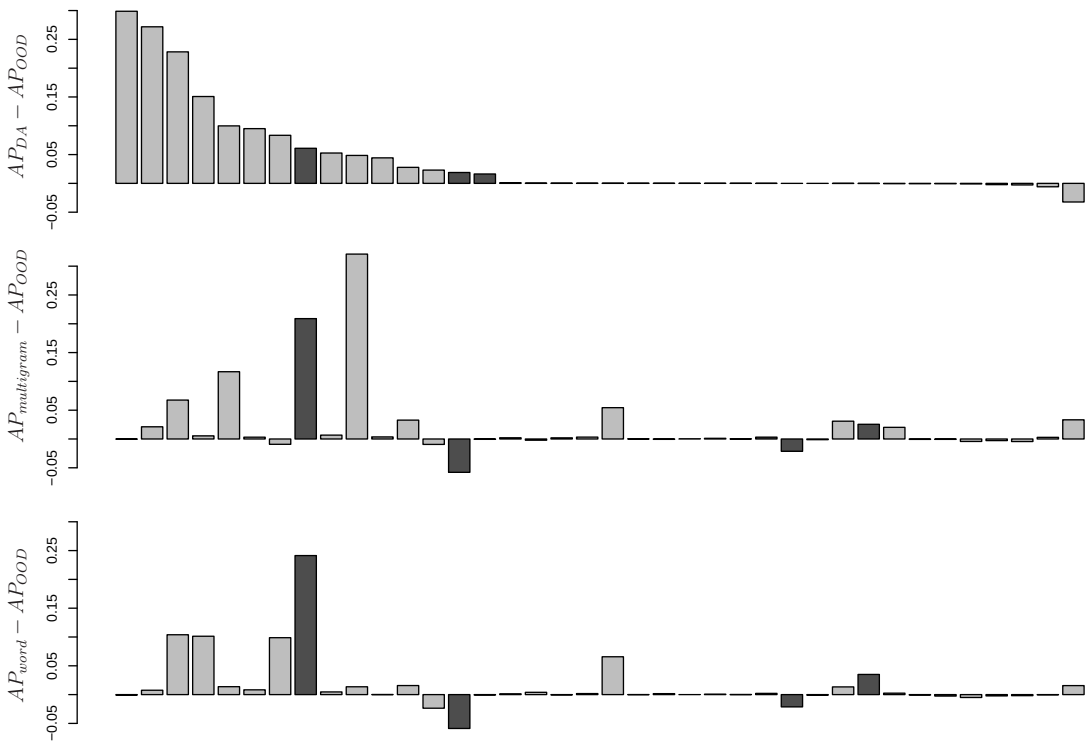


Figure 4.3: Per-query analysis for OOV T queries. The 10 Topics within the CLEF Topics set are shown in darker gray. Top: the difference in AP between the DA and OOD SR systems, where topics are sorted by size of difference. Middle: Using the same sort order, the difference in AP between the CMNT system using RUR and the OOD SR system. Bottom: Using the same sort order, the difference in AP between the CMNT system using word RUR and the OOD SR system.

4.6.1.1 Combination Parameter λ

To select our combination parameter λ for CMNT, we have used held out data in a leave-one-out cross-fold validation. We also want to know, however, how sensitive the optimal choice of λ is to different test topics. To evaluate this, we run an additional oracle experiment where we now select λ to give the best possible AP for each topic. Table 4.4 shows the mean and standard deviation of λ chosen for each topic. Also shown is the MAP attained by choosing the best possible values for λ for each topic, MAP_{best}, and the proportion of MAP_{best} obtained when λ was chosen fairly in the experiments reported above, $\frac{\text{MAP}}{\text{MAP}_{best}}$. First, we note that the standard deviation is large. The optimal setting of λ for most topics is zero, because OOD SR RSVs are

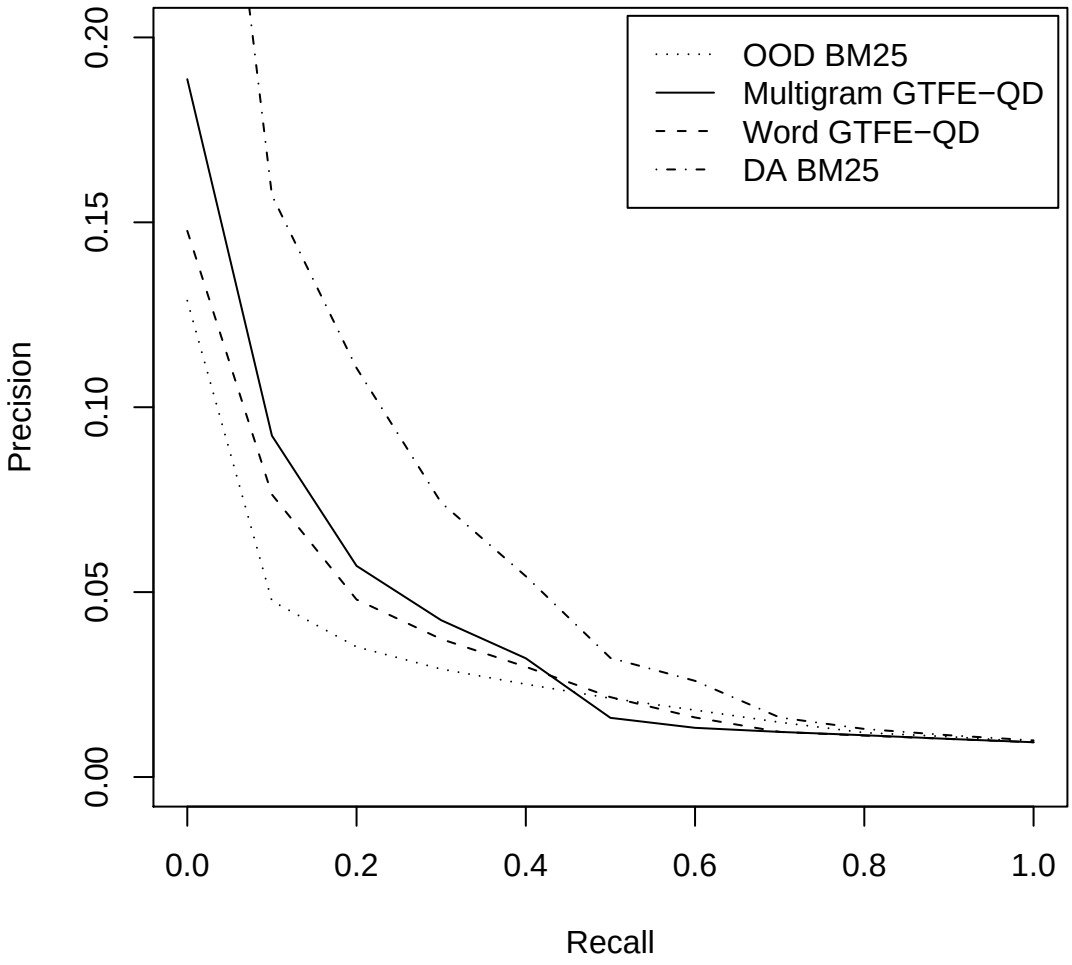


Figure 4.4: Mean interpolated precision vs. recall curves for title-only OOV queries.

Method	mean(λ)	sd(λ)	MAP_{best}	$\frac{\text{MAP}}{\text{MAP}_{best}}$
OOD BM25 + multigram RUR	12.8	29.6	0.0394	0.97
OOD BM25 + word RUR	11.2	23.3	0.0335	0.97

Table 4.4: Mean and standard deviation of CMNT parameter λ found in the oracle study for OOV title queries. MAP_{best} is the MAP obtained using the optimal settings of λ for each topic. The proportion of MAP_{best} obtained in the non-oracle evaluation, $\frac{\text{MAP}}{\text{MAP}_{best}}$, is also shown.

often of little use when the title query contains an OOV word. However, a few queries contain discriminative in-vocabulary words that cause the system to benefit from the contribution from the OOD SR system (thus increasing variance in λ). Secondly, we see that there is no substantial difference between how much combinations using multigram RUR and word RUR rely on evidence from OOD SR (i.e., there is not a substantial difference in the best average choice of λ). Finally, we see that when we chose λ in the fair evaluation reported above, we were able to obtain most (97%) of the MAP that we could have obtained if we had instead used the best possible λ for each topic. This suggests our combination approach is not particularly sensitive to choice of λ.

4.6.2 Title Plus Description Runs

Table 4.5 lists our title plus description results. Looking at CLEF Topics first, we see that our DA system achieves a MAP of 0.0501. For comparison, the best TD result from the CLEF 2006 CL-SR track (using speech recognition transcripts only) reported a MAP of 0.0381 on the same topic set [37]. For the 2007 CLEF CL-SR track, this collection was again used and the best reported TD MAP was 0.0512 [41]. We also note that MAP from TD queries on the OOV Topics set is considerably higher than the title-only counterpart. As in the title-only run, both CombMNZ and interleaving do not yield a MAP measurably higher than the best of either system alone (the apparent improvement in MAP using CombMNZ on CLEF Topics is not statistically significant).

We saw on title-only queries that a trivial backoff combination, in which we used the OOD system for all IV queries and *only* the vocabulary-independent system for OOV queries, worked better than the OOD SR system alone. Using the longer TD queries however, we see from Table 4.5 that this approach does not improve over OOD SR. This is not surprising because the TD queries have additional, useful IV

Ranked List(s)			Combination		OOV Topics		CLEF Topics	
OOD	**RUR**	**DA**	**Rec. Unit**	**Method**	**MAP**	**FRM**	**MAP**	**FRM**
✓			N/A	no comb.	0.0466	0.0	0.0374	0.0
	✓		multigram	no comb.	0.0221	-70.0	—	—
	✓		word	no comb.	0.0176	-82.9	—	—
✓	✓		multigram	CombMNZ	0.0449	-4.9	0.0392	14.2
✓	✓		multigram	interleaving	0.0365	-28.9	0.0362	-9.4
✓	✓		multigram	backoff	—	—	0.0309	-51.2
✓	✓		word	backoff	—	—	0.0333	-32.3
✓	✓		multigram	CMNT	0.0611	41.3	0.0447	57.8
✓	✓		word	CMNT	0.0541	21.4	0.0488	89.6
		✓	N/A	no comb.	0.0816	100.0	0.0501	100.0

Table 4.5: TD run results from 49 topics having at least one OOV word in their title or description field and the TD results on the CLEF 2006/2007 test collection. Rec. unit indicates the recognition unit used to produce the vocabulary-independent index for RUR.

words which are ignored when the OOD RSVs are not utilized for ranking.

Measured on OOV Topics, our CMNT approach using multigram RUR achieved a MAP of 0.0611, with an FRM of 41.3. The word-based RUR also improved MAP over the OOD word system, although it achieved only a MAP of 0.0541 and an FRM of 21.4. As before, both gains are statistically significant.

Figure 4.5 plots interpolated precision vs. recall for TD runs using the OOD, DA, multigram- and word-based RUR systems (without combination) on the OOV topic set. Now, we see that both multigram- and word-based RUR have lower precision for each recall level than the OOD or DA word systems. In the low-precision/high-recall setting, we see that the OOD and DA systems have similar precision and recall. This may be because the longer TD queries sometimes have enough redundancy to cover for OOV words. We see that multigram RUR yields a higher precision than word RUR at low recall, while their precision is approximately equal for recall greater than 0.5.

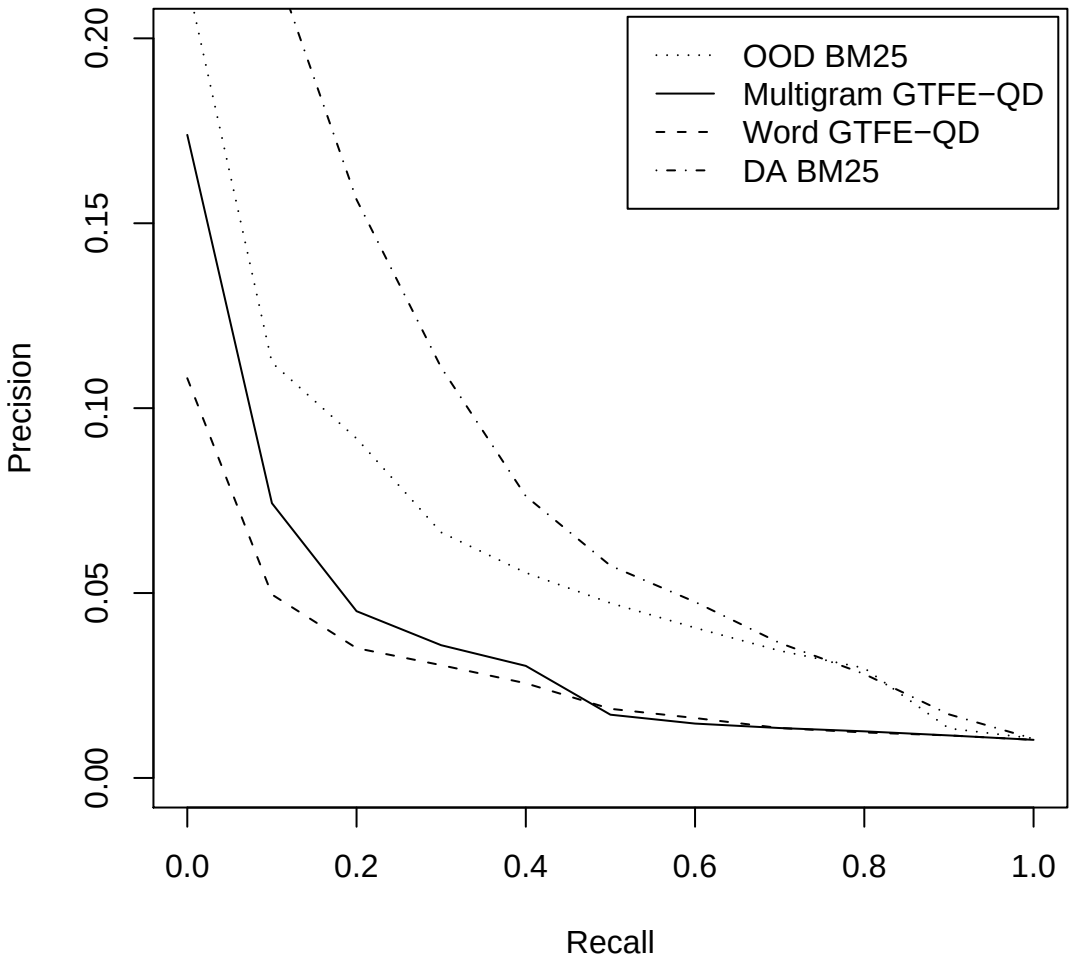


Figure 4.5: Mean interpolated precision vs. recall curves for TD OOV queries.

A potential objection to our combining evidence using Equation 4.4 (CMNT) is that OOV title and description terms may be weighted differently, while the OOD SR system does not know which terms are in the title and which are in the description. To evaluate this concern, we ran a second set of combination experiments where we constrained title and description terms to be equally weighted. That is, we restrict $\gamma = 1$. In this new condition, our CMNT system using OOD SR and multigram RUR yields a MAP of 0.0581 for TD queries on the OOV Topics. Combining with CMNT using word RUR, we obtain a MAP of 0.0522 (a non-significant improvement). Accordingly, knowing whether an OOV word is in the title or description does not appear to significantly help our system. A tentative explanation for this may be simply that, since OOV words tend to be rare and are therefore often good predictors of relevance, knowing which topic field an OOV word occurs in provides little additional information.

4.7 Chapter Summary

In this chapter, we introduced a new approach to combining search results from multiple ranked retrieval systems. In particular, we combined ranked lists of segments from an SR system using OOD LVCSR and from a vocabulary-independent RUR system. By learning a smooth, monotonically increasing normalization of each systems' retrieval status values, we produced a combined ranked list that improved, with statistical significance, MAP on an established set of SR topics. On a set of topics containing OOV title words, by combining systems, we recovered 57.5% of the MAP lost due to OOV words. On TD queries containing OOV words, our best system recovered 41.3% of the MAP lost. The MAP obtained by this combined (CMNT) system was significantly greater than previous start-of-the-art techniques: including OOD SR, combination by backoff, and combining using less-principled score normalizations.

While we improved MAP with respect to an OOD word-based SR system, a gap remains between our combined system's MAP and the MAP from the DA SR system. We attribute this primarily to two causes. First, vocabulary-independent spoken term frequency estimates are not as reliable as those from LVCSR. If we can anticipate a word when constructing the LVCSR system, it is best to include the word in the LVCSR dictionary and language model—although of course anticipating the word may not be possible. Second, our combination approach does not model dependencies between the multiple retrieval status values. For example, we weighted the contribution from each OOV term equally, even though we know that different words should have different effects on the probability of a segment's relevance. We also assumed independence in our combination approach when in fact we would expect RSVs from different systems to be highly correlated. Finally, the gains we obtained required that models be trained for predicting relevance. This required costly relevance judgments and we do not yet know how sensitive CMNT is to the amount of available training data. We expect each of these difficulties will provide a fruitful venue for our future investigations.

Chapter 5

Conclusion

The TREC spoken document retrieval track declared spoken document retrieval to be a "solved problem" and concluded that TREC efforts would be better spent on more challenging tasks [16]. This conclusion was driven by an evaluation perspective: averaging across many topics in broadcast news, the MAP fell only about 10% even with word error rates of around 40%. One prevailing explanation was that, even though many words can be misrecognized, documents contain so many other words that the system is bound to match on something else from the query [1]. Of course, we expect this to be less likely when queries are shorter or when LVCSR is OOD. When LVCSR is OOD, we might anticipate that discriminating collocations of OOV words will themselves more likely be OOV.

A second explanation for the spoken document retrieval track's success was that if a word is really important to a document, it is likely to occur more than once in that document. In this case, it was argued, it is likely to be correctly recognized at least once [1]. Of course, this can not occur if the word is OOV.

In Chapter 4 we saw that, averaged over the complete CLEF 2006/2007 CL-SR topic set, our MAP only fell about 10% by using an OOD system. But on a set of short queries with OOV words, MAP dropped about 70%. By shifting our evaluation perspective to focus on these queries with OOV words, we see that important work remains to be done in SR research.

As a solution to the OOV query word problem, we incorporated evidence from vocabulary-independent term detection to construct an *ad hoc* SR system which seamlessly handles both in- and out-of-vocabulary query words. The system uses vocabulary-independent ranked utterance retrieval in combination with word-based

SR. Many words can not be anticipated by an LVCSR system's dictionary, but without them speech retrieval accuracy (namely, mean average precision) is significantly worse. Our goal, therefore, was to recover the MAP lost because of these words.

In Chapter 2, we presented our methods for vocabulary-independent and word-based indexing. Using phoneme-, multigram-, and word-level automatic speech recognition, we indexed the utterances using the expected counts of phoneme sequences after converting each lattice of hypotheses into a corresponding phonemic lattice. We found that both the multigram and word systems significantly improved phoneme recognition accuracy and reduced the size of our indices.

Next, we used our vocabulary-independent indices in Chapter 3 to estimate the frequency of OOV terms in speech utterances. We proposed several new methods for this estimation, which were motivated by the disparity between the reference and errorfully hypothesized phoneme sequences. The first method learned alternate pronunciations or degradations from example recognition hypotheses and incorporated these variants into a new generative estimator for term frequency. A second method learned transformations of several easily computed features in a discriminative model for the same task. Both methods significantly improved ranked utterance retrieval in an experimental validation on new speech data. We found that the word and multigram systems performed best, using degradations produced via phrase-based statistical machine translation, although each system significantly improved upon a strong generative baseline.

Finally, in Chapter 4 we combined SR results from an OOD word-based system with results from our best, new ranked utterance retrieval methods. This combination approach utilized a normalization learned in an additive model, which mapped the disparate RSVs from each system into estimated probabilities of relevance that could easily be combined. Using this combination, we recovered much of the MAP lost because of OOV words. In an evaluation on a fairly large collection of spontaneous,

conversational speech, we were able to recover 57.5% of the MAP lost on short (title-only) queries and 41.3% on longer (title plus description) queries. While the absolute improvements in MAP for our combined SR systems were small, we attribute this primarily to the difficulty of our SR task and note that our system is competitive with the best reported results from previous studies on the same collection. We expect the absolute improvements would be larger on tasks with a higher DA MAP, although confirming this will require additional work on SR corpora that are not yet available.

5.1 Limitations

Our results must be evaluated within the context of several important limitations.

To create a plausible set of OOV terms for our SR trials, we used an LVCSR dictionary constructed for broadcast news and conversational telephone speech on a collection with a new topic domain, the MALACH interviews. We called the system built with this dictionary the OOD system, because it was built using no words introduced by the new topic domain. Nevertheless, our OOD system was still trained using acoustic and language model data from within the MALACH collection. This suggests that our quantitative results may not generalize to problems where there are other significant differences between the training and testing data (e.g., differences in speaker age, dialect, or channel condition).

A second limitation of this work is that many of our new methods must be trained on human annotated data. For example, our ranked utterance retrieval methods using query degradations required transcribed audio for learning the degradation model. Similarly, the combination approach from Chapter 4 was trained using held out topics with human relevance judgments. In experimental settings, both speech transcriptions and relevance judgments will be required for development and evaluation purposes, in which case cross-fold validation may be used. For practical applications,

these cost-benefit tradeoffs will assume greater importance.

We used an English-only phoneme set, even though many of the words spoken in the collection are non-English. This was probably a reasonable thing to do, given that our acoustic training data included phoneme examples from dialects similar to our test data and all query pronunciations and reference transcripts used English phonemes. Nevertheless, this should be carefully investigated in future work.

While users want relevant segments at the top of their ranked lists, the quality of the ranked list is not the only factor contributing to user satisfaction. Another important consideration is the amount of time required for search. For the most part, we have been guided by the desire to keep search costs down (e.g., we did not look back into lattices at search time). However, for some applications, some of our approaches may be too inefficient as currently formulated. For example, our query degradation approach to ranked utterance retrieval was implemented by trivially re-running the simpler term frequency estimator for each of the many (e.g., 500) degradations. Since many phoneme subsequences are common to many degradations, this is inefficient and should be revisited. Additionally, for the purpose of evaluation we assumed that the speech had been manually segmented into topically coherent segments. How to segments, or even whether to segment, remains an open question.

When indexing must be done as quickly as possible, we may prefer building our vocabulary-independent index of phone sequence counts by recognizing words rather than multigrams, since we can use the same word-level recognition for both OOD SR and vocabulary-independent RUR. However, more work needs to be done to improve the vocabulary-independent RUR system using word recognition (e.g., it performed worse than the multigram system using TD queries on Chapter 4's OOV topics set).

Our method for combining ranked utterance retrieval and word-based SR results made no distinction between different OOV words when predicting segment relevance. While this did not prevent the system from improving MAP in our trials (probably

because most OOV terms are rare and therefore discriminating), this is obviously less than optimal. In particular, we might imagine a case where a new word is used which is not topic dependent. For example, suppose we switched to Australian speech and found the word "crikey". In this case, the model would likely overestimate the contribution to relevance from that term because it is OOV, but not rare. More generally, we expect the transformed RSVs from our vocabulary-independent and word-based SR systems violate the independence assumption central to our combination approach. Relaxing this assumption may improve results further.

Finally, while our methods gave significant improvements on standard evaluation measures (e.g., phoneme error rate and mean average precision), we do not yet know the extent to which these improvements would be important to a human user. Studies with human subjects will be necessary before we could safely draw such conclusions.

5.2 Future Work

This research has opened up several important directions for future work.

The principle reason to use a topically OOD LVCSR system is to avoid the costs of transcribing speech and expanding the dictionary for the new domain. Using even a very conservative estimate for transcription costs (e.g., $100.00/hour of speech), these costs can quickly grow to be impractical. Transcription costs may arise for two reasons in our topic-domain switching scenario. First, of course, a OOD LVCSR system has to be constructed. While all of our OOD speech recognition systems were trained on similar amounts of acoustic data, it is not clear whether the relative performance of each of our speech recognition configurations would be maintained when using significantly less training data. For example, with very little data, we might expect a phoneme system to have better language model parameter estimates than a word system, since, unlike phonemes, most words are rare. While our focus is not on the costs of constructing the OOD system, we expect this will be an important

factor to reconsider when bringing up a new system, particularly if constrained to use only very little training data. This will be particularly important on rare languages or if only very little acoustically similar data is available (e.g., if building a speech recognition system for far-field phonographs).

Second, if we want to use a DA system for SR, we must consider annotation costs for the new domain. For example, we may require transcribed audio to train acoustic models for phonemes in new contexts or to expand the dictionary with suitable new words. This raises many questions which we must leave for future work. For example, if we can reasonably predict what words will accompany a new domain, to what extent is it sufficient to expand the dictionary without re-training acoustic models? Or, how can we find the small portion of data that will be useful for modeling the new domain, while minimizing annotation costs? Unfortunately, however, not only do we not know what *words we care about* for a new domain, we may in general not even know *a priori* what the new domain *is*. For example, we cannot possibly anticipate all topic domains and jargon for Internet podcasts. Certainly, the cost of such an effort would be enormous. Our approach to SR, by combining OOD and vocabulary-independent speech indexing, avoids these additional costs.

To conduct this research we required a complete, end-to-end *ad hoc* SR system. We benefited significantly from existing software (particularly BBN Technology's speech recognition system), but also implemented many new capabilities (e.g., multigram segmentation, our subword indexing approach, several ranked utterance retrieval methods, and our SR evidence combination models). Now that this system is in place, there are many new domains we may consider. In particular, we plan to investigate vocabulary-independent SR methods in non-English, morphologically complex languages. Secondly, we expect to consider problems with heterogeneous signal and channel types (e.g., podcasts). Both of these problems will likely warrant adaptations to our ranked utterance retrieval and evidence combination methods.

This work has established several new baselines for future studies. For example, we proposed a query degradation model built using comparatively sophisticated machine translation technology, but that system was not designed with pronunciation degradation in mind. By formulating the term detection problem in this way, additional degradation models should be easier for other researchers to investigate. Moreover, although we trained our degradation models using transcripts which did not contain the OOV words of interest, there may be applications where this is not necessary. For example, the user may want to find additional utterances containing an OOV word already located in the audio. In that case, we expect a degradation model could leverage these example hypotheses to find more, similar, utterances.

Because of this research, there are several practical questions we can ask that we could not have seen as easily before. For example, we saw that word-based LVCSR produced better phoneme error rates than multigrams, although it did not consistently improve upon multigram-based systems for downstream tasks such as ranked utterance retrieval or *ad hoc* SR. This suggests that more work needs to be done to choose appropriate units for recognition, which can combine both the strengths of words (e.g., long context) and the flexibility of multigrams. Secondly, while word-based GTFE-QD bested multigram-based GTFE-QD in our ranked utterance retrieval evaluation on a small evaluation collection, this was reversed on the much larger SR collection. This highlights the need to reduce false alarms, which become increasingly problematic on larger collections. Finally, while our method of combining vocabulary-independent and word-based SR results required an independence assumption between the RSVs, we could instead model the relevance probability given multiple RSVs jointly. For example, an additive model could still be used, but now with a multidimensional smooth taking each RSV as an argument.

We expect that the research described in this thesis will provide a solid foundation for this future work.

5.3 Implications

Significant challenges remain before speech retrieval technology can become as commonplace as searching the Web, but these challenges are gradually, year by year, being met. Today, we can wonder how our parents ever managed without searching text. Future generations will wonder how we managed without searching speech.

What would life be like if we could easily search all of our speech for information? Consider how much we speak and hear. We go to lectures and meetings. We hammer out grant proposals and shopping lists. We are rarely away from a phone.

We save ancient emails because we can search them. And we do. Would we search our old phone conversations? What would we record if we knew we could later search through it? Our children speak adorable errors, and we hope to remember forever. Our politicians boldly promise solutions, and they hope we will soon forget. Can we preserve all this? Do we want to?

We will find out.

Bibliography

[1] J. Allan. Perspectives on Information Retrieval and Speech. In *Information Retrieval Techniques for Speech Applications*, pages 1–10, London, UK, 2002. Springer-Verlag.

[2] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic Combination of Multiple Ranked Retrieval Systems. In *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 173–181, 1994.

[3] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, O. Frieder, and N. Goharian. Fusion of effective retrieval strategies in the same information retrieval system. *J. Am. Soc. Inf. Sci. Technol.*, 55(10):859–868, 2004.

[4] W. Byrne, D. Doermann, M. Franz, S. Gustman, J. Hajic, D. Oard, M. Picheny, J. Psutka, B. Ramabhadran, D. Soergel, T. Ward, and Wei-Jing Zhu. Automatic Recognition of Spontaneous Speech for Access to Multilingual Oral History Archives. *IEEE Transactions on Speech and Audio Processing, Special Issue on Spontaneous Speech Processing*, 12(4):420–435, July 2004.

[5] J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, Seattle, Washington, 1995. ACM Press.

[6] U. Chaudhari and M. Picheny. Improvements in phone based audio search via constrained match with high order confusion estimates. *Automatic Speech Recog-*

nition & Understanding, 2007. ASRU. IEEE Workshop on, pages 665–670, Dec. 2007.

[7] C. Chelba, J. Silva, and A. Acero. Soft indexing of speech content for search in spoken documents. *Computer Speech and Language*, 21(3):458–478, 2007.

[8] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting of the Association for Computational Linguistics*, pages 310–318, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[9] K. Darwish and W. Magdy. Error correction vs. query garbling for Arabic OCR document retrieval. *ACM Trans. Inf. Syst.*, 26(1):5, 2007.

[10] K. M. Darwish. *Probabilistic Methods for Searching OCR-Degraded Arabic Text*. PhD thesis, University of Maryland, College Park, MD, USA, 2003. Directed by Bruce Jacob and Douglas W. Oard.

[11] H. Daumé III and D. Marcu. Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[12] S. Deligne and F. Bimbot. Inference of Variable-length Acoustic Units for Continuous Speech Recognition. In *ICASSP '97: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1731–1734, Munich, Germany, 1997.

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

[14] J. Fiscus, J. Ajot, and G. Doddington. English Spoken Term Detection 2006 Results. In *Presentation at NIST's 2006 STD Eval Workshop*, 2006.

[15] J. Foote, S. Young, G. Jones, and K. Jones. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech and Language*, 11:207—224, 1997.

[16] J. Garofolo, G. Auzanne, and E. Voorhees. The TREC spoken document retrieval task: A success story. Proceedings of the TREC-9 Conference, 2000.

[17] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, 1990.

[18] D. A. James. A system for unrestricted topic retrieval from radio news broadcasts. In *ICASSP '96: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 279–282, 1996.

[19] Jean-Manuel Van Thong and Pedro J. Moreno and Beth Logan and Blair Fidler and Katrina Maffey and Matthew Moores. Speechbot: an experimental speech-based search engine for multimedia content on the web. *IEEE Trans. Multimedia*, 4:88–96, 2002.

[20] G. J. F. Jones, J. T. Foote, K. S. Jones, S. J. Young, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 30–38, New York, NY, USA, 1996. ACM.

[21] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2nd Edition, 2008.

[22] P. Koehn and H. Hoang. Factored Translation Models. In *EMNLP '07: Conference on Empirical Methods in Natural Language Processing*, June 2007.

[23] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL '07: Proceedings of the 2007 Conference of the Association for Computational Linguistics, demonstration session*, June 2007.

[24] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[25] O. Kolak. *Rapid Resource Transfer for Multilingual Natural Language Processing.* PhD thesis, University of Maryland, College Park, MD, USA, 2005. Directed by Philip Resnik.

[26] J.-H. Lee. Analyses of Multiple Evidence Combination. In *SIGIR Forum: Forum of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–276, 1997.

[27] S. Lee, K. Tanaka, and Y. Itoh. Combining Multiple Subword Representations for Open-Vocabulary Spoken Document Retrieval. In *ICASSP '05: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 505–508, March 2005.

[28] D. Lillis, F. Toolan, R. Collier, and J. Dunnion. ProbFuse: a probabilistic approach to data fusion. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 139–146, New York, NY, USA, 2006. ACM.

[29] B. Logan, P. Moreno, and O. Deshmukh. Word and sub-word indexing approaches for reducing the effects of OOV queries on spoken audio. In *HLT '02: Proceedings of the 2002 Conference on Human Language Technology*, 2002.

[30] J. Mamou, D. Carmel, and R. Hoory. Spoken document retrieval from call-center conversations. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 51–58, New York, NY, USA, 2006. ACM.

[31] J. Mamou and B. Ramabhadran. Phonetic Query Expansion for Spoken Document Retrieval. In *Interspeech '08: Conference of the International Speech Communication Association*, 2008.

[32] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–275, New York, NY, USA, 2001. ACM.

[33] S. Matsoukas, R. Prasad, S. Laxminarayan, B. Xiang, L. Nguyen, and R. Schwartz. The 2004 BBN 1xRT Recognition Systems for English Broadcast News and Conversational Telephone Speech. In *Interspeech '05: Conference of the International Speech Communication Association*, pages 1641–1644, 2005.

[34] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *CIKM '02: Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 538–548, New York, NY, USA, 2002. ACM.

[35] K. Ng and V. Zue. Subword-based approaches for spoken document retrieval. *Speech Commun.*, 32(3):157–186, 2000.

[36] D. Oard, D. Soergel, D. Doermann, X. Huang, G. Murray, J. Wang, B. Ramabhadran, M. Franz, S. Gustman, J. Mayfield, L. Kharevych, and S. Strassel.

Building an Information Retrieval Test Collection for Spontaneous Conversational Speech. In *SIGIR '04: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, U.K., July 2004. ACM.

[37] D. W. Oard, J. Wang, G. J. Jones, R. W. White, P. Pecina, D. Soergel, X. Huang, and I. Shafran. Overview of the CLEF-2006 Cross-Language Speech Retrieval Track. In *Proceedings of the CLEF 2006 Workshop on Cross-Language Information Retrieval and Evaluation*, September 2006.

[38] J. S. Olsson. Combining Speech Retrieval Results with Generalized Additive Models. In *ACL '08: Proceedings of the 2008 Conference of the Association for Computational Linguistics*, 2008.

[39] J. S. Olsson. Vocabulary Independent Discriminative Term Frequency Estimation. In *Interspeech '08: Conference of the International Speech Communication Association*, 2008.

[40] J. S. Olsson, J. Wintrode, and M. Lee. Fast Unconstrained Audio Search in Numerous Human Languages. In *ICASSP'07: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[41] P. Pecina, P. Hoffmannova, G. Jones, J. Wang, and D. W. Oard. Overview of the CLEF-2007 Cross-Language Speech Retrieval Track. In *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, September 2007.

[42] A. L. Powell, J. C. French, J. P. Callan, M. E. Connell, and C. L. Viles. The impact of database selection on distributed searching. In *Research and Development in Information Retrieval*, pages 232–239, 2000.

[43] R. Prasad, S. Matsoukas, C. Kao, J. Ma, D. Xu, T. Colthurst, O. Kimball, R. Schwartz, J. Gauvain, L. Lamel, H. Schwenk, G. Adda, and F. Lefevre. The 2004 BBN/LIMSI 20xRT English Conversational Telephone Speech Recognition System. In *Interspeech '05: Conference of the International Speech Communication Association*, 2005.

[44] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[45] S. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, pages 281–286, 1977.

[46] S. Robertson, S. Walker, S. Jones, and M. H.-B. M. Gatford. Okapi at TREC-3. In *Text REtrieval Conference*, pages 21–30, 1996.

[47] M. Saraclar and R. Sproat. Lattice-Based Search for Spoken Utterance Retrieval. In *NAACL '04: Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2004.

[48] P. Schone, P. McNamee, G. Morris, G. Ciany, and S. Lewis. Searching Conversational Telephone Speech in Any of the World's Languages, 2005.

[49] J. Shao, R. P. Yu, Q. Zhao, Y. Yan, and F. Seide. Towards Vocabulary-Independent Speech Indexing for Large-Scale Repositories. In *Interspeech '08: Conference of the International Speech Communication Association*, 2008.

[50] J. A. Shaw and E. A. Fox. Combination of Multiple Searches. In *Proceedings of the 2nd Text REtrieval Conference (TREC-2)*, 1994.

[51] A. Singhal and F. C. N. Pereira. Document Expansion for Speech Retrieval. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 34–41, 1999.

[52] O. Siohan and M. Bacchiani. Fast Vocabulary-Independent Audio Search Using Path-Based Graph Indexing. In *Interspeech '05: Conference of the International Speech Communication Association*, 2005.

[53] A. Stolcke. SRILM – an extensible language modeling toolkit. In *ICSLP '02: Proceedings of 2002 International Conference on Spoken Language Processing*, 2002.

[54] The CMU Pronouncing Dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, October 2008.

[55] USC Shoah Foundation Institute for Visual History and Education. `http://college.usc.edu/vhi/`, October 2008.

[56] W. Venables and B. Ripley. *Modern Applied Statistics with S*. Springer-Verlag, 2002.

[57] C. C. Vogt and G. W. Cottrell. Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1(3):151–173, 1999.

[58] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 500–225. National Institute of Standards and Technology, 1994.

[59] R. W. White, D. W. Oard, G. J. F. Jones, D. Soergel, and X. Huang. Overview of the CLEF-2005 Cross-Language Speech Retrieval Track. In *Proceedings of the CLEF 2005 Workshop on Cross-Language Information Retrieval and Evaluation*, pages 744–759, 2005.

[60] M. Witbrock and E. G. Hauptmann. Speech recognition and information retrieval: Experiments in retrieving spoken documents. In *In Proc. DARPA Speech Recognition Workshop 97*, 1997.

[61] I. H. Witten and T. C. Bell. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Trans. Information Theory*, 37(4):1085–1094, 1991.

[62] S. Wood. *Generalized Additive Models: An Introduction with R.* Chapman and Hall/CRC., 2006.

[63] S. N. Wood. Monotonic smoothing splines fitted by cross validation. *SIAM Journal on Scientific Computing*, 15(5):1126–1133, 1994.

[64] S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal Of The Royal Statistical Society Series B*, 62(2):413–428, 2000.

[65] S. N. Wood. Thin plate regression splines. *Journal Of The Royal Statistical Society Series B*, 65(1):95–114, 2003.

[66] P. Yu, K. Chen, C. Ma, and F. Seide. Vocabulary-Independent Indexing of Spontaneous Speech. *IEEE Transactions on Speech and Audio Processing*, 13(5):635–643, Sept. 2005.

[67] P. Yu and F. Seide. Fast Two-Stage Vocabulary-Independent Search In Spontaneous Speech. In *ICASSP '05: Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.