Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance

Matthew A. Siegler

1999 December 15

Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Thesis Committee:
Richard Stern, Chair
Tsuhan Chen
Alex Hauptmann
Michael Witbrock, Lycos Inc.

This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

Abstract

Traditionally, indexing and searching of speech content in multimedia databases have been achieved through a combination of separately constructed speech recognition and information retrieval engines. Although each technology has a legacy of research, only recently have efforts been made to study the potential suboptimality of this strategy, and none of these efforts specifically addresses the presence of uncertainty in automatically generated transcriptions.

This research develops a refinement of the most common information retrieval relevance formula, TFIDF, to incorporate uncertainty as a retrieval feature, along with a set of techniques to acquire this uncertainty from multiple hypotheses produced by existing speech recognition data structures. In the process a greater amount of evidence is extracted than is available in the most likely transcription hypothesis, and overall retrieval precision and recall are improved.

The term weighting scheme known as the inverse document frequency is shown to be a special case of the mutual information between the document set and the term, the former requiring a Boolean characterization of term occurrence information and the latter permitting fractional probabilities. The relevance between a query and document from speech recognition is then modelled as a random variable arising from the statistical nature of the speech recognition system. The statistics of this model are then derived from the word lattices and the N-Best lists from the output of the recognizer.

In analyzing the word lattices, the path probabilities for each node are summed. The relative rankings of competing terms of these summed probabilities are shown to be indicative of the probability of term occurrence. A model of this relationship is used to predict term presence and term count, reducing the degradation in retrieval quality due to speech recognition by 24%. In a separate model, the Top-N distinct text-processed hypotheses from the word lattices are used to estimate the term probability and term count. This strategy reduces the degradation in retrieval quality due to speech recognition by 63%. Experiments were performed on a standardized test of broadcast news stories that had been transcribed manually and judged against a set of natural language queries.

Table of Contents

This thesis is dedicated to my wife Erika, whose patience, confidence, and compassion made it possible.

Acknowledgements

I'd like to thank my advisor, Rich Stern, for supervising me through the process of designing and producing both Master's and Doctoral Theses. His advice over the years has proven to be invaluable to me both personally as well as professionally. My thanks also go to the remainder of the thesis committee, Alex Hauptmann, Michael Witbrock and Tsuhan Chen, for assistance in the development of a successful research plan and being especially patient with last-minute changes to the manuscript. A special thanks go to Raj Reddy, without whom the Carnegie Mellon speech group and its critical mass of researchers and students would not exist. In addition, I owe a great deal to the unsung heros of the Computer Science department, the facilities crew, who keep the machines running around the clock and quickly responded to requests for eleventh-hour file restores. My mother and father deserve a great deal of credit in encouraging me to participate in education for a full 25 years of my life. Finally, and most importantly, I thank my wife Erika whose confidence was crucial. I can say that without her it would not have been possible to put forth the effort in this thesis. Thank you, Erika.

1. Introduction

Since the beginning of recorded history there have been libraries to house an ever increasing store of information. At the end of the 20th century, those libraries had become vast collections of paper editions, stored in many thousands of shelves in the largest libraries ever known. However, as we enter into the 21st century the medium for communicating this information is changing, and both audio and video recordings begin to figure more prominently in these collections. In the same way that very large printed libraries need practical methods for searching their contents, the creation of multimedia libraries will undoubtedly require sophisticated methods.

The difference between textual and multimedia libraries is that the contents of textual libraries can be represented electronically in a database, and still remain nearly identical to their physical form. A large body of research has been spawned for automatically indexing and searching these libraries without using artificial intelligence to extract meaning or significance to the contents. Any such system is usually called an *information retrieval* system.

The story is quite different for multimedia libraries as the electronic representation for audio and video varies widely with the archive and the material itself. In addition, methods for automatically extracting multimedia content are still too early in development to be universal. However, there is one subset of multimedia that has been given a great deal of interest in the last five years, and that is the realm of spoken material.

The extraction of speech content from an audio stream, also known as *speech recognition*, has undergone a great revolution since the first attempts began in the 1960's and 1970's. Currently there are inexpensive commercial products for dictation, and more sophisticated systems for automatically extracting speech from larger collections of audio. The method of using automated speech recognition systems to extract content from audio databases, and then feeding those approximate transcriptions into an information retrieval system has recently been termed a *spoken document retrieval* system.

One of the greatest obstacles to retrieving spoken documents is that the speech recognition process is imperfect, generating on the order of a 25% error rate in the transcription. It has already been observed that traditional spoken document retrieval systems using erroneous transcriptions result in less effective retrieval performance, with this loss depending on the size of the task and the types of errors made. However, with one out of every four words incorrectly transcribed it is clear that a retrieval system that assumes perfect recognition is an unsatisfactory solution.

The goal of this work is to construct a retrieval system that can incorporate some measure of uncertainty into its decision making, and at the same time devise a method for extracting that uncertainty from the many available hypotheses in the speech recognition procedure. However a limitation that will be enforced is that these innovations should be available to a great variety of existing systems, and will therefore not use any methods that cannot be reasonably incorporated into any search system.

The remaining chapters are divided as follow:

- Chapter 2 will focus on the elementary structures of information retrieval and continuous speech recognition, and more clearly define the problem in the integration of these two systems.
- Chapter 3 describes a new method for incorporating speech recognition uncertainty into the information retrieval component.
- Chapter 4 details the selection of a database and evaluation experiments to test the new ideas
- Chapter 5 and Chapter 6 contain a set of methods and experiments for improving information retrieval using speech recognition structures known as Lattices and N-Best lists.
- Chapter 7 presents some overall conclusions from this research, and a proposal for future work.

2. Background & Motivation

The principal motivation for this research is that the performance of an information retrieval system is degraded when speech recognition transcripts are used instead of human-generated transcripts [56][44][46]. Because it is so desirable that multimedia databases containing spoken material are transcribed and indexed automatically the emphasis of this research will be on practical approaches that can be ported into other speech retrieval systems. This chapter will first describe the general structure of the two components, Continuous Speech Recognition (CSR) and Information Retrieval (IR). Then the issues brought to bear by the integration of these components will be presented.

2.1 Continuous Speech Recognition

Since the focus of this work is recovering relevant segments of spoken dialogue from a corpus of broadcast news material a method for acquiring the approximate text is required. For this work Sphinx III a powerful continuous speech recognition engine developed at Carnegie Mellon was used [4]. Sphinx III uses a combination of knowledge-based and empirically-gathered databases in order to learn how speech sounds and how to identify what is spoken. It is a statistical pattern recognition system that uses acoustic and linguistic models trained on a large corpus of speech from a variety of sources. This corpus is known as the *training set*. Sphinx takes as its input a sequence of acoustic *features* derived from the digitized audio and at its output generates a non-exhaustive set of possible word sequences and their estimated probabilities. Ideally Sphinx would produce probabilities that exactly match those found in the universe of all possible speech [23].

2.1.1 Basic Structure

To explore the space of possible word sequences Sphinx III generates a set of hypotheses in an efficient manner through the concatenation of a set of atomic-level acoustic models. These models are composed to form phonetic units, which in turn are composed to form the words, and these words are finally composed to form the hypotheses. This hierarchical structure allows a very efficient method for searching the space of possible hypotheses since at each level the sequence of possible units is restricted.

The *lexical model* is a human-generated table of possible words and their permissible phonetic sequences -- their *pronunciations*. Since there are many sequences of phonetic units that do not comprise actual words the lexical model prevents many phonetic sequences from being explored. Figure 2-1 illustrates the components of the lexical model.

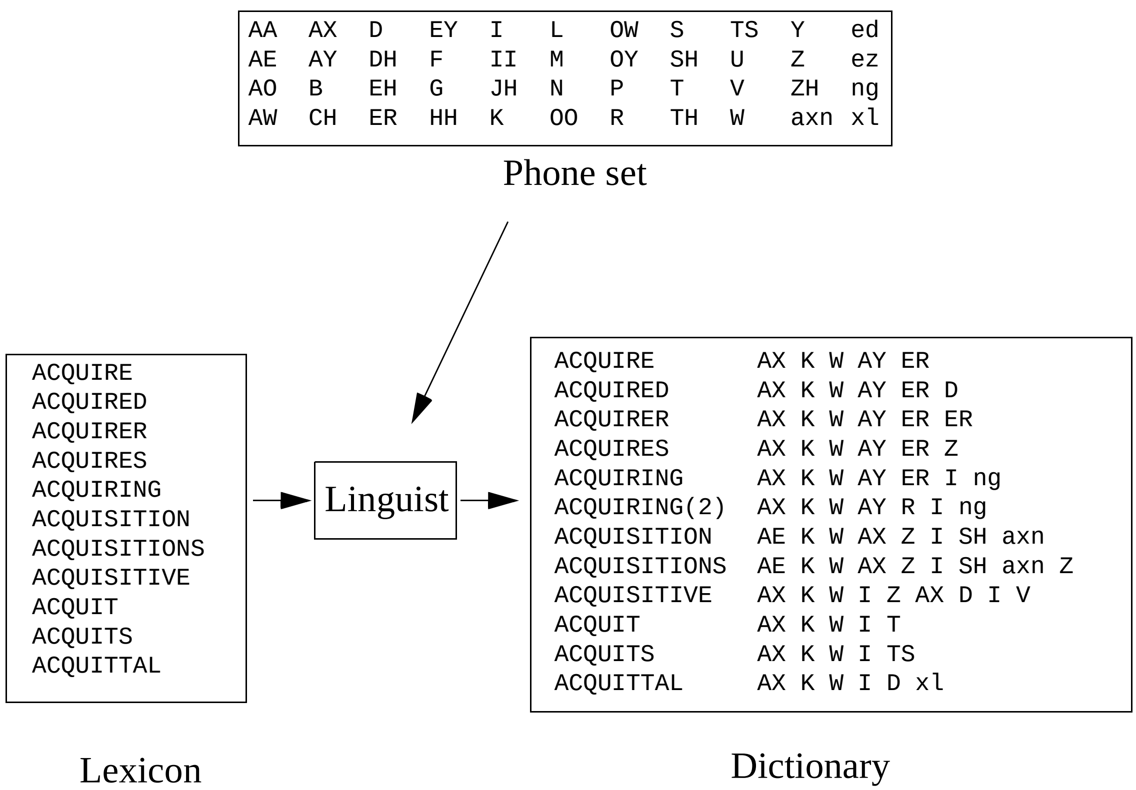


Figure 2-1 Lexical modeling in Sphinx. The phone set is built by hand as are the assignments of pronunciation to the words. The word `ACQUIRING` has two pronunciations.

The *language model* assigns an estimated probability for all possible word sequences. It is built from a very large set of text material, usually of the same sort intended to be recognized. Because there is only a relatively small amount of text material available there are many word sequences that are never observed resulting in underestimated probabilities. Trying to estimate these probabilities without actually observing them in a training set is known as the *sparse data problem*. Figure 2-2 illustrates the language modeling in Sphinx.

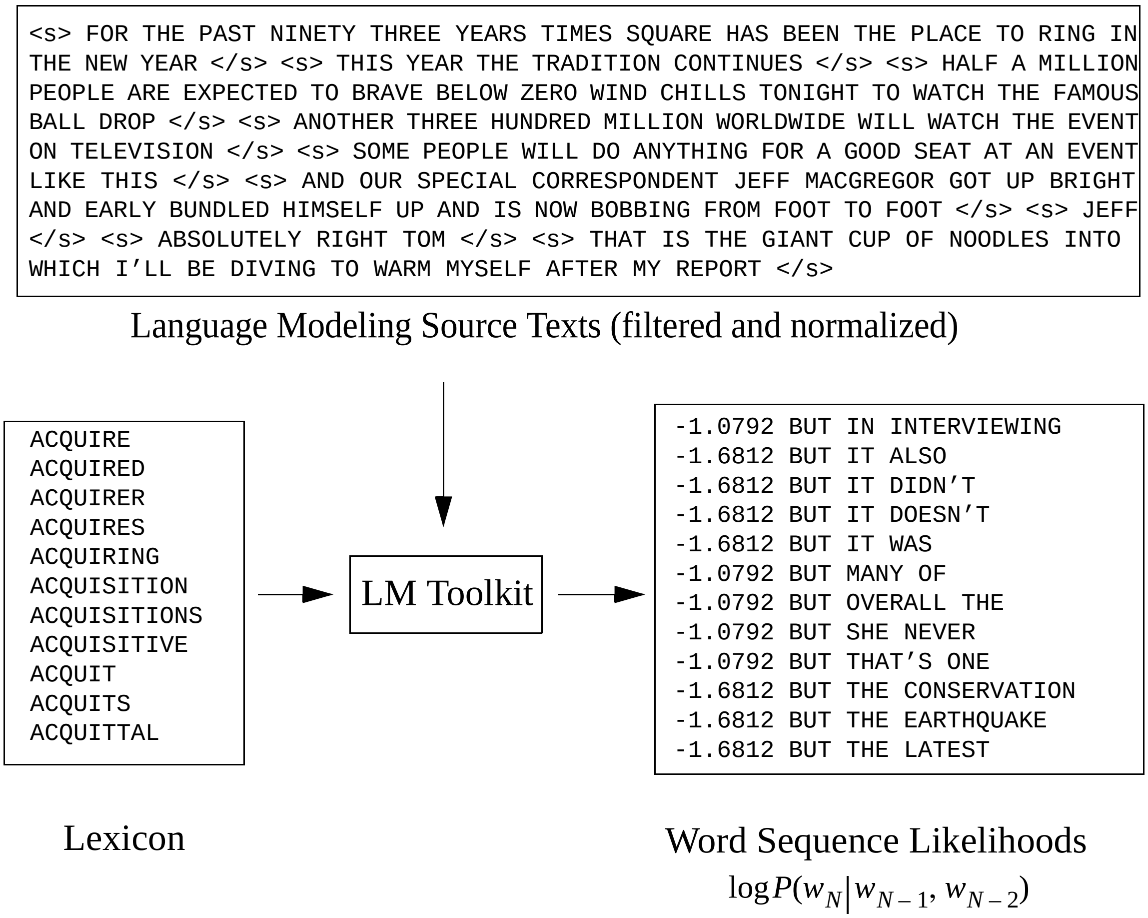


Figure 2-2 Language modeling in Sphinx. The word sequence likelihoods are usually known as trigram scores and the resultant model is called a trigram language model.

The *acoustic model* is a set of automatically generated sub-phonetic units and their statistical models for observed features built from the training set. Some form of training algorithm is used in order to maximize the accuracy of these models without compromising generality toward speech not in the training set. This is known as the *unseen data problem* and there are a variety of heuristics for minimizing its effect.

A variety of scoring mechanisms are used to estimate the posterior likelihood of the observed audio stream from a potential sequence of models by combining evidence from the language and acoustic models. In addition, another set of heuristics ranks a subset of these potential sequences of models and selects a single one as the best hypothesis. This process of searching, scoring, and ranking the space of possible word sequences given the audio stream is called *decoding* the speech from the audio.

2.1.2 Lattices

One of the important data structures that Sphinx III and most other speech recognition systems use is a *word lattice*, a very compact representation for a potentially large set of possible hypotheses in a graph. Figure 2-3 shows a simplified version of a word lattice. Each word is known as a *node* in the lattice, and each connecting branch between nodes is known as an *arc* in the lattice. The nodes have a specific pronunciation from the lexical model as there may be more than one pronunciation for a given word. In Sphinx III each arc is assigned a likelihood of transition between each word through a linear combination of the *acoustic score* and *language score* for that path. By taking the sum of these likelihoods from beginning node to ending node, over a particular *path* through the lattice, we compute the *path likelihood*.

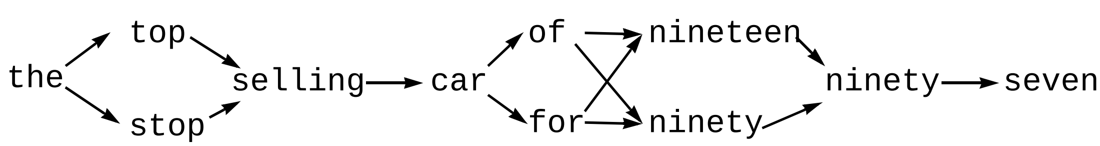


Figure 2-3 A very simplified representation of a speech recognition lattice from Sphinx III. This particular lattice has a total of 8 possible paths through it.

In addition to the identity of the word the starting and ending time for each node is specified. In Sphinx III nodes can have a variety of ending times but a fixed starting time, with the ending time inferred from the node immediately following. The lattice is generated by the decoder in recognition, and contains only a subset of all information for all of the states of the decoder during the recognition. A great deal of other information is not included in the lattice such as the starting and ending times for the phonetic components of each word.

An important thing to realize about the recognition process is that an enormous number of possible word sequences are *not* assigned probabilities. In the example only 8 paths were explored and assigned a likelihood. This vast simplification avoids a great deal of extra computation and affords us a best guess as to what speech events occurred. For the configuration of the decoder in this work the lattice contained an average of ~15 different hypothesized words for every word actually present, going as high as 1500 in some lattices. Even with lattices this full many other paths still remain unexplored. The total number of these paths exceeds D^N where N is the average number of words in a path and D is the size of the vocabulary. In this work, the size of the vocabulary exceeded 64000 and the average number of words in a path was more than 200. The process of eliminating the majority of these paths from the search is called *pruning* and without it we would spend the better part of a universe's lifetime trying to estimate the probabilities for a single speech utterance! As they stand, approximately 10^N scorable paths exist in most lattices.

2.1.3 N-Best Lists

One way of representing a small subset of all the paths contained in the lattice is with an *N-Best list*, a list containing only the N most likely paths through the lattice and their associated scores with N usually being in the neighborhood of 100 [5]. The information found in an N-Best list can be thought as a small set of the decoder's next-best guesses; its size is very small in comparison to the total number of paths. However in some circumstances it is computationally infeasible to work with the entire lattice and the use of an N-Best list is necessary. The N-Best list of length one consists of only the top scoring hypothesis: the *Top-1 hypothesis*. Figure 2-4 shows the production of an N-Best list of length 4 from the example lattice.

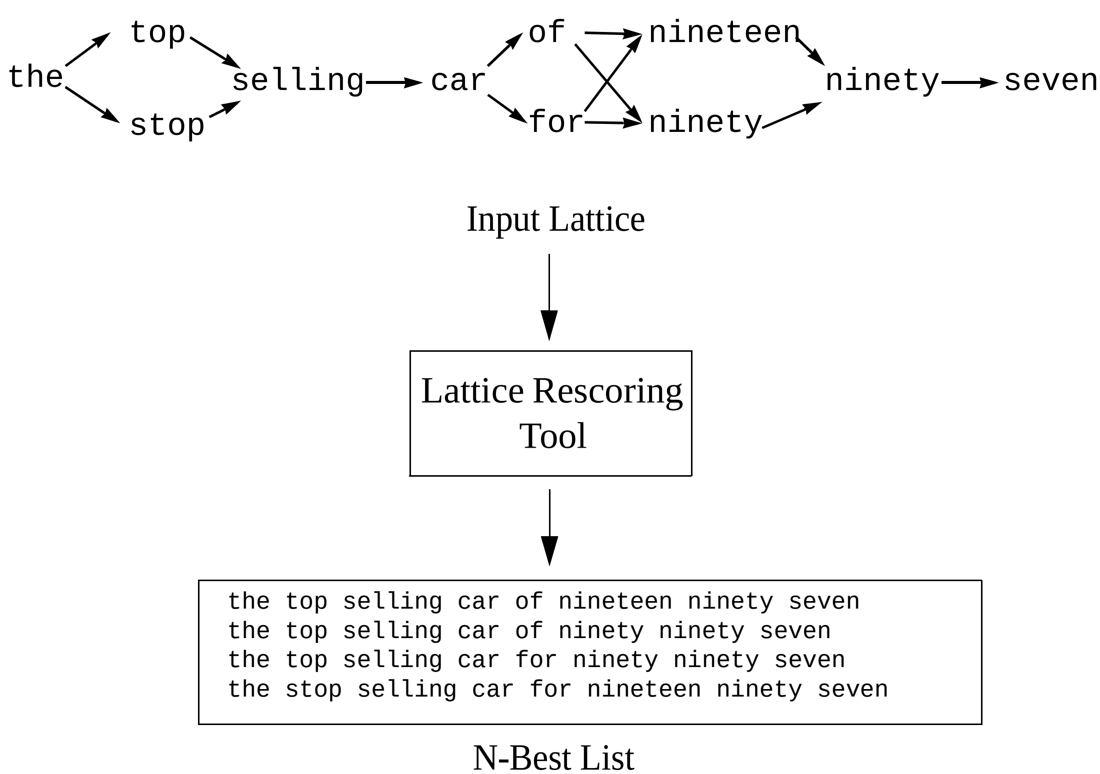


Figure 2-4 Producing the N-Best list in Sphinx III. In this particular example N=4.

2.1.4 Evaluation

The quality of continuous speech recognition is evaluated automatically by comparing the Top-1 hypothesis of a specified portion of audio with a human generated transcription (*the reference)* of the speech contained in that audio. Complicating the matter is that there are often other audible non-speech events in the utterance such as door slams, phone rings, dog barks, background music, etc. Usually these events are to be ignored and only the speech is to be recovered. The most widely used figure of merit for speech recognition is the *Word Error Rate (WER)*. A simple string alignment procedure is used to compare the text string of the hypothesis with the reference and the error is calculated as follows:

$$\text{Word Error Rate (WER)} = \frac{\text{\# Substitutions} + \text{\# Insertions} + \text{\# Deletions}}{\text{\# Correct Words}} \qquad (2\text{-}1)$$

Note that the WER can exceed 100% if there are a large number of insertions. Generally capitalization and punctuation other than the use of the apostrophe in possessive forms and contractions are not counted.

2.2 Information Retrieval

Primarily information retrieval is concerned with the identification of information sources that are related to a user's request [36]. Information retrieval is different from information extraction where content is derived directly from information sources, and question answering where a user's question is answered based on knowledge of information. Each of these could be likened to a human clerk, librarian, and expert. Of the three information retrieval is the easiest to generalize since knowledge of the content is unnecessary. Indeed, a clerk can find a document in an unknown language as long as the symbols in it closely match those in a question and as long as this similarity is likely to yield a relevant match. The model of IR that is used in this work is automatically retrieving *documents* that are most likely *relevant* to a user's *query* by selecting those that contain symbols that identify such *relevance*. Figure 2-5 illustrates the procedure for retrieving documents according to a user's need and presenting them in order of increasing *rank*.

Although the task of a clerk finding a relevant document is an apt metaphor when printed documents are involved, the similarity is not so clear in identifying relevant segments of speech in a large corpus of digitized audio. The problem is that the definition of a spoken document is a flexible notion. Whereas a printed document has a clear boundary, a paragraph structure and a fundamental unit of information (the word or letter), a spoken document is often a seamless part of a longer stream of speech, has incomplete sentences and contains only sounds. The task of determining where spoken documents begin and end is worthy of treatment by itself, but will not be addressed in this work. It is assumed that spoken documents are predefined, non-overlapping regions of audio that contain one or many sentences that discuss one or many topics of interest [56][44].

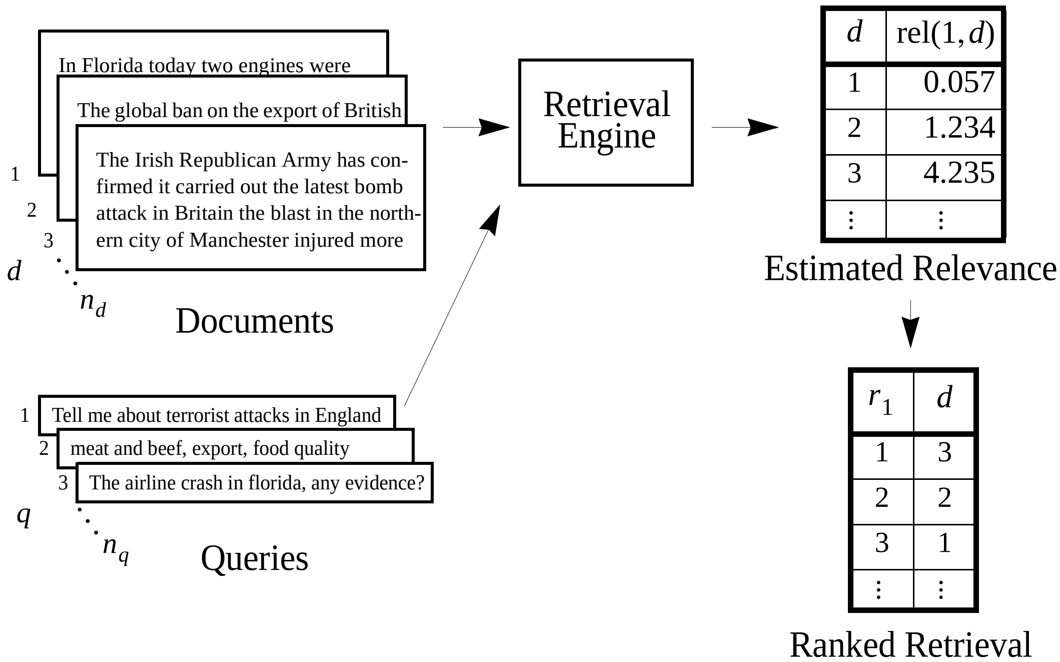


Figure 2-5 Usage model for a typical IR system, showing the estimated relevance and rankings (at various numbers of retrieved documents r_1) of the retrieved documents for query number 1. Generally there is a particular ranking assigned to every document, but only a specified number of the top r_1 documents are usually the user's concern.

Queries are a bit less troubling since users can be instructed to form their question in a particular manner by speaking clearly, typing their request carefully into a terminal, or simply selecting from a predefined set of queries. The choice of using speech, text, or a pointing device illustrates the variety of modalities that are useful, but in this work the queries are freely-formed text in the natural language of the user. The next subsections discuss the various components that go into the conversion of a document into the internal representation and the comparison of this document with the queries to predict the relevance, and the evaluation of IR performance. Figure 2-6 shows examples for each of these components.

2.2.1 Tokenization

The set of symbols that will be used to express the content of a document or query is the set of *tokens*, and the process of translating information from its native format into these tokens is called *tokenization*. There are many ways to perform this operation on text as well as on spoken documents although the most common tokens used in IR and CSR are words [36].

Original Document

The Irish Republican Army has confirmed it carried out the latest bomb attack in Britain. The blast in the northern city of Manchester injured more than two hundred people on saturday.

Tokenization and Text Processing

```
IRISH REPUBLICAN ARMI CONFIRM CARRI LATEST
BOMB ATTACK BRITAIN BLAST NORTHERN CITI
MANCHEST INJUR HUNDR PEOPL SATURDAY
```

Vector Space

17281	45661	169011	219351
20191	56281	176361	253801
36811	66051	179801	271161
39601	73401	201861	300911

Figure 2-6 Procedure for creating internal representation for a document (or query.)

It is possible to use linguistic units other than the word for automatically identifying relevance between text documents and queries especially when the languages involved do not have formal written word boundaries [41]. Even in these situations the alphabet of symbols can be determined automatically by gathering a large body of information in the particular language. The smallest unit possible in any case is an ink stroke.

This flexibility extends to a greater extent in spoken documents, where the information source is fundamentally only a sequence of analog measurements over time having no inherent boundaries or alphabet of symbols. The assumption made in this work is that there is some textual representation of the spoken document that is adequate for capturing most of the information necessary for automatically detecting relevance to a potential query. There are projects that have used units other than text words, such as phonemes, to represent the information in a spoken document [52][57]. Such an endeavor is very ambitious considering the existing body of work devoted to IR using words as tokens, but is not a focus of this work.

2.2.2 Text Processing

Neither the words contained in the queries nor the text transcription generated by the CSR engine are used unmodified in the vocabulary for the IR engine, since it is desirable to produce the smallest set of automatically derived word classes that could represent the information content of any set of words. A very common, and very simplistic way of doing this is through the elimination of very frequently used words (*stopword removal*), the stripping of word suffixes (*word stemming*.) A set of standard tools for operating on arbitrary English words exists in [35], a version of which was used in this work.

2.2.3 Vector Space Model

If the set of tokens that is to be permitted in the IR system is fixed both documents and queries can be represented as sequences of numbers indexing the *vocabulary* of terms. If the order in which these tokens appear is then discarded we can collapse the sequence into a vector containing in each dimension the count of a particular token in the document. The representation of documents and queries as vectors is a very common because it is efficient in storage and also permits the use of very high speed vector mathematics processing. Perhaps it seems unwise to discard the order of the tokens in the document since the content of is apparently lost when all the words are scrambled! The truth is that an IR system that has no understanding of semantics cannot distinguish a carefully structured sentence from one with the words in alphabetical order. Such an IR system has a *vector space* or *bag-of-words* model and is the dominant method in use today [38][36]. There are IR systems that do not use the vector space model, but since they have not yet been shown to reliably outperform their counterparts they will not be the concern of this work [32].

2.2.4 Relevance Formulae

Ultimately, the automatic estimation of document and query relevance rests upon the use of one or many relevance formulae that operate in the vector space. The variety of these formulae runs from the utterly simple Boolean match to highly specialized functions that have no basis other than empirical success. In any case the goal is to assign a real valued score to every document that expresses the estimated relevance of the document to a query, sort the documents by decreasing score, and then present them to the user. It is expected that a user will then examine the documents in this order until exhausting either their patience or the document set.

2.2.5 Evaluation

Since the relevance is only *estimated* there are circumstances where an unrelated document will be assigned a relevance score that is greater than that given to a related document. In order to determine which relevance schemes are superior in this regard many criteria have been proposed for evaluating the relative performance of different automatic IR schemes. Two of the most widely used metrics are the *precision* and *recall*, mostly for their simplicity and reliability in predicting performance across varying data sets. [36]

Suppose that there are n_q queries asking about n_d documents in the test set. For each query q and document d the IR system estimates the relevance value $\hat{\mathrm{rel}}(q, d)$ where the actual relevance is $\mathrm{rel}(q, d)$ having a value of 1 for relevant or 0 for not relevant. The number of *reference matches* is the count of documents that are actually relevant to a query q, defined $M_q = \sum_d \mathrm{rel}(q, d)$ which varies from query to query. It is assumed that there will be at least one relevant document.

The precision is defined as the proportion of retrieved documents that are relevant, and the recall is the proportion of relevant documents retrieved. The free parameter in both of these measures is the *search length*, r_q, defined as the number of retrieved documents that are kept for the query q. Depending on how the IR system is to be used the appropriate setting for r_q is very low (for high precision) or very high (for high recall), and the performance at either extreme may vary widely depending on the relevance equation. For this reason the values of precision at recall at several values of r_q are of interest since it may not be known ahead of time whether precision or recall will be important to the user.

To compute the precision and recall for a given search length, first the reference relevance values $\mathrm{rel}(q, d)$ are assigned Boolean values signifying the defined relevance between query q and document d. For each query q, $\boldsymbol{R}_q$ contains the set of documents d having reference relevance of 1:

$$\boldsymbol{R}_q = \{d | \mathrm{Rel}(q, d) = 1\}$$

The size of this set is defined as M_q, the number of relevant documents (matches) to query q

$$M_q = |\boldsymbol{R}_q|$$

The estimated relevance scores $\hat{\mathrm{rel}}(q, d)$ are real numbers, with larger values signifying higher relevance, between every document d and query q. These values are sorted and the identity of documents with the top r_q values are placed into set $\hat{\boldsymbol{R}}_q$. The following values are then defined:

$$R_{1,1}(q) = |\{d | d \in \boldsymbol{R}_q, d \in \hat{\boldsymbol{R}}_q\}|$$

$$R_{1,0}(q) = |\{d | d \in \boldsymbol{R}_q, d \notin \hat{\boldsymbol{R}}_q\}|$$

$$R_{0,1}(q) = |\{d | d \notin \boldsymbol{R}_q, d \in \hat{\boldsymbol{R}}_q\}|$$

And the precision and recall are defined

$$\mathrm{Precision}(q) = \frac{R_{1,1}(q)}{R_{1,1}(q) + R_{0,1}(q)} \tag{2-2}$$

$$\mathrm{Recall}(q) = \frac{R_{1,1}(q)}{R_{1,1}(q) + R_{1,0}(q)} \tag{2-3}$$

Figure 2-7 shows precision and recall values for both a perfect and a typical IR experiment involving one query and many thousands of documents. A very interesting thing happens when $r_q = M_q$ as both the precision and recall values are equal. If only one number were to be chosen to identify the overall performance the precision and recall at this number of retrieved documents would be the best. This has been called the *r-precision* [33]. Although this is the best single indicator of performance, it is informative to observe how precision and recall change as the number of retrieved documents changes. In addition to the r-precision two other points of interest are when $r_q = \frac{1}{2}M_q$ and $r_q = 2M_q$, and these will be used to judge the relative performance. When required the *average IR quality* will be computed as the average value of these 5 points.

2.3 Integration of CSR and IR

Recently several research groups participated in the largest standardized experiment performing information retrieval on a set of documents generated with speech recognition [44]. The task consisted of approximately 2800 documents and 23 queries, with human relevance judgements made on a large portion of the set. The documents came from approximately 75 hours of television and radio broadcast news programs, with starting and ending times for each document identified by a human. The queries were designed to test a variety of aspects of the retrieval system and cover a wide variety of topics.

Table 2-1 shows the precision and recall values for both reference texts and Top-1 speech recognition transcripts, and the relative difference in performance between them. Note that the overall average degradation, the principal criterion for assessing retrieval quality, is approximately 10%.

<table>
<tr><th rowspan="3">Document Source</th><th colspan="5">Number of retrieved documents r_q</th><th rowspan="3">Average</th></tr>
<tr><th colspan="2">$\frac{1}{2}M_q$</th><th>M_q</th><th colspan="2">$2M_q$</th></tr>
<tr><th>Precision</th><th>Recall</th><th>Precision/Recall</th><th>Precision</th><th>Recall</th></tr>
<tr><td>Reference</td><td>0.45</td><td>0.24</td><td>0.41</td><td>0.26</td><td>0.51</td><td rowspan="2">n/a</td></tr>
<tr><td>Top-1</td><td>0.40</td><td>0.21</td><td>0.36</td><td>0.24</td><td>0.48</td></tr>
<tr><td>difference (d%)</td><td>-12.0%</td><td>-11.3%</td><td>-12.5%</td><td>-6.6%</td><td>-6.6%</td><td>-9.8%</td></tr>
</table>

Table 2-1 Performance of CMU IR system in TREC-7 over varying levels of r_q for both Reference and Top-1 documents.

Although 75 hours is a rather large amount of data from the point of view of speech recognition experiment, 2800 is a very small number of documents when considered as an information retrieval experiment. To put this in perspective a 1998 speech recognition evaluation using 10 hours of speech and a text-based retrieval evaluation on 10 million documents were performed, owing to the intense computing requirements of speech recognition. Generally the speech recognition component takes 10,000 times more time to complete than the information retrieval component.

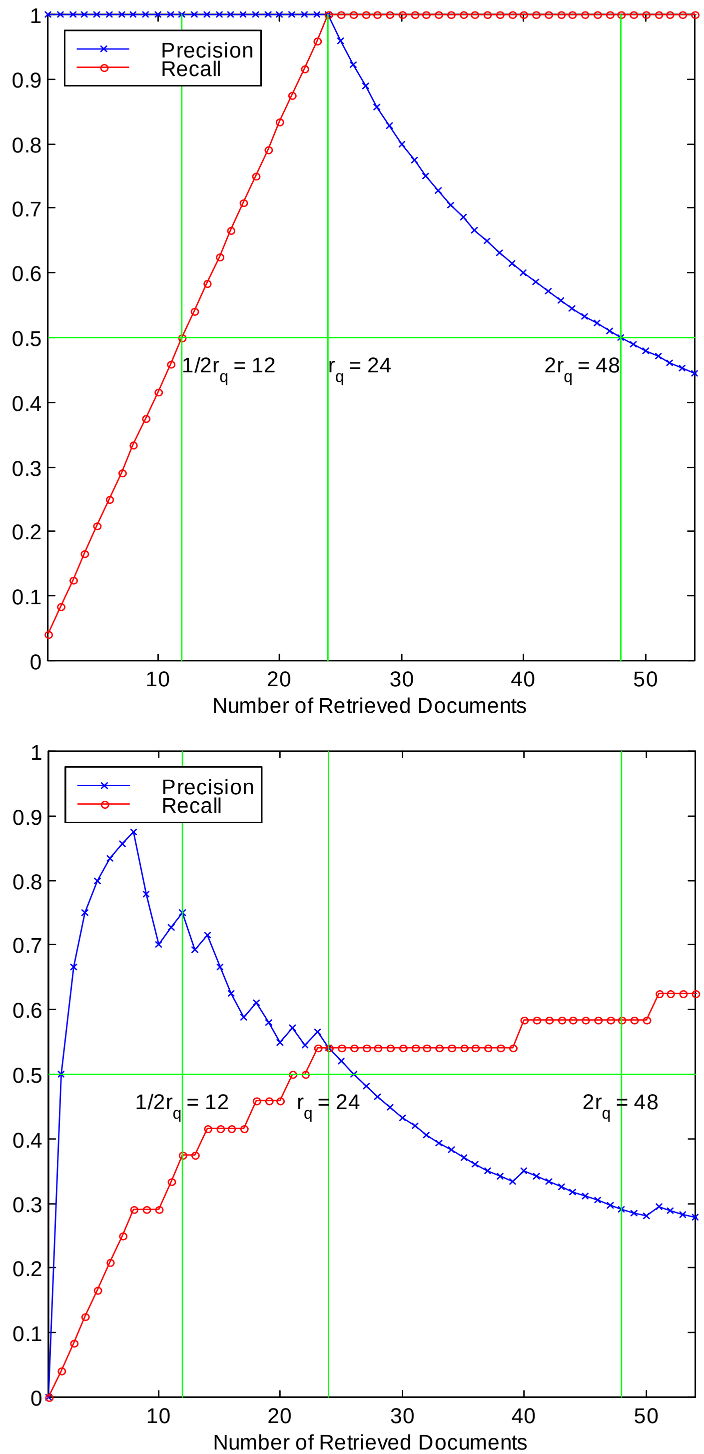


Figure 2-7 Precision and recall for an ideal, and an actual IR experiment. In the upper experiment all the relevant documents were retrieved before any irrelevant ones. In the lower experiment, a normal amount of error occurred in the process. The number of retrievals considered, r_q, varies from 1 to the number of total documents. When this number is the same as the actual number of relevant documents, $r_q = M_q$, the precision is equal to the recall.

2.4 Uniting Speech Recognition and Information Retrieval

This disparity in computing time has greatly restricted the structure of most contemporary research involving CSR and IR in combination. Careful decision making must be made since rerunning the CSR component requires another 1000 hours of processing time whereas the IR component requires only 5 minutes. The result is that most relevant research involves executing the CSR component in a fixed manner, perhaps a few times, and varying the IR component in many different ways with the goal of optimizing overall performance.

Unfortunately only a few groups have considered that using the Top-1 hypothesis from the recognition system is a very significant decision in their experimental design and have looked at ways of using the N-best lists and lattices to derive potential benefit [54][40][49]. Because major variation in the CSR component is expensive, as much generality should be extracted from the recognition process as possible so that a variety of experiments involving *both* the CSR and IR can be performed. Using the Top-1 hypothesis does not provide any room for variation in the CSR component without completely retraining the speech recognition system.

Generally speech recognition evaluations are run with the goal of reducing the overall word error rate of the Top-1 hypothesis and it is not difficult to understand why this experimental design has dominated speech recognition research for so many years. The most important reason is that the usage model for speech recognition systems has been a *dictation machine*, a fictional device that listens to a speaker and transcribes the speech perfectly. In such a scheme there is only one correct transcription and the goal is to approximate it as closely as possible. Many of the design choices in the speech recognition system were made because they helped reduce the total error of the Top-1 hypothesis. In fact, many have attempted to modify the recognition training and testing algorithms so they are explicitly refined to do this [20][21][22][23].

However, the usage model for speech recognition in a speech database retrieval experiment differs significantly in a variety of ways, most importantly in that the ultimate goal is to best identify speech relevant to a query and not to optimize transcription accuracy. Because the principal use of the CSR component in the retrieval application is to estimate the quantity of any content bearing words, any evidence in the speech recognition process that can best do so should be exploited. It is the focus of this thesis to identify and capitalize on such evidence.

3. Integration of Speech Recognition and Information Retrieval

As discussed in the previous chapter a great deal of information is discarded when only the best hypothesis is used to represent a spoken document for later retrieval. However, as described in Section 2.2, the default IR scheme is not designed to incorporate multiple estimates, but instead expects perfectly transcribed documents. In this chapter this discrepancy between the CSR and IR assumptions will be highlighted and a resolution will be proposed. The goal is to create a new structure that allows the IR component to better take advantage of the many other hypotheses found in the CSR phase, with an eye toward probabilistic models of the components that can be used to bring them together.

Although truly probabilistic relevance formulas have been developed for IR in the purely textual domain using a variety of statistical models, their current formulations expect perfect transcription and have a very high level of computational complexity [32]. This chapter will focus on an efficient and practical method for improving the information retrieval component by incorporating measures of uncertainty in the relevance equation parameters that can be derived from the speech recognition procedure.

3.1 Existing Model

Perhaps the most common way of computing the relevance between a query and a document is known as *TFIDF*, or Term Frequency, Inverse Document Frequency, the names of the two important features in the relevance formula [38]. As mentioned in the previous chapter the implementation of the TFIDF relevance formula is within a vector space representation of the documents, queries, and estimated relevances.

3.1.1 Vector Space Formulation of Relevance

Computation of relevance between queries and documents is a straightforward matrix operation, with the various components comprising the features used in the relevance formula:

n_v is the number of words in the vocabulary, n_q and n_d are the number of queries and documents.

Matrix $\boldsymbol{Q}$ is $n_v \times n_q$, containing values proportional to the count of term v in query q

Matrix $\boldsymbol{D}$ is $n_v \times n_d$, containing values proportional to the count of term v in document d

Matrices $\boldsymbol{W}_q$ and $\boldsymbol{W}_d$ are $n_v \times n_v$ diagonal matrices containing the relative weight (term significance) of each term in the vocabulary in the space of the queries and documents respectively.

Matrices $\boldsymbol{N}_q$ and $\boldsymbol{N}_d$ are $n_q \times n_q$ and $n_d \times n_d$ diagonal matrices containing a normalization constant for each query and document respectively to compensate for length.

The relevance $\boldsymbol{R}$ is an $n_q \times n_d$ matrix containing the relevance score between each query and document pair. It is computed as:

$$\boldsymbol{R} = \boldsymbol{N}_q \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{W}_q \boldsymbol{W}_d \boldsymbol{D} \boldsymbol{N}_d \tag{3-1}$$

The methods for computing the matrices $\boldsymbol{Q}$, $\boldsymbol{D}$, $\boldsymbol{W}_q$, $\boldsymbol{W}_d$, $\boldsymbol{N}_q$, and $\boldsymbol{N}_d$ from observed queries and documents are what distinguish the various relevance formulae.

3.1.2 Baseline Implementation: TFIDF Relevance

The queries are collected in $b_{q,v}$, the count of term v in query q with v varying from 1 to n_v and q varying from 1 to n_q. The document *term counts* are $c_{d,v}$, the number of occurrences of term v in document d with v varying from 1 to n_v and d varying from 1 to n_d. The *document length* is defined l_d for each document, with α being a constant exponent used to compress the dynamic range of the parameters.

$$l_d = \left(\sum_v c_{d,v}{}^{\alpha} \right)^{1/\alpha}$$

In this work, the value of α was set to 3, based on previous success with this value in [53][54][55]. The *term presence* $i_{d,v}$ is a Boolean value indicating the occurrence of term v in document d. The value of idf_v, the *inverse document frequency (IDF),* is computed

$$\mathrm{idf}_v = \log_2\left(\frac{1}{n_d} \sum_d i_{d,v} \right) = \log_2(n_d) - \log_2\left(\sum_d i_{d,v} \right)$$

The TFIDF relevance is defined as

$$\hat{\mathrm{rel}}(q, d) = \frac{1}{l_d}\left(\sum_v (b_{q,v})(c_{d,v})(\mathrm{idf}_v) \right) \tag{3-2}$$

Using vector space notation.

$$\boldsymbol{D} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & \cdots \end{bmatrix} \quad \boldsymbol{N}_d = \begin{bmatrix} 1/l_1 & 0 & 0 \\ 0 & 1/l_2 & 0 \\ 0 & 0 & \ldots \end{bmatrix} \quad \boldsymbol{W}_d = \begin{bmatrix} \mathrm{idf}_1 & 0 & 0 \\ 0 & \mathrm{idf}_2 & 0 \\ 0 & 0 & \ldots \end{bmatrix}$$

$$\boldsymbol{Q} = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & \cdots \end{bmatrix} \qquad \boldsymbol{N}_q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \ldots \end{bmatrix} \qquad \boldsymbol{W}_q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \ldots \end{bmatrix}$$

3.1.3 Problems with the Existing Model

Although combining the Top-1 hypothesis with the TFIDF relevance weighting appears to be innocuous, in fact there are some important contradicting assumptions in the two structures. By choosing to keep only the Top-1 hypothesis the assumption is that we only want to know the most likely word sequence. That means even if the two best hypotheses are almost equally likely but differ slightly in their content, only the better one is kept. However, the usage model detailed in Section 2.2 suggests that we would like to identify the maximum amount of content by involving many of the speech hypotheses and not just the most likely one.

The other contradiction is that the TFIDF-based IR system requires the use of word counts, which tacitly assumes that we have perfect knowledge of the document contents. However, we know for a fact that the speech recognition generates a hypothesis from which only an estimate of those word counts can be obtained. The critical issue is that the word counts derived from the hypothesis are not fixed numbers but actually a set of random variables representing word counts. As a result, we desire to revise the TFIDF formula so that hypotheses with higher or lower probability of being correct are regarded as containing greater or lesser evidence in establishing document to query relevance.

3.2 Improving on the TFIDF Formula

If we look more closely at the TFIDF relevance equation, we can observe it is composed of a combination of several features: term counts $\boldsymbol{D}$ or $c_{d,v}$, document weights $\boldsymbol{N}_d$ or l_d, and term significance $\boldsymbol{W}_v$ or idf_v. Further decomposition of these variables reveals that these features are in fact functions of only the two features, term count $c_{d,v}$, and term presence $i_{d,v}$. Although fixed values for these parameters are appropriate for text documents, it is necessary to change their representation to be consistent with estimated hypotheses and explore ways of extracting these representations from the speech recognition data structures.

A speech recognizer may be thought of as a machine that takes the correct transcription and substitutes, deletes, and inserts errors in a truly probabilistic fashion, and generates a set of hypotheses that are subject to this model. Developing such a model of the entire transcribed utterance would be very difficult if not impossible. If instead of the transcription we reduce the input and output to just the features we use for the retrieval system, we can then consider the hypothesized term count $\hat{c}_{d,v}$ and hypothesized term presence $\hat{i}_{d,v}$ as specific values drawn from random variables. These new random variables for term count and term presence are defined as $\hat{C}_{d,v}$ and $\hat{I}_{d,v}$ respectively.

For each document and vocabulary word, there are probability distributions $P_{\hat{C}_{d,v}}(c)$ and $P_{\hat{I}_{d,v}}(i)$ that result from the application of speech recognition in lieu of the correct transcription, and these distributions have the specific parameters $c_{d,v}$, $i_{d,v}$, and a global set of parameters controlling their shape. We can now consider the output of the speech recognition process as an ensemble of independent experiments executed with this probability model, where the event is the production of a vector of $\hat{c}_{d,v}$ and $\hat{i}_{d,v}$ driven by the underlying probability distributions $P_{\hat{C}_{d,v}}(c)$ and $P_{\hat{I}_{d,v}}(i)$.

Suppose now that we desire to use these probability distributions directly instead of just the specific values $\hat{c}_{d,v}$ and $\hat{i}_{d,v}$ from the Top-1 hypothesis. We will need to do two things in order to accomplish this: (1) find a way of estimating the probability distributions of the speech recognition process and (2) develop a relevance equation that can incorporate these probabilities.

3.3 CSR as a Probabilistic Machine

Although it would be convenient to derive the term count and term presence distributions explicitly from a large corpus of training material, we unfortunately find that they vary uniquely for each word for each utterance in the set. The consequence is that if we never witness a word in the training set, we will be unable to construct even an approximate probability distribution if it occurs in the test set. One important goal of this work is to be able to estimate the controlling parameters $c_{d,v}$ and $i_{d,v}$ by analyzing the lattices and the N-best lists. These techniques will be discussed in Chapter 5 and Chapter 6. As mentioned earlier in Section 3.1, the three components of the relevance function are term count, document length, and term significance. The effect of CSR on these parameters for retrieval will be observed in the next sections.

3.3.1 Term Count

Figure 3-1 shows histograms of $\hat{c}_{d,v}$ from the Top-1 hypothesis for various values of $c_{d,v}$. It is straightforward to deduce the maximum likelihood estimate or the expected value from these probability distributions, the latter shown in Table 3-1. Note that the estimates of term count from the Top-1 hypotheses are consistently lower than the reference values, and that the distribution is asymmetric. Because of this structure, it may be fruitless to accurately model the exact nature of the errors caused by CSR and more appropriate to use an approximation. In a later section we will note that the mean and variance of the hypothesized term count are sufficient for predicting the distribution of the relevance.

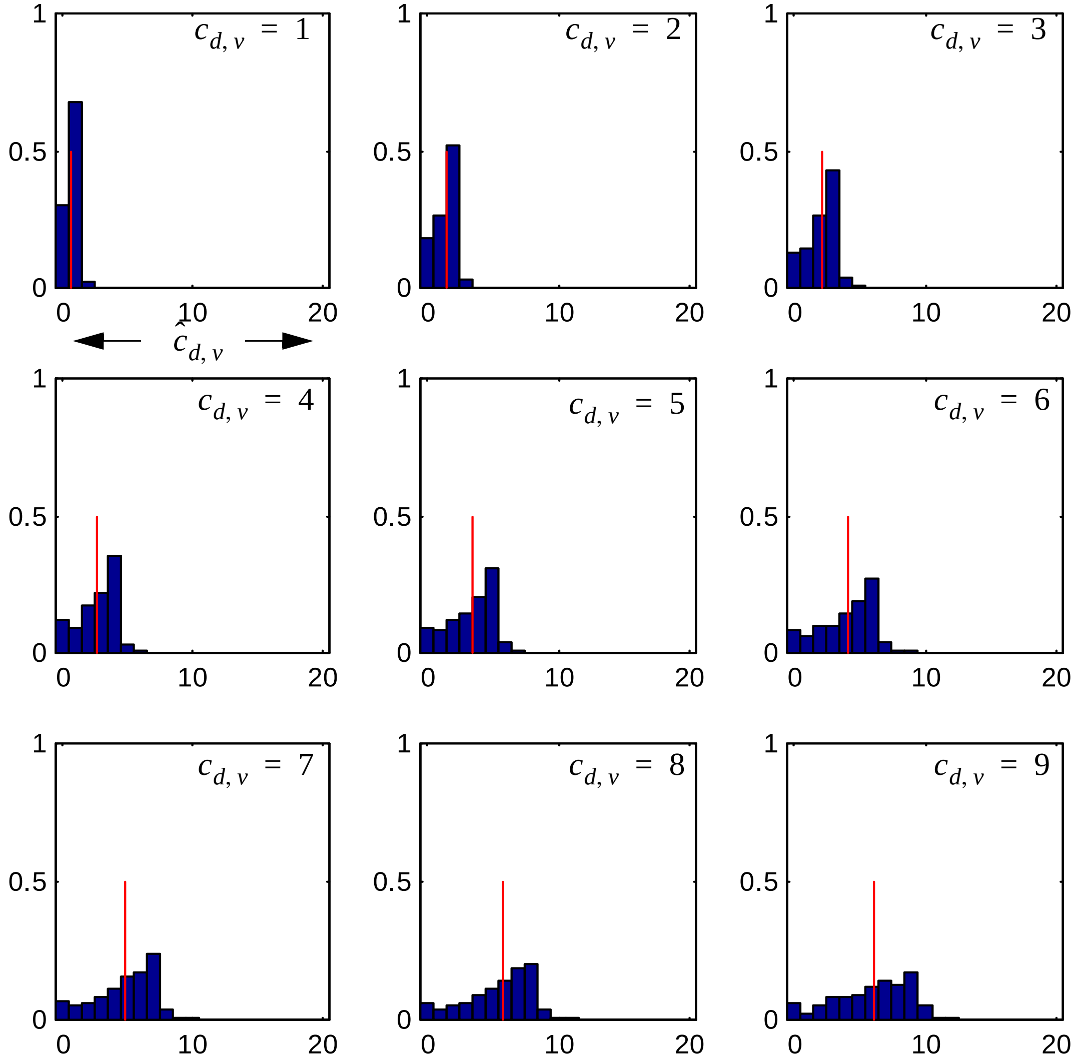


Figure 3-1 Histograms showing the distribution of $\hat{c}_{d,v}$ for various values of $c_{d,v}$. The mean value is shown as the thin line overlaying the distribution.

$c_{d,v}$	1	2	3	4	5	6	7	8	9	$c_{d,v}$
$E\{\hat{c}_{d,v}\}$	0.72	1.42	2.15	2.74	3.48	4.10	4.88	5.76	6.09	$0.65\hat{c}_{d,v}$

Table 3-1 Expected values for the hypothesis term counts given their reference value.

3.3.2 Document Length

Figure 3-2 shows the relationship between the estimated document lengths of the Top-1 hypotheses and the reference document length. Note that the estimated document length is consistently smaller than the reference document length. Because the document length is a nonlinear combination of the term counts, it would be nontrivial to compute its probability distribution directly from the probability distributions of the term counts. The problem is that the space of all possible values for $c_{d,v}$ is rather large, and so deducing the probability model for them will be very difficult when there is a large number of document terms. Luckily, when the number of terms is larger than five the distribution of the document length becomes indistinguishable from a Gaussian, due to the central limit theorem [63].

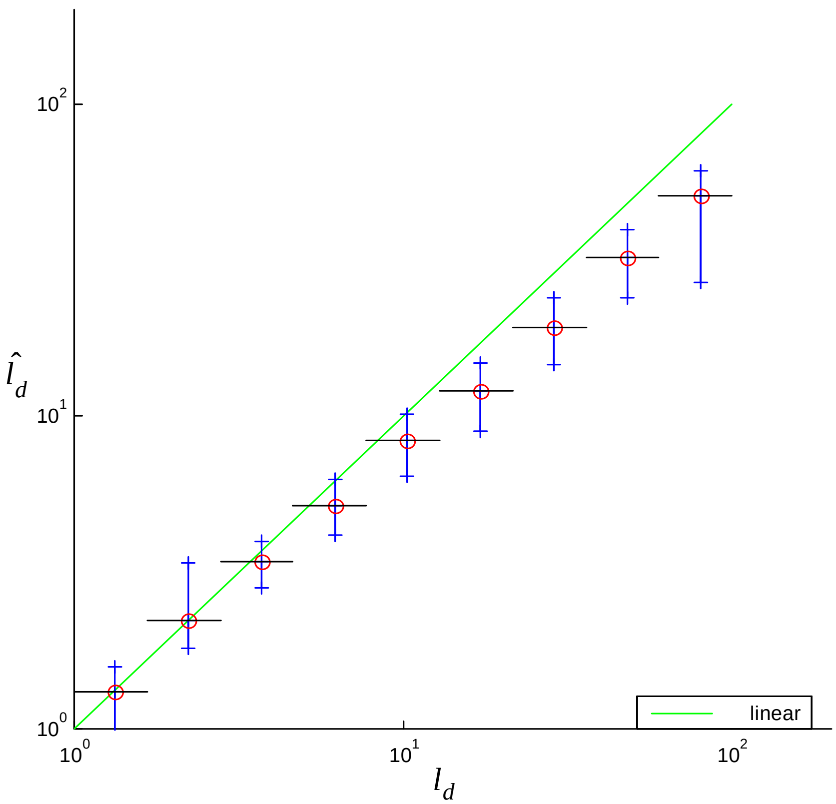
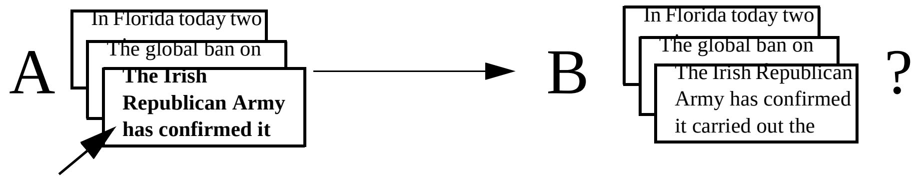


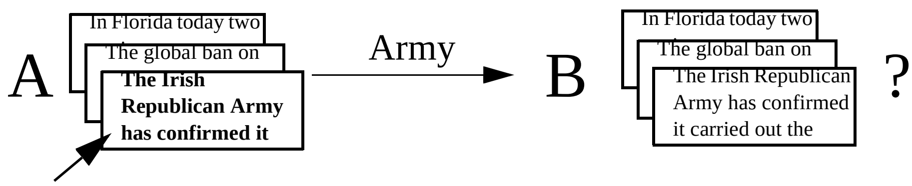
Figure 3-2 Distribution of values for the hypothesized document length $\hat{l}_d$ given the reference document length l_d given the hypothesis document length. The horizontal bars show the range covered, the circles indicate the means, and the crosses show the 1σ range of values.

3.3.3 Term Significance

If we examine the TFIDF relevance equations very closely we see that the IDF term significance weights are in fact a special case of mutual information. Consider a communication problem where a document D_d is to be chosen at random from the complete set of documents: $\boldsymbol{D} = \{D_{d=1}, D_2, \ldots, D_{n_d}\}$ The goal is to transmit the identity of the document from “A” to “B” by sending some portion of its contents.



At “A,” a single term v is drawn randomly and transmitted across the channel according to a predefined probability distribution $P(v|D_d)$ known both to “A” and “B.”



The recipient at “B” knowing the probability distributions for $P(v|D_d)$ calculates the difference in uncertainty about the identity of the document before and after transmission. This is defined as the *mutual information* between the set of documents and the term, defined $I(\boldsymbol{D};v) = H(\boldsymbol{D}) - H(\boldsymbol{D}|v)$ where the entropy and conditional entropy are defined:

$$H(\boldsymbol{D}) = -\sum_{i=1}^{n_d} p(D_i)\log_2[p(D_i)]$$

$$H(\boldsymbol{D}|v) = -\sum_{i=1}^{n_d} p(D_i|v)\log_2[p(D_i|v)]$$

The mutual information indicates the amount of (theoretical) information that this word carries about the documents in general and in some sense the relative importance of this word if it is found in a query. The mutual information between a vocabulary word and the document set becomes the IDF when we use a fixed value for term presence instead of the probability distribution for term presence.

Define the space of independent documents

$$\boldsymbol{D} = \{D_{d=1}, D_2, \ldots, D_{n_d}\}$$

where the probability of document *a-priori* is

$$P(D_d) = 1/n_d$$

Define the document the term presence probabilities

$$P(v|D_d) = P_{I_{d,v}}(i = 1)$$

The document probability conditioned on a term probability is

$$P(D_d|v) = P(v|D_d)\frac{P(D_d)}{P(v)} = \frac{P(v|D_d)(1/n_d)}{\sum\limits_{d' = 1\ldots n_d} P(v|D_{d'})(1/n_{d'})} = \frac{P(v|D_d)}{\sum\limits_{d' = 1\ldots n_d} P(v|D_{d'})}$$

Consider the term significance of term v to be the mutual information with value

$$I(\boldsymbol{D};v) = H(\boldsymbol{D}) - H(\boldsymbol{D}|v)$$

and by definition, this expands into

$$I(\boldsymbol{D};v) = -\sum_{d = 1\ldots n_d} P(D_d)\log_2 P(D_d) + \sum_{d = 1\ldots n_d} P(D_d|v)\log_2 P(D_d|v)$$

using Boolean term presence values instead of probability distributions

$$P_{I_{d,v}}(i = 1) = \begin{cases} 1 \ (i_{d,v} = 1) \\ 0 \ (i_{d,v} = 0) \end{cases}$$

The mutual information reduces to the inverse document frequency.

$$I(\boldsymbol{D};v) = \log_2(n_d) - \sum_{d = 1\ldots n_d} 1(v|D_d) = \log_2(n_d) - \log_2(\sum_d i_{d,v}) = \mathrm{idf}_v$$

Because the mutual information is a fixed number derived from the distribution of the term presence probabilities, it has no probability distribution of its own, unless we are to consider *prior* distributions of term presence probability distributions.

3.4 Probabilistic Relevance Computation

Even if we had perfect knowledge of the probability distributions for the term counts and term presence random variables, we would still need an appropriate way of judging the relevance between a text query and a document composed of the set of these distributions for each vocabulary word. The goal of the probabilistic formulation is to derive a probability distribution for $\hat{\mathrm{rel}}(q, d)$ given the probability distributions $P_{\hat{I}_{d,v}}(i)$ and $P_{\hat{C}_{d,v}}(c)$ so that the *relevance between a document and a query is both a function of observed IR features and the distribution of parameters due to speech recognition.* This distribution, $p_{\mathrm{Rel}(q,d)}(\mathrm{rel})$, apparently does not have a closed form expression in terms of the term presence and term count distributions but the next sections will attempt to arrive at a practical approximation.

3.4.1 Explicit Computation Of Relevance Probability Distribution

The most accurate probability distribution is one explicitly computed from the distributions of the term presence, term count, and document length parameters. Since the document length we are using is a fixed value, it falls out of an explicit computation. First, construct a composite vector of the space of all hypothesized term presences and term counts for each document.

$$\hat{\boldsymbol{d}}_d = \{\hat{\boldsymbol{i}}_d, \hat{\boldsymbol{c}}_d\}$$

Then, compute the mass distribution function of this composite vector.

$$P_{\hat{\boldsymbol{D}}_d}(\boldsymbol{d}_d) = P_{\hat{I}_{d,v=1\dots n_v}, \hat{C}_{d,v=1\dots n_v}}(i_{d,v=1\dots n_v}, c_{d,v=1\dots n_v})$$

Find the space R_z over which the relevance is less than or equal to some fixed value z.

$$R_z = \{\hat{\boldsymbol{d}}_d | \mathrm{rel}(q,d) \le z\}$$

The cumulative distribution function of z is then

$$F_Z(z) = P[\boldsymbol{d}_d \in R_z] = \int_{\boldsymbol{d}_d \in R_z} \dots \int P_{\hat{\boldsymbol{D}}_d}(\boldsymbol{d}_d)\,\mathrm{d}\boldsymbol{d}_d$$

And the probability distribution function of z is then the derivative

$$p_{\hat{\mathrm{Rel}}(q,d)}(\mathrm{rel}=z) = p_Z(z) = \frac{d}{dz}F_Z(z)$$

This seems like an arduous path to take for computing the distribution of relevances given distributions for hypothesized term presence and term count. However, since the number of query terms is very small compared to the number of document terms, it turns out that only a very small space of R_z for the composite vector $\boldsymbol{d}_d$ is actually explored in the product $(b_{q,v})(\hat{c}_{d,v})$. As a result the probability distribution for the relevance reduces in dimensionality from $\sim n_v$ to simply the number of terms that the query and document have in common. Very rarely are all the terms in the query also in the document hypotheses, with 80% of the query-document pairs having 1 or 2 terms in common. With this in mind we turn to simpler formulations of the relevance probability distribution.

3.4.2 Using Expected Values In Estimated Relevance

The simplest possible distribution for the estimated relevance is an impulse at a fixed value:

$$P_{\hat{\mathrm{Rel}}(q,\,d)}(\mathrm{rel}\,|\,\hat{\boldsymbol{d}}_d) = \delta(\mathrm{rel} - \hat{\mathrm{rel}}(q,\,d))$$

Where the hypothesized term presences and counts are collected into one composite vector $\hat{\boldsymbol{d}}_d = \{\hat{i}_{d,v}, \hat{c}_{d,v}\}_{v=1\ldots n_v}$. If the expected values for the probability distributions of term presence, term count, and document length are used instead of the raw distributions:

$$\hat{\mathrm{rel}}_E(q,\,d) = \frac{1}{E\{l_d|\hat{\boldsymbol{d}}_d\}}\left(\sum_v (b_{q,\,v})E\{c_{d,\,v}|\hat{c}_{d,\,v}\}I(\boldsymbol{D};v)\right) \tag{3-3}$$

Strictly speaking, $E\{\hat{\mathrm{rel}}(q,\,d)\} \neq \hat{\mathrm{rel}}_E(q,\,d)$ and this representation fails to capture the potentially large variation in the values of term presence, term count, and document length, but is extremely easy to compute and requires no additional storage beyond the typical document representation in Equation 3-1.

3.4.3 Assuming Independent Gaussian Distributions for Term Count

A satisfactory compromise between explicit and degenerate modeling of the relevance distribution is to assume that the relevance formula is simply the weighted sum of a set of random variables with Gaussian distributions. Figure 3-1 showed the distributions for the reference term count given the hypothesized term count, and these *resemble* Gaussians in that they are unimodal in shape. Obviously the actual distributions are not Gaussian but for the purpose of computing the possible relevance values a Gaussian model is extremely convenient.

Since we have already reduced the relevance equation to a simple sum of intersecting terms in the previous section, now we can turn to a far simpler computation of the distribution assuming Gaussian distributed term counts:

$$\hat{\mathrm{rel}}(q,\,d) = \frac{1}{E\{l_d|\hat{\boldsymbol{c}}_d\}}\left(\sum_v (b_{q,\,v})N(\mu_{\hat{c}_{d,\,v}}, \sigma^2_{\hat{c}_{d,\,v}})I(\boldsymbol{D};v)\right) \tag{3-4}$$

Where we have assumed that the distribution of the document length is degenerate, as we did earlier. Since the term counts are independently distributed by our assumptions:

$$P_{\mathrm{Rel}(q,\,d)}(\mathrm{rel}|\hat{\boldsymbol{d}}_d) = N(\mu_{d,\,q}, \sigma^2_{d,\,q})$$

Define a new variable $\beta_{q,\,v}$ the *modified query term count* as $\beta_{q,\,v} = (b_{q,\,v})I(\boldsymbol{D};v)$ which expresses simultaneously the importance of the term in the query and the importance of the term in resolving the document set. Now the values for $\mu_{d,\,q}$, and $\sigma^2_{d,\,q}$ are defined in the following way:

$$\mu_{d,\,q} = \frac{1}{E\{l_d\}} \sum_{v \in S_{d,\,q}} (\beta_{q,\,v})(\mu_{c_{d,\,v}})$$

$$\sigma^2_{d,\,q} = \frac{1}{E\{l_d\}^2} \sum_{v \in S_{d,\,q}} (\beta_{q,\,v})^2(\sigma^2_{c_{d,\,v}})$$

Where the space $S_{d,\,q}$ of *active terms* v between a query-document pair is defined:

$$S_{d,\,q} = \{v|(b_{q,\,v})(\hat{c}_{d,\,v}) > 0\}$$

And instead of trying to explicitly compute the relevance probability distribution only the estimates for the mean and variance of these Gaussians are required. Note that the most likely estimated relevance value, that is the expected relevance, under this new formulation is:

$$E\{\hat{\mathrm{rel}}_{d,\,q}\} = (\mu_{d,\,q}) = \frac{1}{E\{l_d\}} \sum_{v \in S_{d,\,q}} (\beta_{q,\,v})(\mu_{c_{d,\,v}}) = \frac{1}{E\{l_d\}} \sum_{v \in S_{d,\,q}} (b_{q,\,v})E\{c_{d,\,v}\}I(\boldsymbol{D};v)$$

Which is of course the same formula as we had in Equation 3-3, since the expected value of a sum is the sum of the expected values. In Chapter 5 and Chapter 6 alternatives to using the true probability model for the relevance values will be shown that still improve overall performance by incorporating the inherent uncertainties in the term count and term presence features. In addition approximations for $P_{\hat{I}_{d,\,v}}(i)$, $P_{\hat{C}_{d,\,v}}(c)$, $E\{P_{\hat{I}_{d,\,v}}(i)\}$, and $E\{P_{\hat{C}_{d,\,v}}(c)\}$ derived directly from the N-Best lists and lattices will be discussed, and shown to yield improvements in overall performance.

3.5 Consequences of Probabilistic Relevance

In each of the three previous subsections a different approach to estimating the relevance distribution was discussed. If we did have a perfect model describing the distribution of the relevance of each document to each query, what would we do with it? Hypothetically, the more knowledge we have describing the actual relevance distribution, the more informed a decision can be made regarding the relative merits of document and query pairs. However this has not been properly established.

In Section 2.2.5 a standard procedure for the evaluation of information retrieval was presented. In that procedure a free parameter r_d, the number of retrieved documents to verify, was varied in order to simulate a variety of possible retrieval conditions. When the r_d is very small compared to the number of documents in the test set, $r_d \ll n_d$, there is little hope of retrieving all of the matching documents and the figure of merit is usually the precision. In such a circumstance we desire that all of the retrieved documents be relevant to the query, a situation not unlike a world wide web search engine. However, if we desire to find as many of the matching documents as possible, we are looking to maximize recall. An example where recall is important is a search through a law library for legal precedents. As we will see below the choice of r_d has an even greater effect on the relative rankings of documents scored using the probabilistic relevance, due to uncertainty that arises from the variance of the relevance.

3.5.1 Using only the Mean of the Relevance

The standard evaluation criterion described in Section 2.2.5 restricts us to working with a simple sorted list of document relevances for each query. Given any distribution the single number that ranks the document with respect to other documents in terms of its relevance would be either the maximum likelihood or the expected value of the relevance. Since we have already demonstrated in the last section that the relevance is the sum of a fair number of independent random variables, and that such a sum is fairly likely to have nearly a Gaussian distribution, we know that both the maximum likelihood relevance and the expected value for the relevance are the same: the mean.

If the mean value of the relevance is the sole criterion for the establishment of the relative rankings for the documents then there is no need to model the distribution beyond this value. Because of the symmetry of the distributions and regardless of any inherent variance in the relevance the expected ordering of documents in decreasing relevance will always be the same as the ordering by mean relevance.

3.5.2 Utility for the Variance of the Relevance

Although the current evaluation scheme cannot incorporate it, there is a possibility that the variance of the relevance could provide valuable clues about the relative quality of retrieved documents. Figure 3-3 shows distributions for the relevance of the first five documents retrieved for a specific query. The values chosen for the term count variance were drawn empirically from the test set data shown in Figure 3-1. In the table the first five documents are ranked by their mean value according to the traditional rules of the evaluation criterion. When sorted in this way the information about term count variance is discarded.

d	μ_d	σ_d^2
2374	9.59	17.5
648	8.33	17.4
1275	7.43	11.0
1821	5.20	7.95
2112	5.17	23.60
870	4.85	1.32

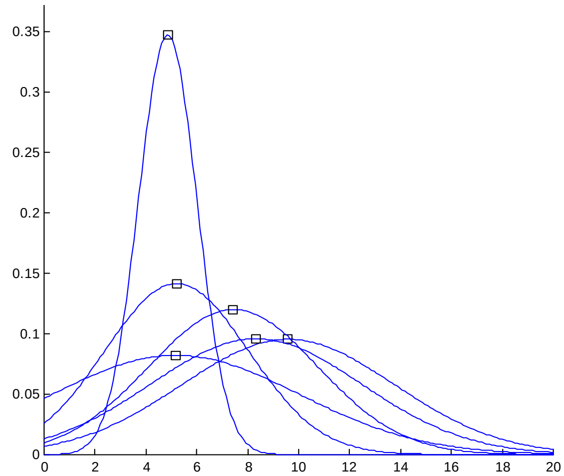


Figure 3-3 First six documents, sorted by decreasing relevance, for a particular query. Note that the variance has a wide range, principally due to the range in the number of query terms involved in its computation.

Given a set of relevance mean and variances: $\mu_{d,\,q}$ and $\sigma^2_{d,\,q}$, the probability that document j will be more relevant than document k given query q is the probability that the difference will be positive:

$$Q(\mu_{j,\,q} - \mu_{k,\,q}, \sigma^2_{j,\,q} + \sigma^2_{k,\,q}, 0)$$

Where the function Q is defined as the integral of a Gaussian distribution from the right:

$$Q(\mu, \sigma^2, x) \;=\; \int_x^{\infty} \frac{e^{\frac{1}{2}\frac{(x' - \mu)^2}{\sigma^2}}}{\sqrt{2\pi\sigma^2}} dx'$$

Because the variance of the relevance has a wide range of values, ranking the documents according to their *probability of greater relevance* becomes dependent on how many documents you are comparing them to. Figure 3-4 shows that the relative rankings of documents 2112 and 870 depend on whether you are comparing them with the 7 most likely documents or 100. Since the variance for document 2112 is so large, even when compared to a document having a much smaller relevance there is still a nontrivial probability that document 2112 will have a smaller relevance.

It is somewhat odd that the rank order of the documents in terms of probability of greater relevance depends on the comparison documents and does not lead to an obvious conclusion for how to rank them. It also points out a flaw in the evaluation criterion which fails to incorporate any sense of reliability due to the variance in relevance scores. Until a new evaluation method is designed, the variance of the relevance and thus the variance for term counts can be ignored for the purpose of estimating the relevance of documents to queries.

d	$p(\mathrm{rel}_{2112} > \mathrm{rel}_d)$	$p(\mathrm{rel}_{870} > \mathrm{rel}_d)$
1	0.245	0.137
2	0.311	0.210
3	0.351	0.232
4	0.498	0.453
5	-	0.474
6	0.526	-
7	**0.558**	**0.573**
8	0.560	0.583
9	0.589	0.622
10	0.605	0.646
20	0.667	0.797
50	0.748	0.933
100	0.805	0.996

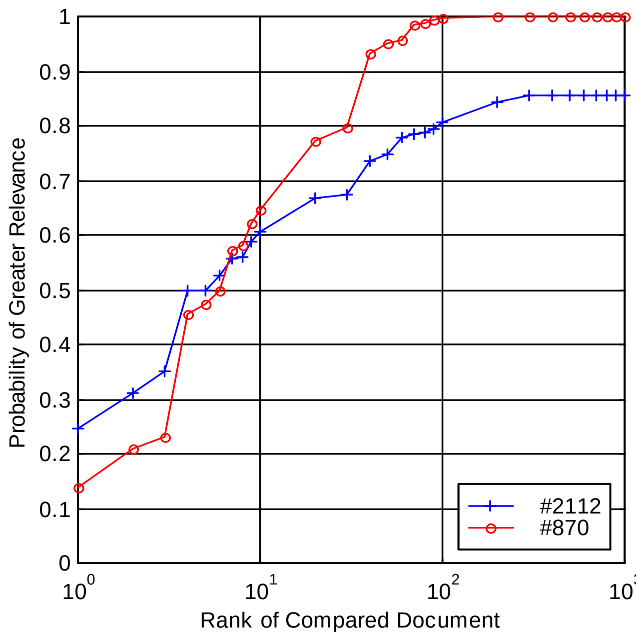


Figure 3-4 Probability that documents 2112 and 870 are more relevant than documents of particular ranks. Because document 870 has a relevance with a lower variance than 2112, it is more likely to be relevant than document 2112 for most of the document set, although it is less likely for the top 5.

3.6 Summary

The standard application of TFIDF relevance to the Top-1 hypothesis from speech recognition does not account for the multiple competing hypotheses due to the uncertainty in the recognition procedure. To cope with this a new formulation for relevance was introduced that is conceptually based on the TFIDF equation but uses probability-based features derived from speech recognition. The term weighting component of the TFIDF equation, the inverse document frequency, was shown to be a special case of the mutual information between a term and the document set when perfect knowledge of the transcription is present. In addition to the weighting parameter, the effect of speech recognition on the input term count and term presence was modeled with new random variables having independent distributions. The use of statistical parameters to represent these distributions led to a probabilistic formulation of the relevance of documents to queries that is approximately Gaussian due to the relevance equation being composed of a weighted sum of independent random variables. This ultimately resulted in a relevance equation that simultaneously incorporates the multiple competing hypotheses and the presence of uncertainty in the recognition process.

In Chapters 5 and 6 methods will be explored for extracting the statistical parameters that constitute the random variables of term presence and term counts directly from the recognition data structures. The focus will be on improving the estimate of term presence probability and term count accuracy by using the multiple hypotheses found in lattices for Chapter 5 and the N-Best lists for Chapter 6. Once these estimates have been acquired they can be used in the new relevance formula which now can use fractional term presence rather than requiring Boolean decision making.

The next chapter describes in detail the databases used and outlines a procedure for evaluating the estimates of term presence and term count as well as the quality of retrieval from their integration into the final relevance equation.

4. Experiment Design

In Chapter 2 both the speech recognition and information retrieval components were described in only a general way. This chapter contains the implementation details for each, along with a description of the databases used and the evaluation methods employed.

4.1 Speech Recognition

The configuration of Sphinx III in this work was identical to that used in the 1997 TREC-7 Evaluation [44]. The vocabulary contained 65,314 words derived from the most common in the language model training data. The acoustic model was built with approximately 100 hours of broadcast news speech captured from television and radio programs including ABC Nightline, ABC World News Now, ABC World News Tonight, CNN Early Prime News, CNN Headline News, CNN The World Tonight, CSPAN Washington Journal, and NPR All Things Considered. The language model was constructed from a union of the 1992-1996 Broadcast News Text corpus and the 1996 Newswire Data from the Linguistic Data Consortium [66], a total of approximately 500 million words of training data.

In order to complete the recognition phase in approximately 35 times real time a technique was used to speed up the recognition search engine [54]. In previous experiments this method incurred a relative error increase of approximately 10%, leading to a word error rate on the test set of 32.1% compared with 23% to 35% for other similarly configured systems at the TREC-7. A comparable “No-Holds-Barred” version of the Sphinx III recognizer would most likely have yielded a 29-30% word error on the same test set. Since the speech recognition system requires discrete utterances on the order of 10-20 seconds a version of CMUseg [67][68] was used to chop the speech into smaller *document subsegments* before recognition.

4.2 Information Retrieval

As discussed in Chapter 3 the information retrieval component was simple to reflect the predominant approach in the field. Performed identically to the TREC-7 effort the text processing served to map the approximately 65000 word vocabulary down to a core of approximately 39000 words by removing words from a fixed *stoplist* of 812 words and by removal of suffixes (*stemming*) from the remainder according to an enhancement of the Porter algorithm [35]. The enhancements properly convert possessive forms of many words, and have a set of 2400 special mappings derived from the 1995 edition Houghton-Mifflin dictionary [65] that express irregularities in the English language. For example, mapping the word FEET into FOOT.

4.3 Database

The speech database in this research is a subset of the TREC-7 test set with a total of 23 queries and 2597 of the 2866 original documents. Because of unresolved runtime errors in the recognition compoment lattices could not be produced for the remainder of the stories. Rather than confound the experimental design with these stories the subset was chosen. Of this set 326 documents were relevant to at least one query. The average document length (before text processing) was 170 words, with a total of approximately 440,000 words. The average query length was 8 words.

To show how the number of documents interacts with retrieval performance, four differently sized document sets were constructed. The set of size 326 consists only of the documents relevant to at least one query, and the set of size 2597 contains the entire set of documents. To form document sets of size 652 and 1304 randomly selected documents from the non-relevant document set were added. Because the particular choice of these documents would affect the results 100 different randomly generated documents sets were created. Any experiments using the 652 or 1304 size sets were executed on each of the 100 instances of the set and the performance values were averaged. For the document sets of size 326, 652, 1304, and 2597, the fraction of documents relevant to any query are 1, $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ respectively. To test the new ideas of this work, a two-tiered evaluation system was developed to quantify innovations in the term count and term presence probability estimators from Section 3.4 as well as the overall system performance improvements.

4.4 Evaluation of Estimators

To evaluate the effectiveness of the term count and term presence probability estimators a simple comparison will be drawn between the performance of the Top-1 hypothesis and the new estimator. For term presence, the *word precision* and *word recall* will be used. In a similar way to the use of a Receiver Operating Characteristic Curve (ROC) in detection experiments, the *term precision* and *term recall* measure the ability to predict the presence of a word in the original reference texts.

4.4.1 Term Presence

First we consider the case where the probability estimators are Boolean, that is they assign a value of 0 or 1 to signify the absence or presence of a term in the hypothesis. In the previous chapter a new model was proposed that assigned a probability instead of a Boolean value to the estimated term presence, and a method for evaluating this model will be explained later.

To compute the term precision and term recall, either the entire test set may be considered as a whole, or the values for each individual document can be averaged. The advantage of the latter is that documents containing a very large number of terms will not dominate the test set. The advantage of the former is that it accounts for each word as a separate data-point, regardless of document boundaries. Because the size of the documents varied widely in the test set, ranging from 1 to 1169 terms, the document-averaged term precision and term recall will be used to evaluate the term presence estimator. The per-document term precision and term recall are defined:

$$\pi_d = \frac{\sum_{v=1}^{n_v} (\hat{i}_{d,v})(i_{d,v})}{\sum_{v=1}^{n_v} (\hat{i}_{d,v})} \qquad \phi_d = \frac{\sum_{v=1}^{n_v} (\hat{i}_{d,v})(i_{d,v})}{\sum_{v=1}^{n_v} (i_{d,v})}$$

Where $i_{d,v}$ and $\hat{i}_{d,v}$ are the reference and hypothesis term presence values. The overall term precision and term recall are computed as the average per-document over the set:

$$\pi = \frac{1}{n_d} \sum_{d=1}^{n_d} \pi_d \qquad \phi = \frac{1}{n_d} \sum_{d=1}^{n_d} \phi_d$$

Table 4-1 shows term precision and recall for the reference texts and the Top-1 hypothesis. Note that the term precision and term recall are approximately equal for the Top-1 hypothesis, which is most likely the result of simultaneously minimizing word insertions and deletions in the speech recognition system.

Test Set	π	ϕ
Reference	1.00	1.00
Top-1	0.76	0.74

Table 4-1 Term presence precision and recall performance of the Top-1 hypothesis.

However, some accommodation for term precision and term recall metrics is needed since we are primarily interested in estimators for term presence that use probabilities rather than Boolean values. One possible extension is to evaluate a decision rule $P_{\hat{I}_{d,v}}(1) > \theta_i$ that assigns a Boolean value to $\hat{i}_{d,v}$. Such a hypothesis testing scheme is sensible because we are certain that the decision is either true or false; either the term actually was spoken or it was not. In order to observe the performance of this decision rule, the threshold θ_i is varied smoothly over the interval $[0.0, 1.0]$ and the values for term precision and term recall are noted for the new estimates of $\hat{i}_{d,v}$.

Another useful measurement is the *oracle performance* for the decision rule using the term precision estimator. An oracle experiment is an attempt at finding the upper bound to a decision rule, where a free parameter is varied locally to maximize some global performance. Although not indicative of actual performance the oracle performance is a good demonstration of the potential performance. The criterion for assessing global performance in this case is some unknown combination of term precision and term recall. Since we do not know outright whether it is preferable to have a very high term precision or term recall for information retrieval it is prudent to explore the space that maximizes a linear combination of the two: $\pi(1-\lambda) + \phi\lambda$, with λ varying smoothly over the interval $[0.0, 1.0]$ in each experiment. Figure 4-1 illustrates both the actual and the oracle performance of an example term presence estimator with the parameters θ_i and λ varied for each as well as the Top-1 hypothesis performance. Note that there is significant performance to be gained by optimizing the value of θ_i for each document if this is possible.

4.4.2 Term Count Error

To evaluate the performance of term count estimators, the average *term error* of the document set will be used [34]. The term error has been used by several organizations to evaluate the overall error of a system when the order of the words is not relevant, as is the case in a vector space model [51]. The Term error for each document is defined as

$$\tau_d = \frac{\sum_{v=1}^{n_v} |\hat{c}_{d,v} - c_{d,v}|}{\sum_{v=1}^{n_v} c_{d,v}}$$

The overall term error for the test set is then computed as the average of the error for each document.

$$\tau = \frac{1}{n_d} \sum_{d=1}^{n_d} \tau_d$$

Because the estimator will yield a probability distribution and not a single number, comparisons between the reference count and the expected value of the estimated count will be applied to test the estimator performance. The performance for the term count estimate will be averaged in the same way that the performance of the term presence estimate was averaged over the documents and not for the entire set.

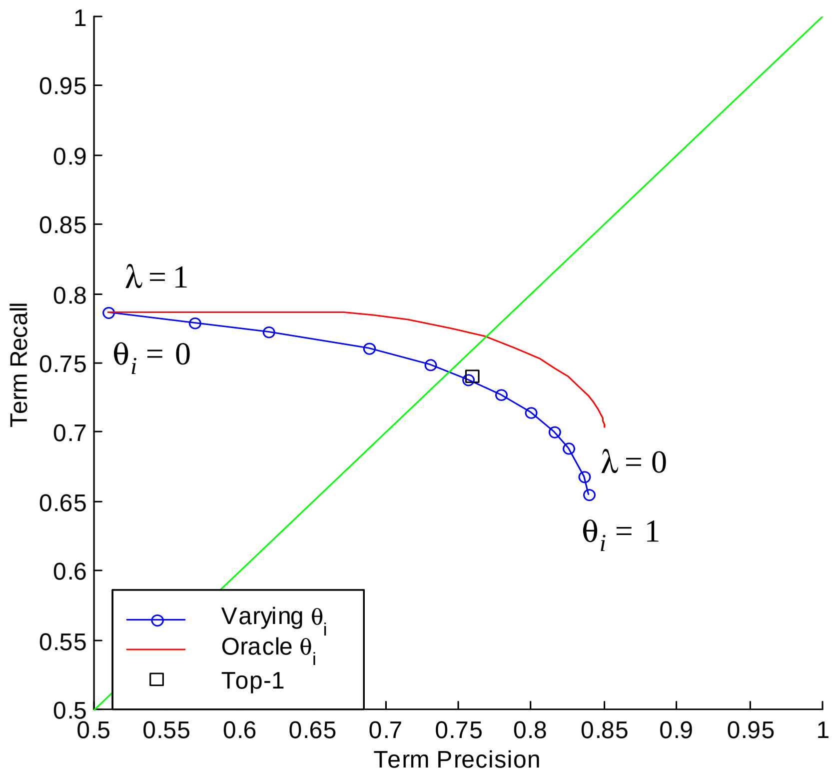


Figure 4-1 Term precision and recall for a sample term presence estimator. Performance of the Top-1 estimator is the square box. The curve with circles is the performance with the value of θ_i sampled at several points over $[0.0, 1.0]$. The other curve is the oracle performance, where θ_i takes on the value that maximizes the criterion $\pi(1-\lambda)+\rho\lambda$, with the parameter λ sampled at several points over $[0.0, 1.0]$.

The second criterion for comparing the hypothesized term counts to the reference term counts will be the *correlation coefficient* averaged over the document set [63]. This value is computed over the terms that have either a nonzero reference or nonzero hypothesis value, and then averaged across all the documents. The term error and correlation coefficient for the reference texts and the estimated term counts based on the Top-1 are shown in Table 4-2.

Test Set	Term Error: τ	Correlation: ρ
Reference	0.000	1.000
Top-1	0.137	0.330

Table 4-2 Term error and correlation coefficient measures of the term count averaged over the document set, using both reference texts and Top-1 hypotheses.

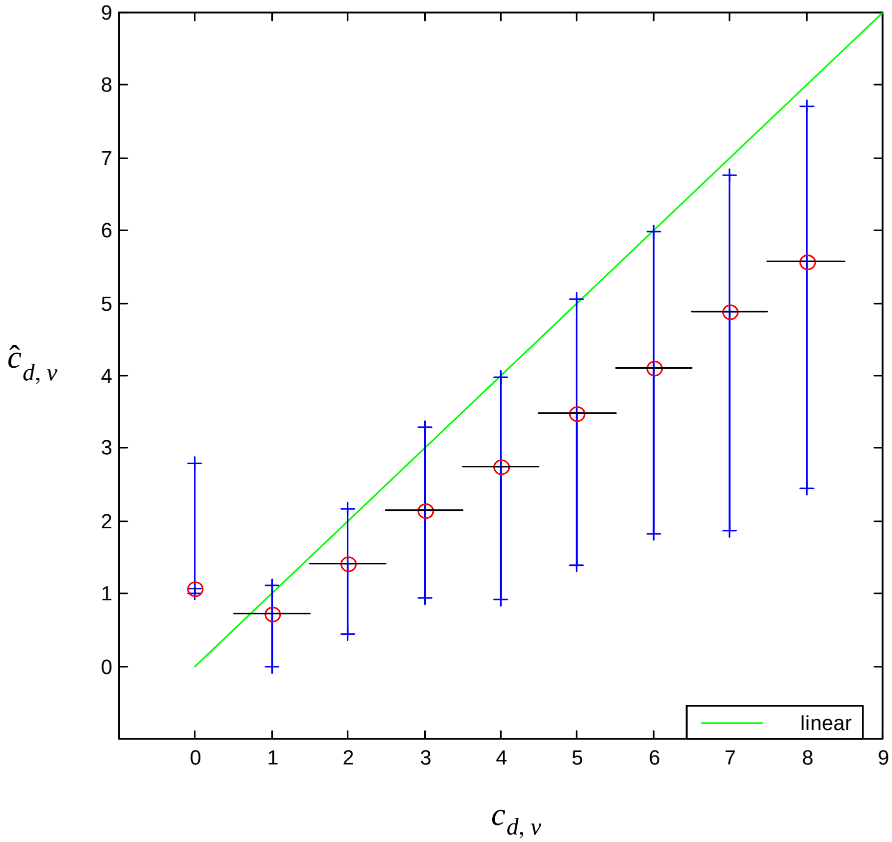


Figure 4-2 Distribution of values for the term counts $\hat{c}_{d,\,v}$ estimated from the Top-1 hypotheses given the reference counts $c_{d,\,v}$. The horizontal bars show the range covered, the circles indicate the means, and the crosses show the 1σ range.

Another clue to the effectiveness of the term count estimator does not require using the expected value in order to be measured. A two-dimensional histogram can show how the reference values are distributed for different values of the estimate, including any deletions or insertions by the speech recognition system. Figure 4-2 is very similar to a contour map showing the density of points occurring at each value of reference versus hypothesis term count. Each horizontal line collects the reference points within its extremum, and is vertically positioned at the mean value of the hypothesized values, which is also marked with a circle. The cross-hairs above and below the mean mark the 1σ points such that they contain a total of ~84% of the data, with equal amount of data above and below the mean value.

4.5 Full System Evaluation

Employing the evaluation criterion laid out in Section 2.3, Figure 4-3 shows how IR quality degrades for reference texts as n_d is increased but M_d is held fixed. The vertical axis measures the average of the five points of precision and recall (the overall IR quality) discussed in Section 2.2.5. Beyond the TREC-7 test set of 2597 additional documents were randomly selected from the 1997 Primary Source Media (PSM) database and included to observe this degradation.

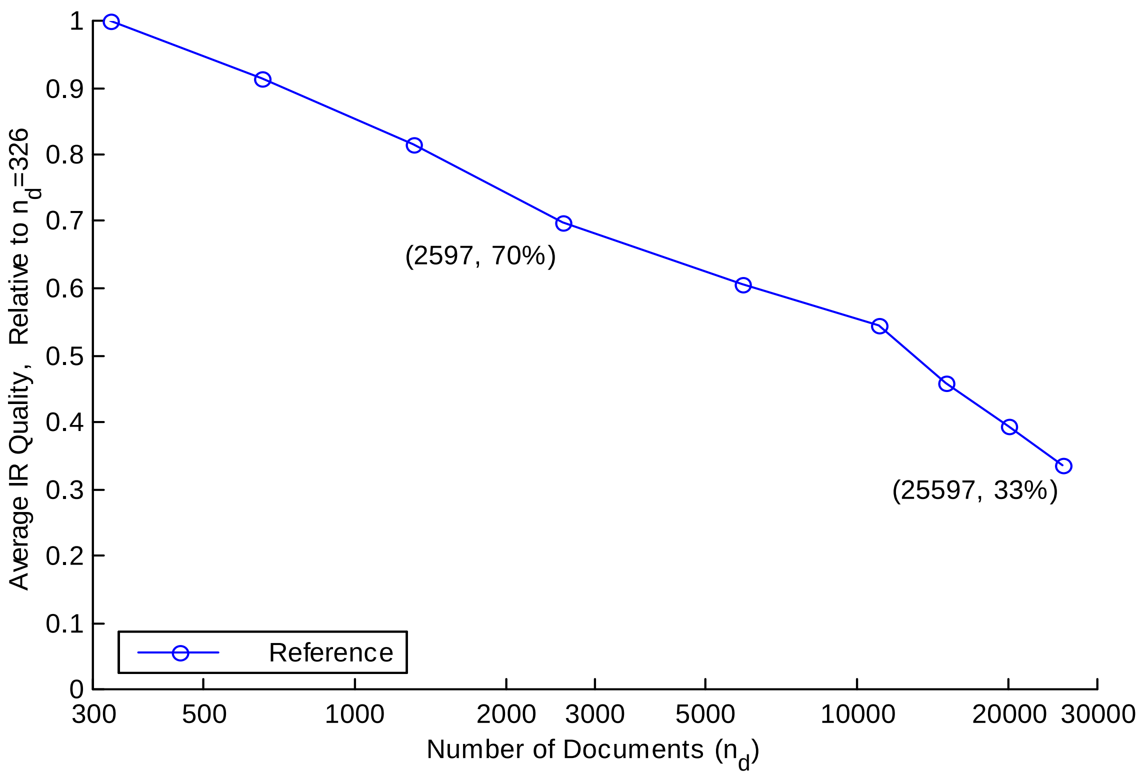


Figure 4-3 Degradation in average IR quality for reference transcriptions as the size of the document set increases but the number of relevant document remains fixed at 326. Average IR quality is the average degradation of the 5 points of precision and recall specified in Section 2.2.5, relative to performance with $n_d = 326$.

In contrast, the degradation proceeds far more rapidly for speech recognition documents as the number of documents is increased. Figure 4-4 shows a comparison of the relative degradation of average IR quality for reference texts and Top-1 hypotheses as the test set increases. At 2597 documents the performance with the Top-1 documents is 89% that of the reference texts, but we have exhausted all the speech recognition documents at this point in the database.

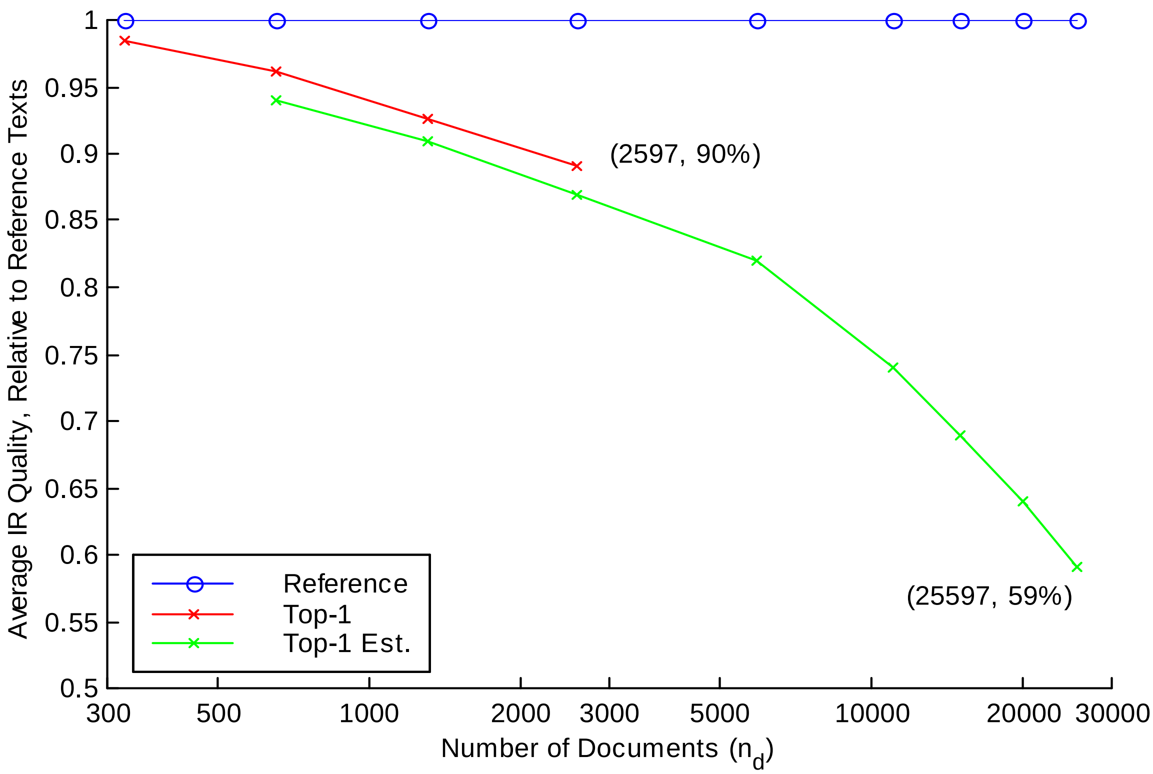


Figure 4-4 Degradation in IR performance for speech recognition transcripts (Top-1) relative to performance of reference texts, as the size of the document set increases but the number of relevant documents remains fixed. The performance measure is the average degradation of the 5 points of precision and recall specified in Section 2.2.5.

In order to predict how this trend may continue as the number of documents increases, the same reference texts from the PSM database used in the previous experiment were added instead of actual speech recognition documents. Although this experiment tracks the Top-1 hypotheses very well it can only partially predict the results of an experiment using only speech recognition documents. However, in the neighborhood of 50,000 documents the degradation of the average IR quality using the Top-1 hypotheses could be as high as ~50% relative to the reference documents.

With this in mind the most important figure of merit for the experiments in this work will be how they mitigate performance degradation with speech recognition documents. At 2597 documents this amounts to some 10% relative to the reference texts for the traditional method using the Top-1 hypotheses. It is especially important that the improvements do not deteriorate as the number of documents is increased. The hope is that as the number of documents increases beyond the size of the existing test set the improvements made will become more significant, although we will be unable to test this without a larger database.

5. Extracting Relevant Content from Lattices

A lattice contains the complete search space traversed by the decoder of the speech recognition system as it attempts to evaluate various transcription hypotheses. Compared with only a single path in the Top-1 hypothesis there are 10^{100} paths represented in a typical lattice resulting in an average of 10 times as many hypothesized terms. Since lattices contain so many additional hypothesized terms they present an attractive opportunity for detecting content.

The goal of this chapter is to analyze the lattice directly, to isolate the total score associated with each individual node, to identify those nodes that are in competition, and to use the relative scores to improve the term presence and term count estimates. The focus will be on estimating the probability that a specific term occurs in a document considering all hypotheses in the lattice.

5.1 Properties of Lattices

As described in Section 2.1.2 the lattice is a compact representation of all hypotheses explored by the speech recognizer recognition and their recognition scores. The scores in this case are the weighted sum of the acoustic log-likelihood and the linguistic log-likelihood of observing the acoustics given the acoustic, lexical and language models. Although we desire them to be, these scores are not truly the log-likelihoods of the reference transcript given the acoustic observation and the universe of all possible spoken utterances and audible sounds. As in the last chapter, we can assume that the speech recognition score is somehow a measure of the relative merit of the hypothesis given the acoustic observation and the models.

5.1.1 Probability Based Model Used in Recognition

The manner in which the speech recognition system assigns a score to every hypothesis is straightforward and described here to motivate the use of probabilities in predicting term presence and term count. Given acoustic observations $\boldsymbol{O}$, the hypothesized transcription $\hat{S}$, and the combined model of acoustics, lexicon, and language model Λ, the output of the acoustic and language model are the probabilities $p(\boldsymbol{O}|\hat{S}, \Lambda)$ and $p(\hat{S}|\Lambda)$. The overall recognition probability is assigned to their product,

$p(\hat{S}, \boldsymbol{O}|\Lambda) = p(\boldsymbol{O}|\hat{S}, \Lambda)p(\hat{S}|\Lambda)$. To estimate the probabilities of particular document terms and the hypotheses containing them, we compute the hypothesis probability $p(\hat{S}|\boldsymbol{O}, \Lambda)$ given the observations and models

$$p(\hat{S}|\boldsymbol{O}, \Lambda) = \frac{p(\boldsymbol{O}|\hat{S}, \Lambda)p(\hat{S}|\Lambda)}{p(\boldsymbol{O}|\Lambda)} \tag{5-1}$$

To compute the prior probability of the acoustic observation $p(\boldsymbol{O}|\Lambda)$, we define the universe of all possible hypotheses as $\boldsymbol{S}$ and divide it into two partitions: $\boldsymbol{S}_1$, those hypotheses explored during recognition and $\boldsymbol{S}_0$, those hypotheses pruned by the search heuristics

$$p(\hat{S}|\boldsymbol{O}, \Lambda) = \frac{p(\boldsymbol{O}|\hat{S}, \Lambda)p(\hat{S}|\Lambda)}{\sum_{S \in \boldsymbol{S}} p(\boldsymbol{O}, S|\Lambda)} = \frac{p(\boldsymbol{O}|\hat{S}, \Lambda)p(\hat{S}|\Lambda)}{\sum_{S \in \boldsymbol{S}_0} p(\boldsymbol{O}|S, \Lambda)p(S|\Lambda) + \sum_{S \in \boldsymbol{S}_1} p(\boldsymbol{O}|S, \Lambda)p(S|\Lambda)}$$

The resultant formulation is simple

$$p(\hat{S}|\boldsymbol{O}, \Lambda) = \frac{p(\boldsymbol{O}|\hat{S}, \Lambda)p(\hat{S}|\Lambda)}{\boldsymbol{M}_0 + \boldsymbol{M}_1}$$

Where $\boldsymbol{M}_1$ is the known sum of the recognition probabilities for all paths in the lattice, and $\boldsymbol{M}_0$ is the unknown sum of the recognition probabilities for all the pruned paths. Because we cannot compute the total probability $\boldsymbol{M}_0 + \boldsymbol{M}_1$ we cannot deduce $\boldsymbol{M}_0$ directly, and its value will vary unpredictably from lattice to lattice.

5.1.2 Summing the Lattice Probabilities: Total Node Probabilities

If we look at a typical lattice in Figure 5-1, we see that in general it is composed of different paths containing competing words in varying numbers. Occasionally there will be a word that all paths must contain, a so-called *articulation point* [4]. In other situations, there will be more than 1000 possible hypothesized words all competing for log-likelihoods.

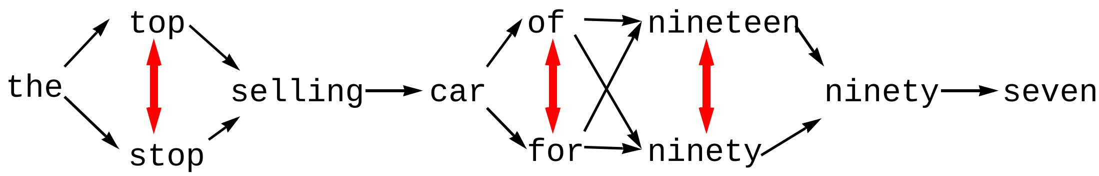


Figure 5-1 Competing words in a simple lattice.

Although the recognition system assigns a probability to every possible path through the lattice, for the purpose of estimating the term presence and term counts we are particularly interested in the probability for only *a particular node* with respect to its competing nodes. The probability of node n in the lattice can be computed using the conditional sum of hypotheses that contain it

$$p(n|\boldsymbol{O}, \Lambda) = \frac{\sum_{\hat{S} \in \boldsymbol{S}_n} p(\hat{S}|\boldsymbol{O}, \Lambda)}{\boldsymbol{M}_0 + \boldsymbol{M}_1} \tag{5-2}$$

where the space of paths containing the node is defined as $\boldsymbol{S}_n = \{\hat{S}|n \in \hat{S}\}$. Calculating the sum of the probabilities through any node requires the computation of the forward and backward probability, only an order ~N computation where N is the number of nodes in the lattice [64]. With the probability of a particular node isolated from the probabilities of the paths that contain it the paths themselves can be discarded altogether. The remainder of the work in this chapter will assume that the paths are unimportant once the node probabilities are available.

5.1.3 Compensating for the Missing Mass $\boldsymbol{M}_0$ Due to Pruning

In order to continue investigating the lattices, their hypotheses, and their attendant likelihoods, some method for compensating for the missing paths is necessary. Because a speech recognition system must finish its job in a reasonable amount of time an extremely large subset of all the possible hypotheses is left unexplored. This method of pruning the search space to a narrow collection of hypotheses is very effective in reducing the amount of computation required. The most common way to compensate for the missing mass $\boldsymbol{M}_0$ is to assume that it is related to the length of the utterance.

The rationale is that the pruning method is insensitive to the acoustic length of the hypothesis N_f and therefore removes a greater fraction of hypotheses as the utterance becomes longer and longer [28][30]. Although the total node probability in Equation 5-2 is strictly correct in practice the dynamic range of the value is compressed via the application of a length dependent exponent to the probability estimate

$$\tilde{p}(n|\boldsymbol{O}, \Lambda) = p(n|\boldsymbol{O}, \Lambda)^{[1/N_f]} \tag{5-3}$$

where N_f is the total number of observations (frames) in the utterance.

5.2 Estimating Term Presence Directly from Node Probabilities

Computing the node probability is easy but computing the probability of a particular *term* is nontrivial since it can occur many times within a lattice. Reworking Equation 5-2 to compute the probability that a term v is in the transcription we obtain:

$$P(v|\boldsymbol{O}, \Lambda) = \frac{\sum_{\hat{S} \in \boldsymbol{S}_v} p(\hat{S}|\boldsymbol{O}, \Lambda)}{\boldsymbol{M}_0 + \boldsymbol{M}_1}$$

Where $\boldsymbol{S}_v$ is the space of all possible transcriptions that contain the term, obviously an extremely large space of hypotheses, and not necessarily related to the probabilities assigned to paths containing particular nodes in the lattice. We can express this difference in the following way:

$$P(v|\boldsymbol{O}, \Lambda) = \frac{\sum_{\hat{S} \in \boldsymbol{S}_{0,v}} p(\hat{S}|\boldsymbol{O}, \Lambda) + \sum_{\hat{S} \in \boldsymbol{S}_{1,v}} p(\hat{S}|\boldsymbol{O}, \Lambda)}{\boldsymbol{M}_0 + \boldsymbol{M}_1} = \frac{\boldsymbol{M}_{0,v} + \boldsymbol{M}_{1,v}}{\boldsymbol{M}_0 + \boldsymbol{M}_1}$$

Where $\boldsymbol{S}_{1,v}$ is the space of hypothesized transcriptions containing term v that were investigated by the recognition system and $\boldsymbol{S}_{0,v}$ is the space of hypothesized transcriptions containing term v that were pruned by the search. The total probability mass for these spaces are defined $\boldsymbol{M}_{1,v}$ and $\boldsymbol{M}_{0,v}$ respectively. Although we can know $\boldsymbol{M}_1$ and $\boldsymbol{M}_{1,v}$ by investigating the lattice, we can never know the values for $\boldsymbol{M}_0$ or $\boldsymbol{M}_{0,v}$, and the latter is most certainly dependent on the term in question. A similar method to compensate for the missing probabilities as used in the previous section will be used:

$$\tilde{P}(v|\boldsymbol{O}, \Lambda) = P(v|\boldsymbol{O}, \Lambda)^{[1/N_f]}$$

Since the formulation of term presence requires, strictly speaking, a single probability reflecting the possibility that a particular term occurred at all, we must combine the multiple sources of evidence coming from each instance of the term in the lattice. The first technique we will consider is strictly probabilistic, in that it models each individual occurrence of a term as independent, and attempts to find the overall probability that any one of them occurs. In the second technique, the maximum value of the individual occurrences of each term is used.

5.2.1 Term Presence Based on the Minimum Probability of Occurrence

If each occurrence of a term in a lattice is considered an independent event we can compute the probability that at least one of them is correct in the following way. Every instance k of a particular term in the lattice for each document d is assigned a relative probability of $p_{d,v,k}$ using the node probability equation Equation 5-3, and the total number of occurrences is n_k and the probability at least one of the terms has actually occurred is:

$$p_{n \geq 1}(d, v) = 1 - \prod_{k=1}^{n_k} (1 - p_{d,v,k})$$

The term precision and recall on the test set when using the estimate $\hat{i}_{d,v} = p_{n \geq 1}(d, v)$ is shown in Figure 5-2. The performance of the normal estimator is poor and even the oracle performance is significantly worse than the Top-1 term presence estimator.

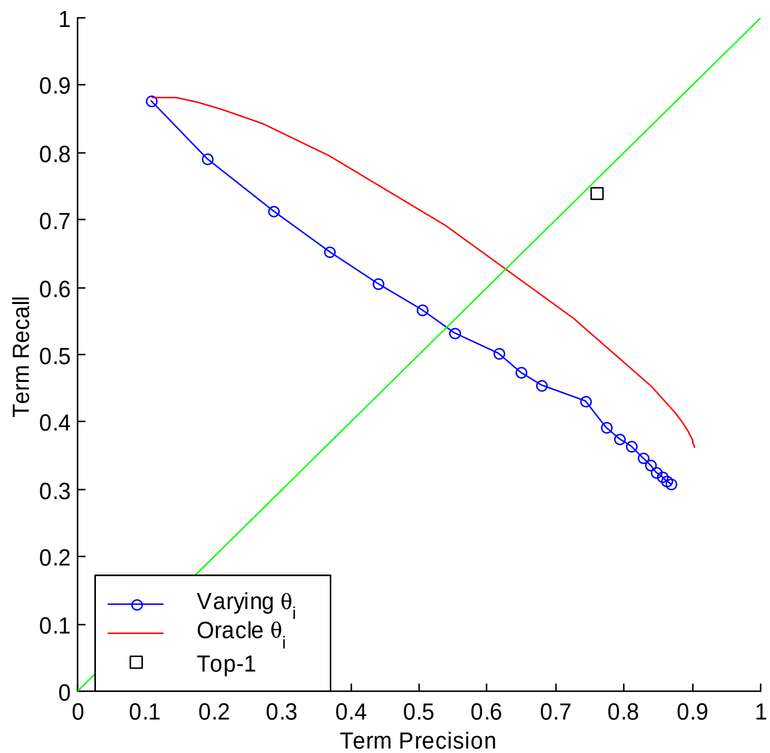


Figure 5-2 Term precision and recall using estimates based on the minimum probability of term occurrence.

5.2.2 Term Presence Based on the Maximum Probability of Term Occurrence

A slightly different way of assigning the term probability to a multiply occurring term is to simply take the maximum occurrence probability among all those in the lattice for a term is used instead:

$$P_{\max}(v) = \max(p_{v,k})$$

Figure 5-3 shows the term precision and recall using the relative probabilities. The situation has not improved visibly from the previous criterion.

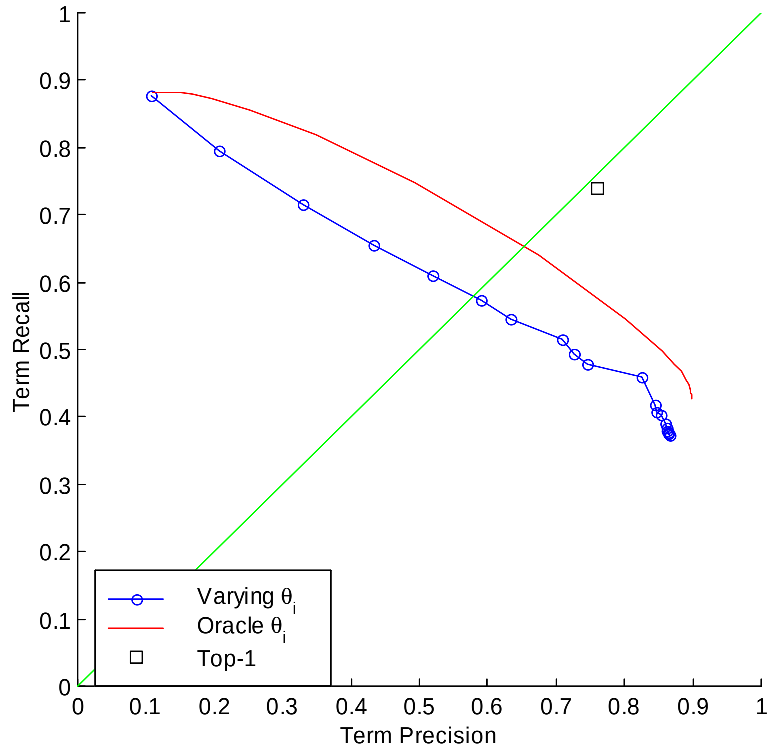


Figure 5-3 Term precision and recall using estimates based on the maximum probability of term occurrence.

Both methods of term presence that used node probabilities directly resulted in significant decreases in performance for the term presence estimator when compared to the Top-1 estimator. It is possible that the technique used to compensate for the missing probability mass was unsuccessful and resulted in overly optimistic or pessimistic predictions regarding node presence. In the following section we consider using the relative ranking of node probabilities instead of their actual values.

5.3 Estimating Term Presence from Ranks of Node Probabilities

The previous section showed that the total node probabilities were not reliable in predicting term presence and so cannot be used directly in the estimates. In the absence of reliable evidence regarding the missing path probabilities, the only value of the total node probabilities is in their relative ranking. Producing the Top-1 hypothesis relies on the logic that acquiring the highest scoring path is an effective way of gathering evidence about term presence and term count. In a similar way, scoring the total probability for each node, and then ranking them either with their competing nodes (locally) or with the nodes over the entire lattice (globally) offers a way of using the scores without relying on their absolute values.

5.3.1 Global Rank of Node Probability

Figure 5-4 shows the term precision and recall for an estimate of term presence based on a global rank of the node probabilities. In this estimator the nodes were ranked with all the nodes in each recognition subsegment of the document and assigned a rank value (duplicating for ties) with 1 being the highest ranking node. When a term occurred in more than one location, the best ranking node was chosen to represent it. In a sense this finds the best ranking node for each term in the document.

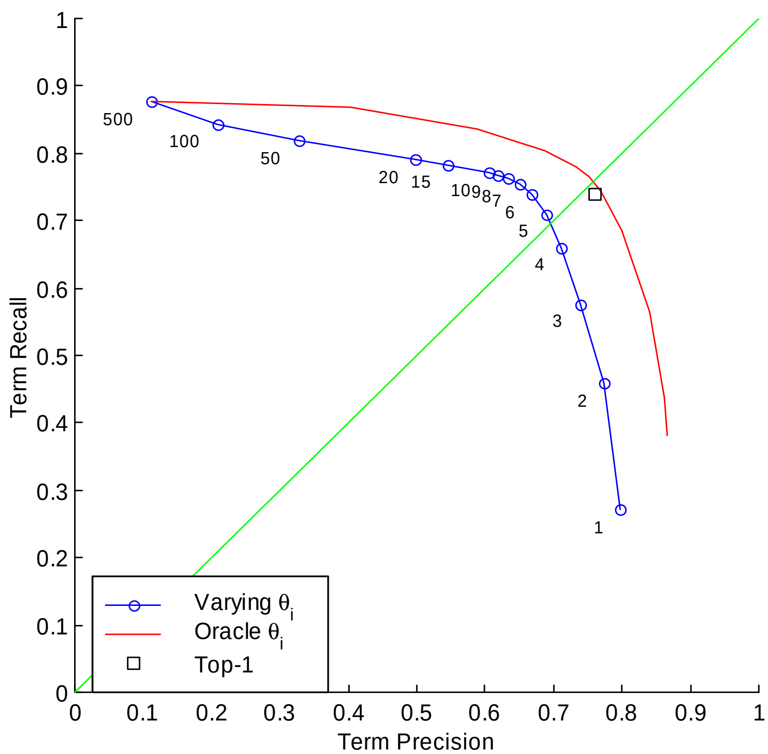


Figure 5-4 Term precision and recall using an estimate of the term presence derived from the relative rank of competing terms in the lattice. The labelled points are the maximum rank threshold.

We can see that a ranking threshold between 4 and 10 yields a range of term precision and recall that is more competitive but still shy of the results achievable with the Top-1 hypothesis. We also see that even with perfectly chosen threshold for each document, the oracle results are still only marginally better than the Top-1.

5.3.2 Rank of Node Among Competitors

An obvious flaw of the previous metric is that there is no accounting for the number of competitors for each node since the rank was measured blindly over each recognition utterance. A refinement is to find all nodes in competition with each node and find the average rank for the node along its duration. This results in fractional rankings since there are occasions where a node will be in competition with another only during a portion of its duration. Before, the best-ranking node for a term is used to predict its occurrence. Figure 5-5 shows the term precision and recall for the term presence based on the average ranking among competitors for the nodes in the test set, with a noticeable improvement over the global ranking strategy used previously. Although the result of the oracle experiment has not changed visibly we can see that at a threshold of 1.4 the term precision and recall is almost identical to that of the Top-1 hypothesis.

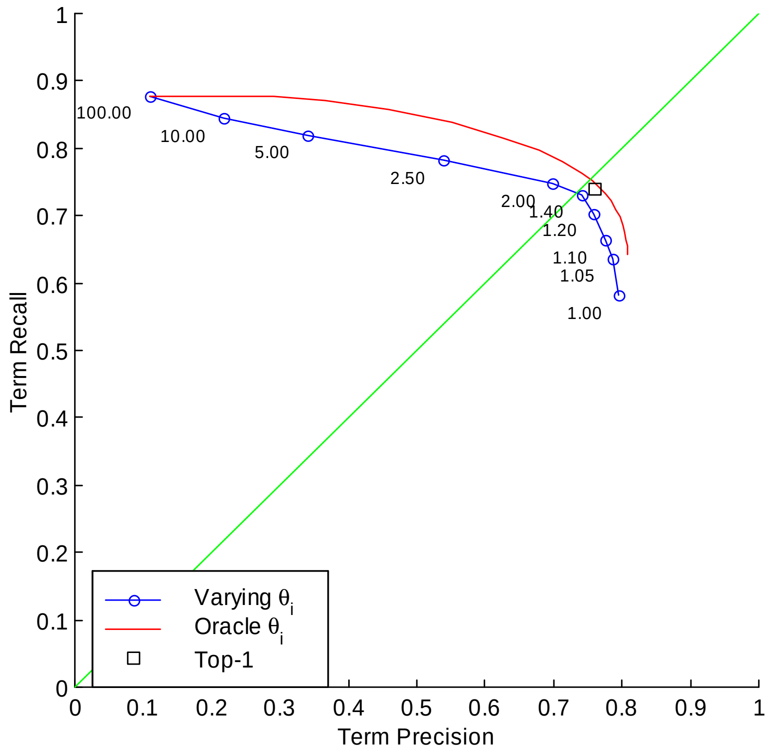


Figure 5-5 Term precision and recall using an estimate of the term presence derived from the average relative rank with respect to competing terms in the lattice. The labelled points are the maximum rank threshold.

5.4 Estimating Term Counts from Ranks of Node Probabilities

So far we have only considered the estimation of term presence from lattice node probabilities. The situation is somewhat more difficult in the estimation of term counts because we have discarded the paths and therefore have no constraint on the string of terms that can be selected from the lattice. However, we have also made the assumption that the hypothesized term counts are statistically independent of one another which simplifies the procedure.

Considering each instance of a term in the lattice at a node to be an independent Bernoulli trial indicating the presence or absence of a term at that location, the estimate of the total number of occurrences of a term is the sum of independent Bernoulli trials. The value of p, the probability of success in this case representing the presence of a term, can be estimated for each node independently by observing the relationship between the ranking of that node and the probability of occurrence for the term.

As discussed in Section 3.5 we are interested in estimates for the expected value of the term count and not in the exact distribution of its values due to the sum of Bernoulli random variables. Since the expected value of the sum of random variables is a the sum of their expected values, we can compute the expected term count in the following manner. First, empirically find an approximate relationship between node rank and term presence $\hat{i}_{d,v} \approx f(\min(r_{d,v,k}))$ through observations on training data, where $r_{d,v,k}$ is the relative ranking of the k^{th} occurrence of term v in document d. Then, compute the expected term count by taking the sum of these probabilities over the multiple occurrences of the term in the lattice assuming that each observation of the term in the lattice at a node is independent:

$$\hat{c}_{d,v} = \sum_{k = 1 \ldots n_k} f(r_{d,v,k}) \tag{5-4}$$

Figure 5-6 shows the probability of term occurrence versus the best *global* ranking of a node for that term computed from the document set. It is apparently neither linear, exponential, logarithmic, nor hyperbolic but can be approximated with a piece-wise curve made of a linear and hyperbolic fit. Figure 5-7 shows the probability of term occurrence versus the best *local* ranking of a nodes for that term. It is clear that the relationship decreases more rapidly than in the global ranking since the number of compared nodes for the local ranking is far smaller.

The distribution of reference term counts for the estimates using Equation 5-4 based on globally and locally ranked node probabilities are shown in the left and right panels of Figure 5-8. Clearly a strong linear relationship is present because the piece-wise model is successful. A summary of term count error metrics for both estimators in Table 5-1 shows that the estimates of term count derived from the local rank of probability are superior to the estimate from the Top-1 hypothesis in both term error and correlation.

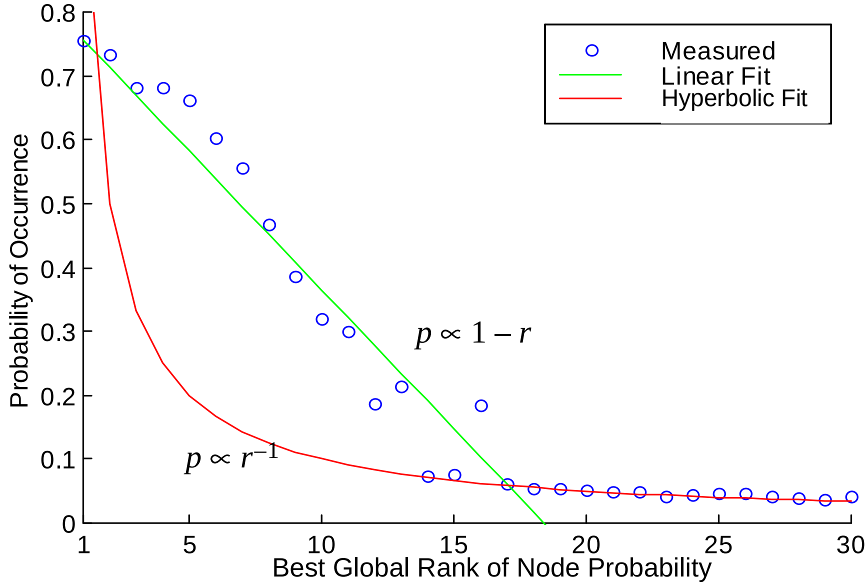


Figure 5-6 Relationship between rank of node probability among all nodes in the lattice and the probability that the term occurring at that node exists in the reference transcript. From rankings of 1 to 17 it is roughly linear, and above 17 it is hyperbolic.

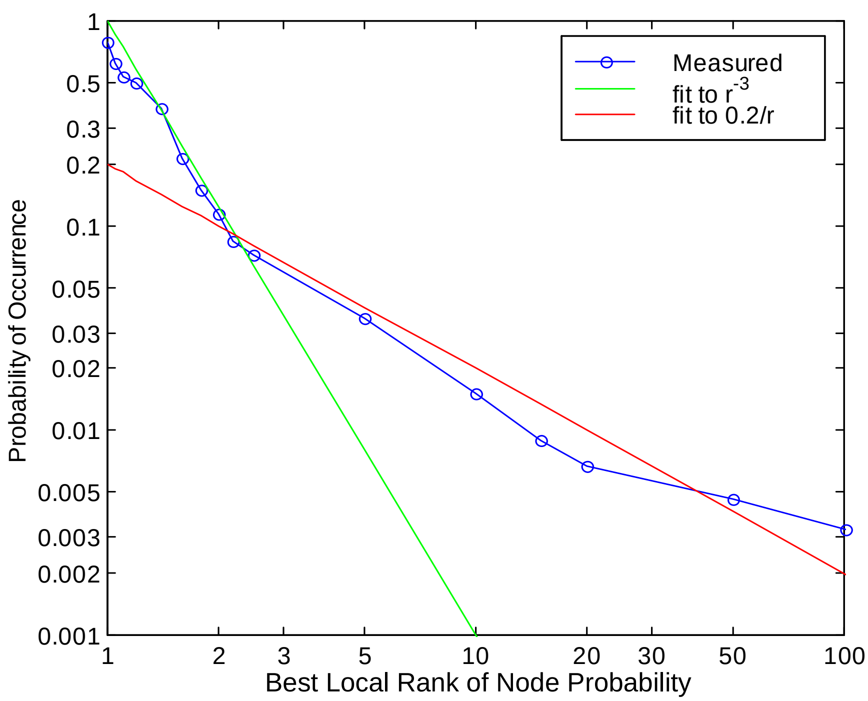


Figure 5-7 Relationship between local rank of node probability among competing nodes and the probability that the term occurring at that node exists in the reference transcript. Note that both axes are logarithmic in this figure.

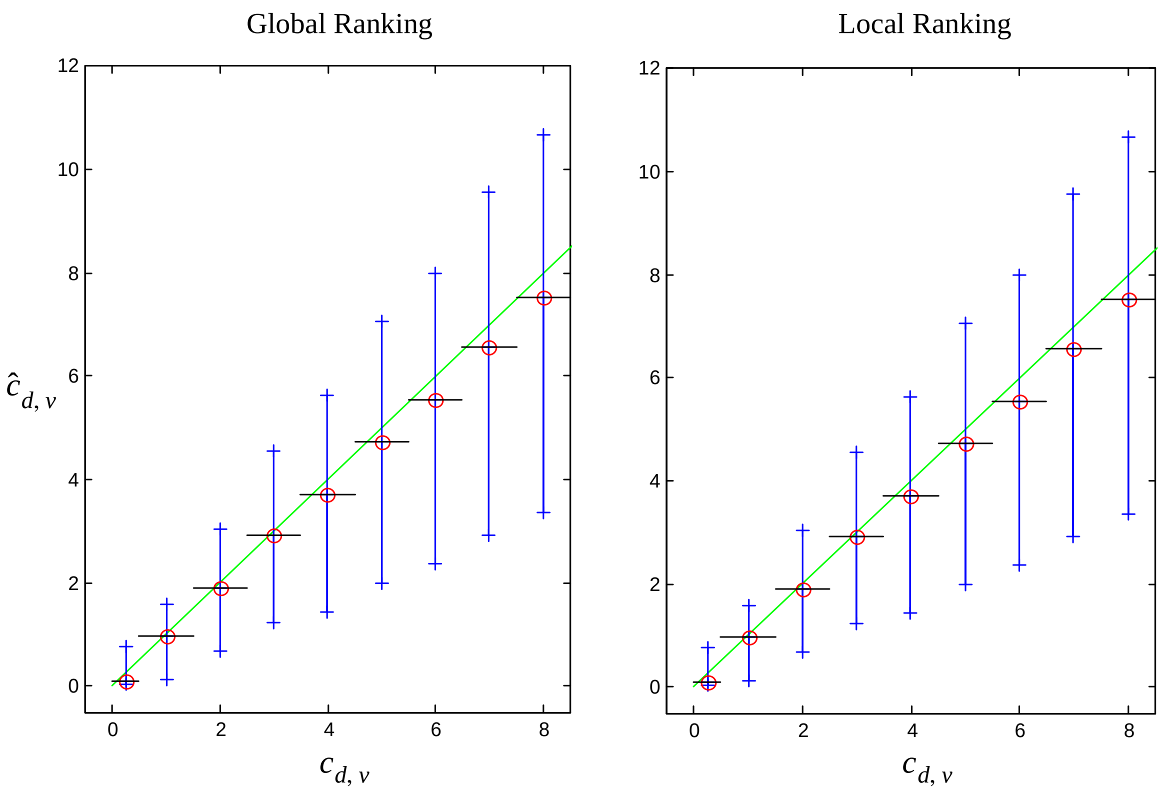


Figure 5-8 Distribution for the estimates of expected term count $\hat{c}_{d,v}$ based on the lattice node probability rankings given the reference term counts $c_{d,v}$. The horizontal bars show the range covered, the circles indicate the means, and the crosses show the 1σ range of values.

Test Set	Term Error: τ	Correlation: ρ
Reference	0.000	1.000
Top-1	0.137	0.330
Global Rank	0.239	0.727
Local Rank	0.136	0.750

Table 5-1 Term error and correlation coefficient measures of the term count averaged over the document set using relative rank of node probabilities in the lattice. The reference and Top-

5.5 Evaluation

The IR precision and recall for reference texts and Top-1 hypotheses are shown in Table 5-2 as a reference whereas Table 5-3 shows the precision and recall using the term presence and term count estimators derived from the relative rank of node probabilities. The global rank is apparently not very useful as an estimator since it has consistently poorer performance than the Top-1 results, but the local rank produces slightly better precision and recall than Top-1 when the document set is large.

Doc. Source	N_d	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade from Ref.
Ref.	326	0.66	0.35	0.56	0.36	0.73	n/a
	652	0.60	0.32	0.50	0.34	0.67	
	1304	0.54	0.29	0.45	0.30	0.59	
	2597	0.45	0.24	0.41	0.26	0.51	
Top-1	326	0.64	0.34	0.53	0.36	0.73	2.2%
	652	0.57	0.30	0.47	0.33	0.67	4.0%
	1304	0.50	0.26	0.41	0.28	0.57	7.3%
	2597	0.40	0.21	0.36	0.24	0.48	9.8%

Table 5-2 Comparison of retrieval performance using reference and Top-1 hypotheses. Precision and recall values for sets with $N_d < 2597$ were generated by constructing subsets of the entire testing set. The last column shows the average degradation (over all 5 points) due to using other than reference texts.

Doc. Source	N_d	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade from Ref.
Global Rank	326	0.59	0.31	0.53	0.36	0.73	5.5%
	652	0.52	0.27	0.45	0.31	0.61	11.4%
	1304	0.44	0.23	0.37	0.26	0.51	16.8%
	2597	0.35	0.19	0.29	0.21	0.42	21.8%
Local Rank	326	0.64	0.34	0.52	0.36	0.72	2.8%
	652	0.57	0.30	0.48	0.33	0.66	3.9%
	1304	0.50	0.27	0.40	0.29	0.58	6.1%
	2597	0.42	0.23	0.35	0.24	0.49	7.4%

Table 5-3 Retrieval performance for term presence and term counts estimators using node rankings from the lattices.

To see whether the improvements are due to the improved term presence or the term count, a new set of retrieval runs was executed where the new estimators were added incrementally. In these experiments the local rank estimator was used for the full document set of 2597. Table 5-4 shows the improvements in precision and recall due to the estimators of term presence and term count, but it is hard to say whether there is any significance to these values.

Term Presence	Term Count	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade Ref.	Improve Top-1
Top-1	Top-1	0.40	0.21	0.36	0.24	0.48	9.8%	0%
Local	Top-1	0.42	0.23	0.35	0.24	0.48	7.8%	17%
Top-1	Local	0.42	0.23	0.35	0.24	0.48	7.8%	17%
Local	Local	0.42	0.23	0.35	0.24	0.49	7.4%	24%

Table 5-4 Using mixed sources of information for estimating term presence and term count for $N_d = 2597$ from Top-1 hypotheses and local rank estimators. The final column shows improvement in average IR quality over using the Top-1 hypothesis.

5.6 Discussion and Summary

The premise of this chapter was that the very large quantity of hypotheses in the lattice afforded us an opportunity for detecting more content in comparison to simply selecting the highest scoring recognition hypothesis. A great deal of effort was required in order to extract a meaningful measure of the relative merits for each hypothesized term in the lattice, and in the end a total of 24% improvement in overall IR quality compared to Top-1 hypothesis was obtained using the best estimators.

One goal was to derive a measure of the probability that each term in the lattice occurs in the reference text using a function of the recognition scores assigned by the decoder. An important observation was that we can acquire this measure through disregarding the specific paths in the lattice and summing all recognition scores that lead into a node. In a sense it allowed the retrieval process to concentrate on each individual hypothesized term in the lattice without concern for the massive number of paths containing the node. This is a substantial savings in computational cost compared with the investigation of the relative probability for every path in every node. In addition it allows the potential information retrieval content of a speech recognition lattice to be represented simply as a list of nodes and their relative probabilities. Such a list is vastly smaller than the original lattice and also fits into the vector-based structure of the retrieval engine.

The reality of the node probabilities is that the missing mass $\boldsymbol{M}_0$ in Equation 5-2 is unknown and therefore we can only derive a coarse estimate of relative probability. Consequently, we are reliant on measures that are insensitive to the denominator in this equation and such as the rank. Indeed the relative rankings of node probabilities far outperformed the value of the relative node probabilities themselves when the term precision and recall are compared for each of these estimators.

In ranking the node probabilities both global and local contexts were explored. Although they appeared to have identical oracle term precision and recall, in practice the local rank was superior in estimating the term presence, term count, and in overall IR quality. This is consistent with the desire to isolate each node as a separate event in the lattice that is only dependent on its immediate competitors for total recognition score.

However, in order to estimate the actual term presence and term count an empirical relationship between the node probability rankings and the probability of occurrence was necessary. This relationship required a piecewise model, an undesirable necessity since a new model will have be computed every time the database changes. Such a model is not likely to be a general solution for any generic spoken document retrieval system unless a quantity of adaptation material was available to train these models. In the next chapter another speech recognition data structure, the N-Best list, is used to estimate the term presence and term counts in a more general way.

6. Extracting Relevant Content from N-Best Lists

In the last chapter we only considered the paths in the lattice in computing an isolated node probability with which to estimate term presence and term count. After the total node probability was obtained any sense of path continuity between nodes was discarded. Ultimately it was the ranking of node probabilities among competitors that yielded usable information for estimating the term presence and term count.

In this chapter we explore ways of using the N-Best list, a small subset of the hypotheses contained in the lattice that is found by selecting those that contain the highest recognition scores (see Section 2.1.3). To produce the N-Best list the rescoring tool selects hypotheses of successively lower recognition score by gradually substituting nodes that are in lower scoring paths. This is reminiscent of the technique used to measure the local ranking of node probabilities in the following way: If a node is consistently in paths that have far better scores than its competitors the number of the Top-N paths containing the competitors will be proportionally smaller. Similarly if the nodes in a competing set are equally likely they will appear in equal proportion throughout the Top-N paths. At any rate, it is not only the relative ranking of the node probabilities that matters but how far apart the recognition scores of its competitors are compared to other nodes in the lattice and their competitors.

Although the relative ranking of the Top-N hypotheses is intended to reflect the relative ranking of the accuracy of the hypotheses, there is no guarantee that the recognition scores can be interpreted as direct indications of hypothesis accuracy let alone relevant content. The curve shown in Figure 6-1 is a rough sketch of the relationship for a typical utterance illustrating that a fair number of hypotheses are all equally likely to be the correct transcription. When selecting the hypothesis with the highest posterior probability we are applying a decision rule that maximizes the likelihood of correct transcription. For a dictation machine this model is appropriate since the ultimate goal is to produce a single hypothesis with the minimum error, but in the case of information retrieval there is no equivalently objective notion of "best."

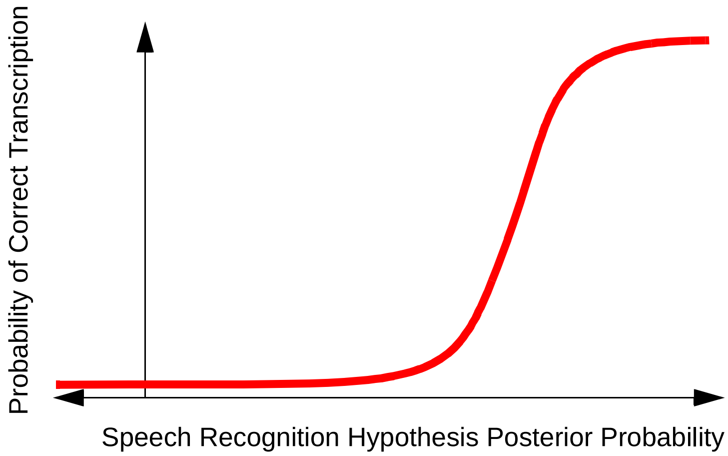


Figure 6-1 Model of the relationship between speech recognition posterior probability and probability of correct transcription for all hypotheses. Most paths are unlikely to be correct, with a small number possibly correct, and an indeterminate transition region.

6.1 Problems With the Traditional N-Best List

Although for retrieval we desire a hypothesis with zero error and know such a hypothesis could exist (one identical to the reference text) we are confident that this hypothesis is not necessarily at the most extreme right of the curve. This is because the model of speech recognition is not accurate enough in a large vocabulary system, especially due to the relatively small amount of training material. In addition the pruning algorithms guarantee that the search space is incomplete leaving the possibility that the correct transcription is never hypothesized [27].

The model for selecting hypotheses for information retrieval should instead focus on extracting the largest number of correct content bearing terms possible (high term recall) while minimizing the number of incorrectly hypothesized content bearing terms (high term precision.) Term recall and precision are used instead of word recall and precision because the text processing outlined in Section 2.2 maps many words into one term and removes some words outright. Since the relative rank of the hypotheses by posterior probability does not imply they are similarly ranked by term recall or precision, a subset of hypotheses rather than the highest ranking hypothesis should be chosen.

The N-Best hypotheses is one approximation to acquiring the set of paths, all approximately equal in probability, that contain the highest term precision and recall. However, it is important that this list be generated in a way that is consistent with the usage of its contents in information retrieval. As described in Section 2.1.3 the recognition system first generates a lattice containing a subset of all possible paths and then scans these paths in an efficient manner generating paths with successively lower scores. When the requisite number of paths are explored (the “N” in N-Best) the procedure ends. The first ten entries of a typical N-Best list is shown in Figure 6-2.

<sil> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> it is your **money**

<lip> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> it is your **money**

<sil> <lip> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> it is your **money**

<sil> just **ahead** <inh> why are **americans pay** more than a <sil> **billion dollars** so much for **sugar** <sil> it is your **money**

<sil> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> is is your **money**

<sil> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** to much for **sugar** <sil> it is your **money**

<lip> just **ahead** <inh> why are **americans pay** more than a <sil> **billion dollars** so much for **sugar** <sil> it is your **money**

<lip> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> is is your **money**

<lip> just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** to much for sugar <sil> it is your money

<sil> just **ahead** <sil> <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> it is your **money**

just **ahead** <inh> why are **americans pay** more than a <sil> **million dollars** so much for **sugar** <sil> it is your **money**

Figure 6-2 Excerpt of a typical N-Best, as would normally be extracted from the speech recognition lattice. Words in <> are placeholders for common noises (such as lip and inhale) and detected silence periods. Boldfaced words are content bearing, and the only remaining words after text processing. Only the first and fourth hypotheses are distinct.

What is noteworthy about this N-Best list is the small amount of variation in the content bearing words, highlighted in boldface, and the fact that from this list of ten only the first and fourth remain distinct after text processing. Clearly, the N-Best extraction procedure must be tailored to suit the information retrieval structure if we are to extract more potential content.

6.1.1 A Better N-Best List

If only those hypotheses that differ in their post-processed content are kept a new N-Best list containing a far more richer collection of hypotheses is produced. Figure 6-3 shows the first ten entries of the N-Best list using the same lattice of the previous example. All of the hypotheses differ in content bearing words as that was a criterion for selecting them. On average $\sim N\sqrt{N}$ unprocessed paths must be explored to generate a total of N unique processed hypotheses for the test set, or approximately 32000 to generate a total of 1000 unique hypotheses. The outcome of this new list generation is that 1000 unique N-Best items appear for deriving term content whereas before the effective size was only $1000/(\sqrt{1000}) = \sim 32$.

```
ahead america pai million dollar sugar monei
ahead america pai billion dollar sugar monei
ahead america pai million dollar sugar babi monei
ahead america pai million dollar sugar cane monei
ahead america pai million dollar shorter monei
ahead america pai million dollar sugar beet monei
ahead america pai million dollar sugar heat monei
ahead america pai million dollar show monei
ahead america pai billion dollar sugar babi monei
ahead america pai billion dollar sugar cane monei
```

Figure 6-3 An excerpt of a typical N-Best list when only hypotheses differing after text processing are used. All the words are content bearing in this list since they are processed already and the entire list in the previous figure is represented in the first two hypotheses.

6.1.2 An Even Better N-Best List

One point that was glossed over in the discussion of N-Best list production is that the speech recognition system breaks spoken documents into several segments in order to limit their length (see Section 4.1.) In this work, the documents were broken into segments of approximately 10 seconds depending on the presence of silence regions. Because the lattices are produced for each segment they have to be concatenated in order to form a lattice for the complete document and some documents contain well over a hundred lattices. Since the process of generating the N-Best list requires memory proportional to n^2 where n is the number of nodes in the lattice, the list can take an exceedingly long time to produce.

To avoid this effort, a method was designed to create N-Best lists for each segment separately and combine them together in a composite N-Best list that accounts for the relative scores of each hypotheses in each segment. Figure 6-4 shows how this method uses only a portion of each N-Best list, in order to achieve an overall size with approximately the count of interest. The remaining hypotheses are discarded.

A second method dispensed with the resorting process altogether and just combined the hypotheses until all of them were exhausted. Figure 6-5 shows this effect. The average number of paths increases tremendously when this is done from 1000 to a list on the order of 10^{20}. Obviously, the actual hypotheses list need not be generated since the permutations of all the hypotheses and their scores can be generated directly from the set of N-Best lists.

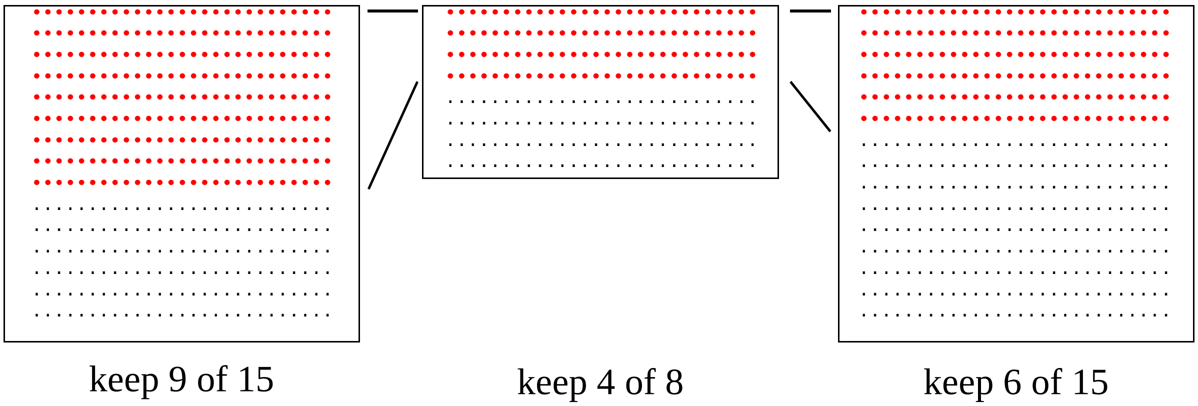


Figure 6-4 Producing an N-Best list from segments of a document, where the overall N-Best is held constant. Total N-Best list has length 9 x 4 x 6 = 216.

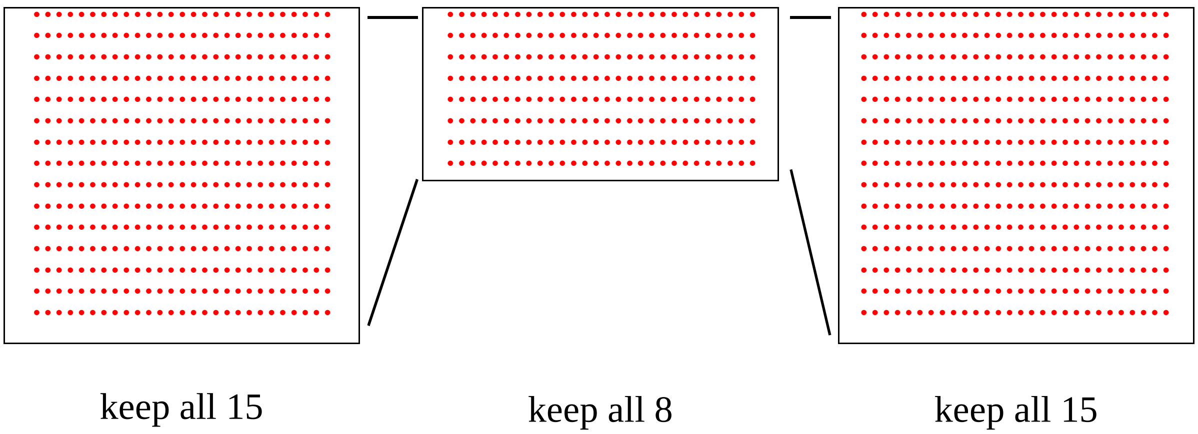


Figure 6-5 Producing an N-Best list from segments of a document, where the overall N-Best is not held constant. Total N-Best length is 15 x 8 x 15 = 1800.

6.2 Primary Assumptions

Now that we have two options for generating N-Best lists the question remains: how do we extract probability distributions of term presence and term count from them? In what way can we observe the occurrence and number of words in each hypothesis, and derive a meaningful estimate of probability models? Before this can be done, four critical assumptions need to be made to simplify the computation:

6.2.1 N-Best List is a Population

The N-Best list is treated as a population of the possible hypotheses. All hypotheses not listed in the N-Best list are assigned a probability of zero, similar to assigning $\boldsymbol{M}_0 = 0$ in Equation 5-2.

6.2.2 N-Best Hypotheses are Equiprobable

While the speech recognition assigns a log-probability to every path in the N-Best list, there is some question as to whether this value can be used to rank the hypotheses in order of decreasing predicted accuracy. Despite the fact that it is effective in selecting the hypothesis with the lowest error, the recognition score does not obviously distinguish the relative merits of one hypothesis over another. A safe assumption is that all of the hypotheses above some threshold are approximately equally capable of estimating the content of the spoken utterance.

6.2.3 Hypothesis Independence

For each document d, all hypotheses in the N-Best list are independent, identically distributed composite events containing a vector of hypothesized term presences $\hat{i}_{d,v}$ and term counts $\hat{c}_{d,v}$ defined over the space of terms v. The distribution for the hypotheses is completely defined by the distribution of the term presences $P_{\hat{I}_{d,v}}(i)$ and term counts $P_{\hat{C}_{d,v}}(c)$, and the parameters for these distributions are the reference term presence and term count.

6.2.4 Term Independence

For each document d, $P_{\hat{I}_{d,v}}(i)$ and $P_{\hat{C}_{d,v}}(c)$ are independent across terms $v = 1 \ldots n_v$. This is perhaps the most crucial of the assumptions, because it allows the modeling of the term presence and term count for each term separately. The joint model of the distribution of all the terms would be an extremely high-dimensional problem, one for each term. As noted in Section 3.4.1 the relevance equation is composed of an extremely small number of terms, being the sum of the term significance-weighted term counts that are in the queries, divided by the document length. The hypothesized relevance, as described in Section 3.4.3, will be assumed to have a Gaussian distribution.

6.3 Extracting Term Presence and term counts

With these assumptions in mind the approaches to discovering the hidden probability models for $P_{\hat{C}_{d,v}}(c)$ and $P_{\hat{I}_{d,v}}(i)$ are straightforward. For each of the N text-processed hypotheses, there are values for term presence $\hat{i}_{d,v}$ and term count $\hat{c}_{d,v}$ that can be computed directly, either 0 or 1 for term presence and a nonnegative integer for the term count. Although it is likely that the set of observations for each term v are interdependent, there is no appropriate place in the vector space model to insert these dependencies. This is an obvious limitation of the vector space model when trying to incorporate multiple hypotheses in the relevance estimation.

6.3.1 Term Presence

The collection of the term presences for each term v into a vector is used in the computation of the document term significance $\boldsymbol{W}_d$, but is not retained after this computation. As described in Equation 3-4, the probability of the term presence is the basis for the computation of the mutual information between the document set and the term. Therefore, simply determining the approximate probability of the term presence for each word is sufficient to meet the requirements for the relevance equation, and is easily extracted from the N-Best lists in the following manner:

For each hypothesis n the term presence is $i_{d,v}^{(n)}$, and is Boolean valued. The probability that a term occurs in a randomly selected hypothesis is approximated:

$$\hat{P}_{I_{d,v}}(1) \approx \frac{1}{N} \sum_{n=1\dots N} i_{d,v}^{(n)} \qquad \hat{P}_{I_{d,v}}(0) = 1 - \hat{P}_{I_{d,v}}(1) \tag{6-1}$$

In the table shown in Figure 6-6, the performance of this estimator is shown over the test set. In each row, a range of estimated term presence values are compared with the fraction of the terms having these probabilities that actually occurred in the reference text. Even if a term is in every hypothesis, the probability that it is in the reference is still only 77.1% for $N = 1000$, but a much higher 90.7% for $N \sim 10^{20}$. The requirement is obviously more demanding when there are an enormous number of hypotheses. Similarly, a prodigious number of hypotheses can ensure that more of the reference terms actually appear in at least one hypothesis, as the first row shows the number of unhypothesized terms.

The table also shows the theoretical upper bound for term precision and term recall if a simple threshold-based decision rule was used with this estimator to decide (Boolean) on the presence or absence of a particular term. The first row reveals a term recall upper bound of $(100-24.1)\% = 75.9\%$ for $N = 1000$ and $(100-17.6)\% = 83.4\%$ for $N \sim 10^{20}$. The last row shows a term precision upper bound of 77.1% for $N = 1000$ and 90.7% for $N \sim 10^{20}$. The strength of the term presence estimate is also illustrated in the curve in the right half of Figure 6-6 revealing an approximately linear relationship.

The effectiveness of such a decision rule can be more clearly seen in Figure 6-7, where both actual and oracle performing thresholds are used in the rule. The square shows the performance for the Top-1 hypothesis. The performance point at which the two curves have equal precision and recall is approximately 74% for both sets of N-Best lists. In the oracle experiments, the point at which the precision equals the recall is approximately 77%, which is marginally greater than in the non-oracle experiment. Although we could select this point as representative of the overall performance of this estimator, we have not yet demon-

range of $\hat{P}_{I_{d,v}}(1)$		$E\{i_{d,v}\}$	
		N=1000	N~10^{20}
0.00		*0.241*	*0.176*
0.00	0.10	0.031	0.022
0.10	0.20	0.077	0.063
0.20	0.30	0.125	0.115
0.30	0.40	0.166	0.174
0.40	0.50	0.196	0.236
0.50	0.60	0.242	0.320
0.60	0.70	0.352	0.399
0.70	0.80	0.373	0.486
0.80	0.90	0.523	0.559
0.90	1.00	0.710	0.707
1.00		0.771	0.907

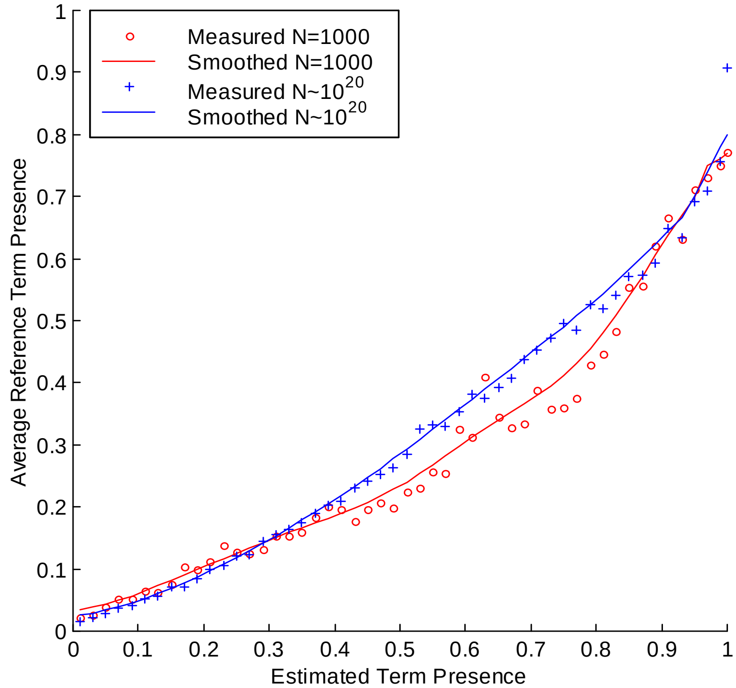


Figure 6-6 Estimates of the term presence derived directly from the N-Best lists, evaluated over the entire test set. The two left columns indicate the range of the estimate, where the right column is essentially the fraction of these that actually occurred. In the first row, the value indicates the fraction of reference terms that are never hypothesized.

strated that there is a particular trade-off between recall and precision that is optimal for the IR component of the system. One very notably feature of the performance curves is that a very large N-Best list has a better term precision or recall only at the extreme levels of recall or precision.

6.3.2 Term Counts

The collection into a vector of the term counts for each term is already performed in the relevance computation and placed into each column of the $\boldsymbol{D}$ matrix known as $\boldsymbol{D}_d$ (see Equation 2-2.) For the purposes of computing the probability distributions, an ensemble of matrices, one for each of the N hypotheses, will be formed in $\boldsymbol{D}_d^{(n)}$, with d for each document in $[1, n_d]$. The elements of $\boldsymbol{D}_d^{(n)}$ are the term counts for hypothesis number n in document number d.

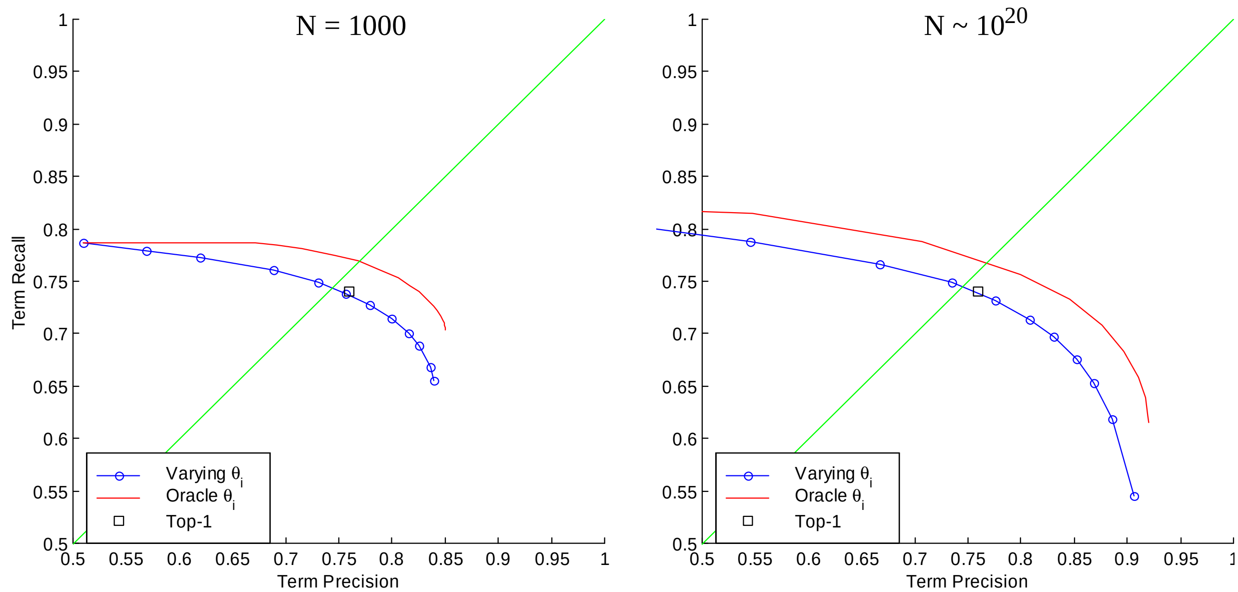


Figure 6-7 Term precision and recall using an estimate of the term presence derived directly from the N-Best lists, when N = 1000 on the left and N ~ 10^{20} on the right. Each circle is a test for word detection of the form $\hat{P}_{I_{d,v}}(1) > \theta_i$ where θ_i varies from 0.0 to 1.0. The Oracle curve shows the best possible performance if the threshold is set to the ideal value for each document. The square is the performance for the Top-1 hypothesis.

To compute the estimate of the term count from the N-Best hypotheses we assume each hypothesis is equal in probability and that the term counts within each hypothesis are independent of each other. For each hypothesis n the term count is $c_{d,v}^{(n)}$, and given all hypotheses equal probability the expected value of the term count, called the *expected term count*, is computed as:

$$\hat{c}_{d,v} = E\{C_{d,v}\} = \frac{1}{N}\sum_{n=1\ldots N} c_{d,v}^{(n)} \tag{6-2}$$

As described in Section 3.4, this value will be used to predict the term count of the reference transcript and incorporated into the relevance distribution. The distributions of expected term counts over specified ranges of reference term counts are show in Figure 6-8. The range of expected term counts shown covers 99.7% of the terms in the database, although there are some terms with very high counts. Table 6-2 shows that the term error and correlation coefficient for the expected term count derived from the very large N-Best lists is superior to those computed from the Top-1 hypothesis.

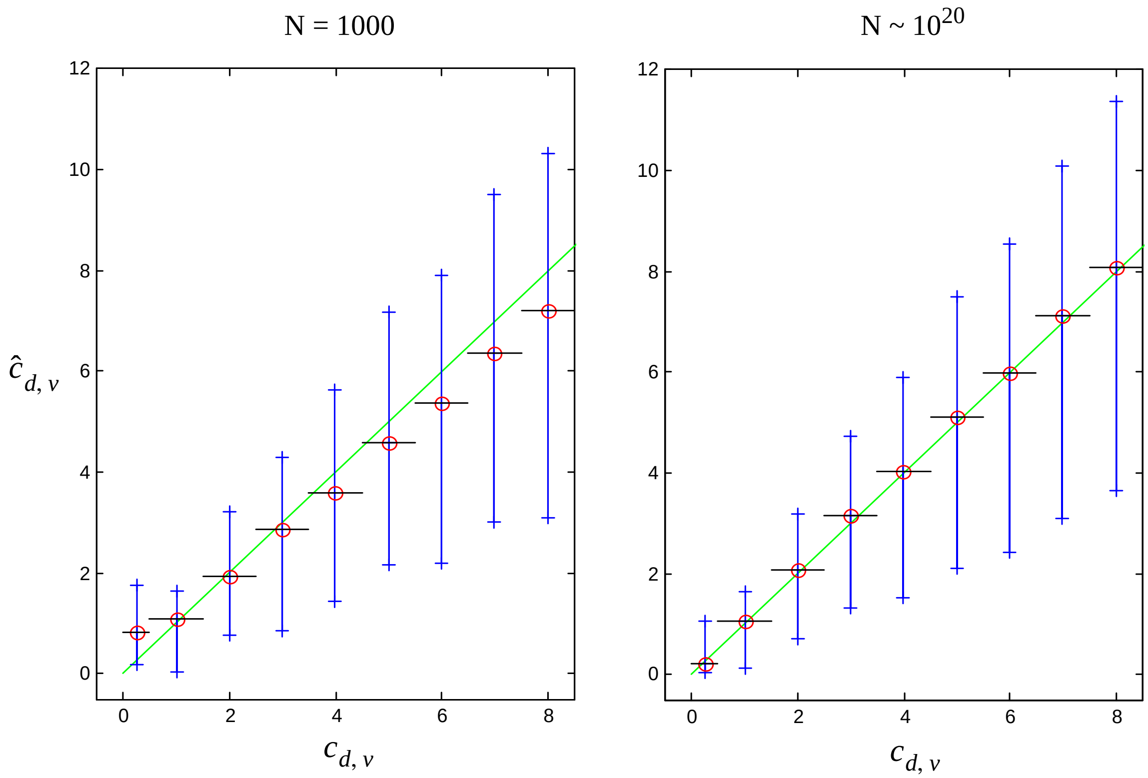


Figure 6-8 Distribution for the estimates of the term count $\hat{c}_{d,v}$ using the expected value from the N-Best lists given the reference term counts $c_{d,v}$. The horizontal bars show the range covered, the circles indicate the means, and the crosses show the 1σ range of values.

Test Set	Term Error: τ	Correlation: ρ
Reference	0.000	1.000
Top-1	0.137	0.330
N-Best, N=1000	0.155	0.583
N-Best, N~10^{20}	0.125	0.725

Table 6-1 Term error and correlation coefficient measures of the term count averaged over the document set using N-Best lists. The reference and Top-1 hypotheses figures are shown as a comparison.

6.3.3 Document Length

Computing the probability model for the document length is a fairly straightforward application of the same technique used to derive the term counts, except that the averages are applied across each hypotheses and not for each term. In this case, there is no need to be concerned with term count independence since we are incorporating the inter-term variability by averaging across the documents instead of the hypotheses.

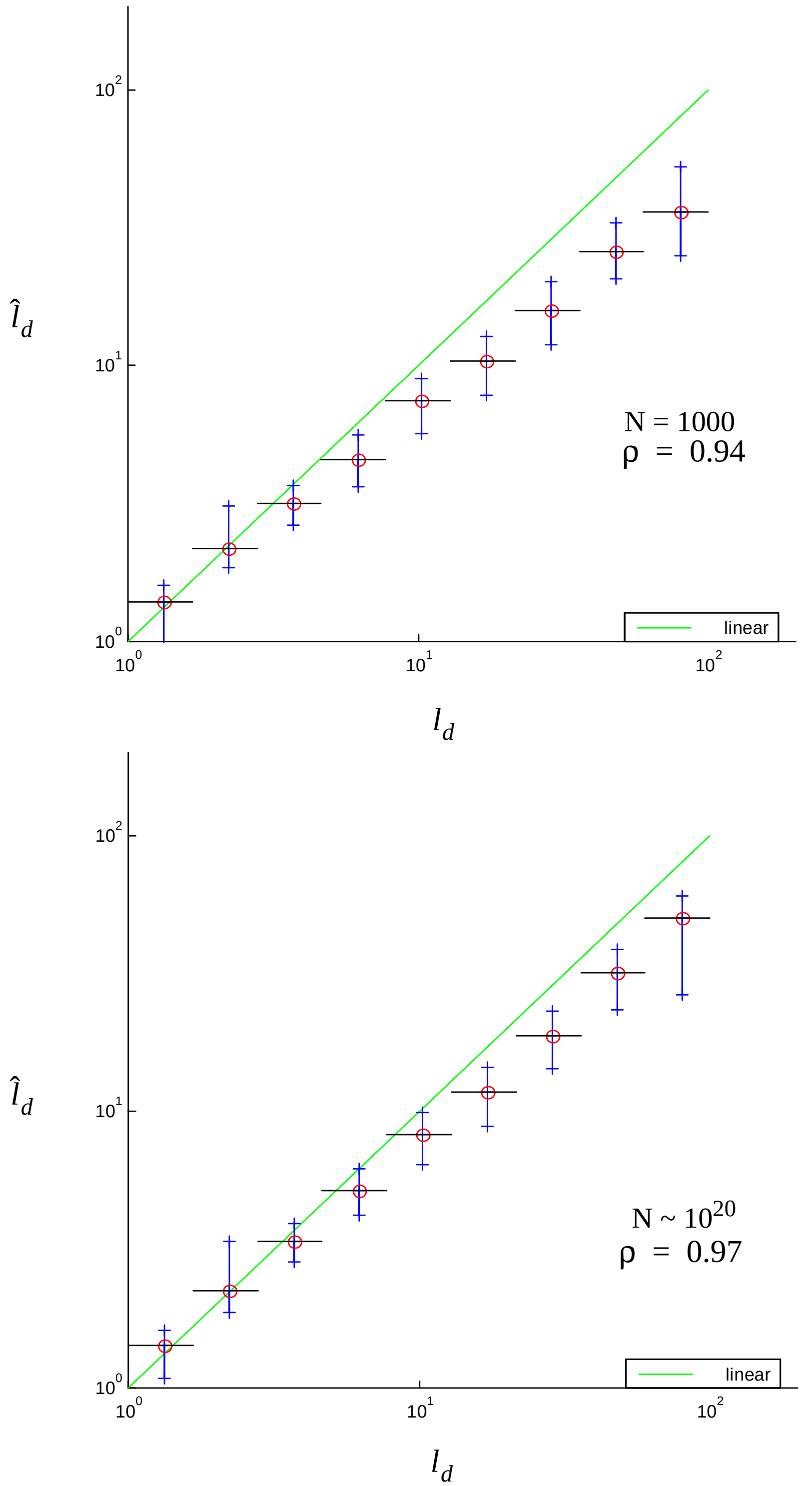


Figure 6-9 Comparison between the document length parameter $\hat{l}_d$ derived from the N-Best lists and the reference document length l_d. The horizontal bars show the range covered, the circles indicate the means, and the crosses show the 1σ range of values.

The Document Length for each hypothesis n is:

$$l_d^{(n)} = \left(\sum_v (c_{d,v}^{(n)})^{\alpha} \right)^{1/\alpha}$$

Computing the expected value over all the hypotheses:

$$\hat{l}_d = E\{ l_d^{(n)} \} = \frac{1}{N} \sum_{n = 1 \dots N} l_d^{(n)} \tag{6-3}$$

Figure 6-9 shows a plot of the reference values for the document length versus this estimated value. Note that the estimate for document length is consistently somewhat lower than the reference values, although the correlation is linear with a value of 0.97.

6.4 Retrieval Experiments

Given estimators for term presence and term count and a formulation for document length, the next step is to evaluate the new estimators in a retrieval experiment and compare them with the precision and recall for reference texts and Top-1 hypotheses shown in Table 6-2. The final column for the Top-1 hypotheses shows the degradation of the overall IR quality in comparison to the reference texts. As shown in Figure 4-4 the degradation is proportional to the number of documents in the test set (N_d).

Doc. Source	N_d	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade Ref.
Ref.	326	0.66	0.35	0.56	0.36	0.73	n/a
	652	0.60	0.32	0.50	0.34	0.67	
	1304	0.54	0.29	0.45	0.30	0.59	
	2597	0.45	0.24	0.41	0.26	0.51	
Top-1	326	0.64	0.34	0.53	0.36	0.73	2.2%
	652	0.57	0.30	0.47	0.33	0.67	4.0%
	1304	0.50	0.26	0.41	0.28	0.57	7.3%
	2597	0.40	0.21	0.36	0.24	0.48	**9.8%**

Table 6-2 Comparison of retrieval performance using reference and Top-1 hypotheses. Precision and recall values for sets with $N_d < 2597$ were generated by constructing subsets of the entire testing set. The last column shows the average degradation (over all 5 points) due to using other than reference texts.

Table 6-3 shows how the new estimators do indeed improve upon the Top-1 hypothesis by reducing the degradation in precision and recall, especially when the number of documents increases. The use of the extremely long N-Best lists seems to outperform the shorter N-Best lists in the long run.

Doc. Source	N_d	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade Ref.
N-Best N = 1000	326	0.65	0.34	0.55	0.38	0.76	0.7%
	652	0.58	0.31	0.49	0.34	0.67	1.7%
	1304	0.49	0.26	0.43	0.30	0.59	4.7%
	2597	0.39	0.21	0.36	0.25	0.50	**8.8%**
N-Best N ~ 10^{20}	326	0.65	0.34	0.55	0.37	0.73	0.7%
	652	0.58	0.31	0.49	0.33	0.66	2.6%
	1304	0.51	0.27	0.43	0.29	0.58	4.4%
	2597	0.42	0.22	0.38	0.26	0.51	**4.3%**

Table 6-3 Comparison of retrieval performance using Reference, Top-1, and N-Best sources for the document. Precision and recall values for sets with $N_d < 2597$ were generated by constructing subsets of the entire testing set. The last column shows the average degradation (over all 5 points) due to using other than Reference texts.

To identify the contributions to overall IR quality from the improved term count and term presence estimators, the Top-1 and N-Best sources were combined. In Table 6-4 the improved estimates from the N-Best lists (where N ~ 10^{20}) are added incrementally with the reduction in overall IR quality degradation shown in the final column. The term count estimator yields the most substantial increase in overall IR quality, contributing 45% of the 63% improvement.

To confirm that these results are not specific to the TREC-7 database, 11,349 stories from the TREC-8 database [46] were used in a similar configuration with 50 queries and 932 documents relevant to at least one query. The speech recognition engine and information retrieval engines were also similar to those used in the TREC-7 database. Table 6-5 shows the improvements in retrieval precision and recall using the N-Best document source as opposed to the Top-1 source for different sized subsets of the full TREC-8 database. For document sets of size 932, 1864, 3728, 7456, and 11349, the fraction of documents relevant to any query in each sized document set is approximately 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{12}$, the first four being the same as those used in the TREC-7 results. In addition to being superior to the Top-1 hypothesis in each document set the use of N-Best lists provided a 53% improvement in average IR quality for the largest set, indicating that the TREC-7 results are representative of potential performance in other databases.

Term Presence	Term Count	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade Ref.	Improve Top-1
Top-1	Top-1	0.40	0.21	0.36	0.24	0.48	9.8%	0%
N-Best	Top-1	0.42	0.23	0.36	0.24	0.48	7.3%	21%
Top-1	N-Best	0.41	0.22	0.37	0.25	0.51	6.1%	45%
N-Best	N-Best	0.42	0.22	0.38	0.26	0.51	4.3%	63%

Table 6-4 Using mixed sources of information for estimating term presence and term count for $N_d = 2597$. The N-Best lists used here have N ~ 10^{20} The final column shows improvement in average IR quality over using the Top-1 hypothesis.

Document Source	N_d	$\frac{1}{2}R$ Prec.	$\frac{1}{2}R$ Rec.	R Prec. & Rec.	$2R$ Prec.	$2R$ Rec.	Degrade Ref.
Reference	932	0.69	0.37	0.60	0.36	0.72	0 %
	1864	0.63	0.33	0.55	0.34	0.68	0 %
	3728	0.59	0.32	0.50	0.31	0.62	0 %
	7456	0.53	0.28	0.43	0.28	0.56	0 %
	11349	0.50	0.27	0.40	0.26	0.52	0 %
Top-1	932	0.66	0.35	0.52	0.31	0.63	9.9 %
	1864	0.63	0.33	0.48	0.30	0.59	7.6 %
	3728	0.56	0.30	0.45	0.27	0.55	7.8 %
	7456	0.51	0.27	0.40	0.25	0.50	11.5 %
	11349	0.45	0.24	0.34	0.20	0.41	16.1 %
N-Best N ~ 10^{20}	932	0.67	0.35	0.53	0.32	0.64	8.4 %
	1864	0.63	0.33	0.49	0.31	0.61	6.0 %
	3728	0.56	0.30	0.46	0.28	0.56	7.7 %
	7456	0.51	0.27	0.41	0.25	0.51	6.3 %
	11349	0.47	0.25	0.37	0.24	0.47	7.6 %

Table 6-5 Results for stories from the TREC-8 database, with comparison of retrieval performance using reference, Top-1, and N-Best sources for the document. The N-Best sources yield an relative improvement in IR quality of approximately 53% over the Top-1 sources in the case where $N_d = 11349$.

6.5 Discussion and Summary

N-Best lists prove to be more capable of resolving the difference between likely and unlikely hypothesized terms than the lattice analysis of the previous chapter. There is great power in the ranking of the hypotheses, and the distance in recognition score between a node and its competitors in comparison to other nodes and their competitors is apparently indicative of the relative merit of the node. In terms of degradation from reference texts the best methods of this chapter produced an average IR quality 63% better than the Top-1 hypotheses, mostly attributable to the improvements in term count. These results were also confirmed on a separate database with four times as many stories, yielding an improvement in average IR quality of 53%.

The fraction of N-Best lists containing a term turns out to be very successful at predicting term presence. This is a satisfying conclusion since much of the literature regarding confidence estimation of hypotheses has used this same feature in estimating word errors [27][28][29][30][31]. The reduction of term error and increase in term count correlation via an expansion of this technique into estimating the term count is also consistent with these findings.

Using N-Best lists that are unique *after* text processing is applied creates a much larger set of hypotheses than the default method of generating N-Best lists, creating on average of $\sqrt{N}$ times as many unique hypotheses. This reveals that we should trust the speech recognition's internal structure in generating and scoring hypotheses, but we should ensure that the stopping criterion for testing hypotheses is tailored toward the task to which they are applied.

Although using evidence from 1000 unique hypotheses did not at first seem to be a conservative choice of size it appears that using as many available hypotheses as possible leads to better estimates of term presence, term count, and better overall IR quality as well. Producing an N-Best list, regardless of length, guarantees that in the selection of successively lower scoring recognition hypotheses the choice of nodes will be made such that the overall recognition score is decreased as little as possible. This was not possible when the path scores were combined and discarded in the methods of the last chapter and appears to be the critical difference between the two methods.

Another advantage of the N-Best lists is that they permit estimates of term presence and term count that can be used directly, obviating the step of modeling the relationship that was required to convert relative rank into term presence and term count in the previous chapter. This is very important because the absence of an extra layer of modeling means that the results are more general and do not require a training or adaptation set in order to apply the technique to a new database.

7. Conclusions and Suggestions for Future Work

Information retrieval from databases of transcription produced by speech recognition requires more sophisticated methods in order to accommodate the uncertainty arising in the speech recognition process. This work addresses that uncertainty by incorporating the multitude of hypotheses explored into the estimation of parameters used for information retrieval, and modifies the retrieval equations so that they can receive the relative probability of these hypotheses.

The degradation in overall precision and recall for documents generated by speech recognition was halved through this research without incurring any significant cost to the retrieval engine and only a minor computational demand in the speech recognition phase. In addition, the methods described do not require an overhaul of the dominant set of information retrieval or speech recognition systems for them to be applied to practical applications.

7.1 Contributions

This work demonstrated that the standard model for information retrieval from documents transcribed automatically from speech recognition by using only the best scoring hypotheses is suboptimal because of the uncertainty in the recognition process. This led to the development of a new term weighting scheme based on the mutual information of the words in the hypotheses, which reduces to the popular inverse document frequency under standard assumptions. This new weighting scheme is able to incorporate probabilistic measures of term presence rather than Boolean values.

The speech recognition process was modeled as a probabilistic machine that affects the presence and count of terms in the reference in a stochastic manner, with the parameters of these statistics presumed to be observable in the output of the speech recognition data structures. The subsequent relevance formula that use these random variables instead of traditionally fixed values produces a relevance estimate with a distribution instead of a fixed value. It was shown that in the absence of a more sophisticated evaluation scheme, the expected value of this distribution is an adequate choice for the relative ranking of documents.

A method for estimating term presence and counts from lattices by using the sum of the forward and backward probabilities leading into a node was used as an indication of the relative probability that the term at the node is correct. This probability turned out to not be a dependable measure to base these estimates upon, although the relative ranking of the probabilities were noticeably better at predicting term presence. The relative rankings of these probabilities, both over the entire utterance as well as among the competing nodes in the lattice, were used to estimate the term presence and term counts. Generally the ranking among competing nodes was more successful, leading to reduction in the degradation in overall IR quality by 24%.

A set of techniques for the estimation of term presence and counts from the hypotheses contained in N-Best lists were developed and evaluated. In addition, producing N-Best lists containing only unique hypotheses after text processing yielded a much larger number of hypotheses from which to estimate the term presence and term count. A further refinement allowed the use of segmented N-Best lists in order to a achieve very large set of hypotheses without incurring extra computation. It was demonstrated that the larger set of hypotheses is desirable. Estimates of term presence and term count were visibly improved over those from the Top-1 hypotheses and the reduction in degradation to overall IR quality was 63% for the TREC-7 database. In a subset of the TREC-8 database containing four times as many documents, the reduction in degradation was found to be 53%, confirming the generality of these results.

Although lattices contain more potential terms for relevance computation it remains to be seen how they might be adequately separated from the incorrect hypotheses in a manner that is superior to that of the N-Best lists. While the estimates using the N-Best lists maintained the continuity of paths, in the lattice-based analysis of this chapter they were discarded and perhaps assuming that the lattice nodes could be treated independently was too optimistic. Even so, using the relative ranking of the node probabilities instead of their values turned out to be more successful than using the actual node probabilities in predicting term presence.

7.2 Suggestions for Future Work

- Using Classification and regression trees (CART) to employ more heroic confidence metrics for the estimation of the term presence and term count estimators based on the lattice. These methods have shown a great deal of promise for predicting word errors in spoken material.

- Validation of the results in this work on much larger databases (> 1000 hours of speech.) Many institutions are hard at work reducing the computational demands of speech recognition, which would assist in the research effort regarding such large data

- Comparison of performance results on more than one speech recognition engine, especially with variations in the intrinsic Word Error Rate. The effects of very high word error rate were not investigated in this work. It is possible that the behavior of the speech recognition will not be as stable, requiring a more sophisticated model of degradation.

- An attempt to expand the relevance formula to be consistent with the empirical formula used by various institutions. Although the term count and term presence primitives are found in most any of the relevance formulae, they do not conform to the weighted-sum assumptions used to justify a Gaussian distribution for the relevance.

- Development of a new evaluation criterion for retrieval systems that produce a distribution for the relevance between a document and query that account for the variance as well as the mean value.

Glossary of Recurring Symbols

n_q .... The number of queries
q .... Query number $q = 1 \dots n_q$
n_d .... The number of documents
d .... A document $d = 1 \dots n_d$
n_v .... The number of terms
v .... Term number $v = 1 \dots n_v$
$\mathrm{rel}(q, d)$ .... Human judged relevance $\{0, 1\}$
$\widehat{\mathrm{rel}}(q, d)$ .... Estimated relevance $\boldsymbol{R} \geq 0$
r_q .... Number of retrieved documents for query q
M_q .... Reference matches to query q, number of documents such that $\mathrm{rel}(q, d) = 1$
$i_{d, v}$ .... Reference document term presence, $\{0, 1\}$
$\hat{i}_{d, v}$ .... Hypothesis document term presence, $[0, 1]$
$c_{d, v}$ .... Reference document term count, $\boldsymbol{Z} \geq 0$
$\hat{c}_{d, v}$ .... Hypothesis document term count, $\boldsymbol{R} \geq 0$
$b_{q, v}$ .... Query term count, $\boldsymbol{Z} \geq 0$
l_d .... Reference document length
$\hat{l}_d$ .... Hypothesis document length
idf_v .... Inverse document frequency of term v
τ .... Term error
ρ .... Term Correlation
$I(\boldsymbol{D}; v)$ .... Mutual information between term v and the document set

Index of Special Terms

Bibliography

CSR Basic Methods

[1] X.Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, R. Rosenfeld, "The SPHINX-II speech recognition system: an overview", *Computer Speech and Language*, 1993, v. 7, pp. 137-48.

[2] B. Lowerre, *Ph.D. Thesis*, Carnegie Mellon University CS Tech Report, 1976.

[3] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE*, vol. 77, Feb. 1989.

[4] M. Ravishankar, "Some Results on Search Complexity vs. Accuracy," *Proc. 1997 DARPA Speech Recognition Workshop*, 1997.

[5] R. Schwartz and Y. Chou, "The N-best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses", *Proc. ICASSP'90*, April 1990, pp. 81-84.

CSR Keyword Spotting, Phone Recognition, and the Vocbulary Size Problem

[6] A. Asadi, R. Schwartz, J. Makhoul, "Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system", *Proc. ICASSP-91*, pp. 305-308, 1991.

[7] P. Geutner, "Using morphology towards better large-vocabulary speech recognition systems", *Proc. ICASSP '95*, 1995.

[8] D. James, S. Young, "A fast lattice-based approach to vocabulary independent wordspotting", *Proc. ICASSP '94*, 1994.

[9] M. Weintraub, "Keyword-spotting using SRI's DECIPHER large-vocabulary speech recognition system", *Proc. ICASSP '93*, vol II, 1993.

CSR Broadcast News Transcription Systems

[10] F. Kubala, T. Anastasakos, H. Jin, L. Nguyen, R. Schwartz, "Transcribing radio news", *Proc. ICSLP '96*, vol. 2, 1996.

[11] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K Seymore, M. Siegler, R. Stern, E. Thayer, “The 1996 Hub-4 Sphinx-3 system”, *Proc. Speech Recognition Workshop*, Feb. 1997.

CSR: Language Models Improvements

[12] R. Iyer and M. Ostendorf, “Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models”, *Proc. of ICSLP*, pp. 236-239, 1996.

[13] F. Jelinek, “The development of an experimental discrete dictation recognizer”, *Proc. IEEE*, vol. 73, Nov. 1985.

[14] S. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer”, *Trans. IEEE ASSP-35*, vol. 3, pp. 400-401, March 1987.

[15] K. Seymore, R. Rosenfeld, “Using Story Topics for Language Model Adaptation”, *Proc. of Eurospeech '97*, September 1997.

Confusability Studies

[16] L. R. Bahl, P. de Souza, P. S. Gopalakrishnan, D. Kanevsky, D. Nanamoo, “Constructing groups of acoustically confusable words”, *Proc. ICASSP-90, 1990*.

[17] M. Ferretti, G. Maltese, S. Scarasi, “Language model and acoustic model information in probabilistic speech recognition”, *Proc. ICASSP-89*, 1989.

[18] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, pp. 294-296, Prentice Hall, 1993.

[19] D. Roe, M. Riley, “Prediction of Word Confusability for Speech Recognition”, *Proc. IEEE ICSLP-94*, 1994.

Corrective and Discriminative Training Theory

[20] L. Bahl, P. Brown, P. de Souza, R. Mercer, “A new algorithm for the estimation of hidden Markov model parameters”, *Proc. ICASSP ’88*, vol. 1, 1988.

[21] W. Chou, C.-H. Lee, B.-H. Juang, “Segmental GPD training of a hidden Markov model based speech recognizer”, *Proc. ICASSP ’92*, 1992.

[22] B.-H. Juang, S. Katagiri, “Discriminative learning for minimum error classification”, *IEEE Trans. Signal Processing*, vol. 40, Dec. 1992.

[23] B.-H. Juang, W. Chou, C.-H. Lee, “Minimum Classification Error Rate Methods for Speech Recognition”, *IEEE Trans. on Speech and Audio Processing*, v. 5, number 3, May 1997.

Corrective and Discriminative Training from N-best lists

[24] Y.-L. Chow, “Maximum mutual information estimation of HMM parameters using the N-best algorithm for continuous speech recognition”, *Proc. ICASSP ’90*, April 1990.

[25] X. Huang, M. Belin, F. Alleva, M. Hwang, “Unified Stochastic Engine (USE) for speech recognition”, *Proc. ICASSP ’93*, 1993.

[26] R. Schwartz, S. Austin, F. Kubala, J. Makouhl, L. Nguyen, P. Placeway, G. Zavaliagkos, “New uses for the N-best sentence hypotheses within the byblos speech recognition system”, *Proc. ICASSP ’92*, 1992.

Confidence Estimation

[27] L. Chase, *Ph.D. thesis*, Carnegie Mellon University Robotics Tech Report, 1997.

[28] S. Cox, R. Rose, “Confidence Measures for the Switchboard Database,” *Proc. ICASSP-96*, 1996.

[29] L. Gillick, Y. Ito, “Confidence Estimation and Evaluation”, *Proc. 1996 LCVSR Hub-5 Workshop*, 1996.

[30] L. Gillick, Y. Ito, J. Young, “A Probabilistic Approach to Confidence Estimation and Evaluation,” *Proc. ICASSP-97*, 1997.

[31] P. Jeanrenaud, M. Siu, H. Gish, “Large Vocabulary Word Scoring as a Basis for Transcription Generation”, *Proc. Eurospeech ’95*, 1995.

Pure IR Studies

[32] W. Cooper, A. Chen, F. Gey, “Experiments in the probabilistic retrieval of full text documents,” *Proc. TREC-3*, 1995.

[33] D. Harman, “Overview of the Second Text REtrieval Conference (TREC-2),” *Proc. TREC-2*, 1993.

[34] A. Martin, J. Fiscus, M. Przybocki, B. Fisher, “1998 Hub-5 Workshop: Information Extraction,” *MITAGS*, Maryland. 1998.

[35] M. Porter, “An algorithm for suffix stripping”, *Program*, 14(3):130-137, July 1980.

[36] C. Rijsbergen, *Information Retrieval*, Butterworth, London, UK, 1975.

[37] S. Robertson, S. Walker, M. Beailieu, M.Gatford, A. Payne, “Okapi at TREC-4”, *Proc. TREC-4*, 1995.

[38] G. Salton, editor, *The SMART retrieval system-experiments in automatic document processing*, Prentice-Hall, NJ, 1971.

[39] A. Singhal, C. Buckley, M. Mitra, “Pivoted document length normalization”, *Proc. ACM SIGIR ’96*, 1996.

[40] A. Singhal, J. Choi, D. Hindle, D. Lewis, “AT&T at TREC-7,” *Proc. TREC-7*, 1998.

[41] R. Wilkinson, “Chinese Document Retrieval at TREC-6,” *Proc. TREC-6*, 1997.

[42] H. Zipf, *Human behavior and the principle of least effort*, Addison-Wesley, Cambridge, Mass., 1949.

CSR and IR integration

[43] J. Foote, G. Jones, K. Sparck Jones, S. Young, “Talker-independent keyword spotting for information retrieval”, *Proc. Eurospeech ’95*, 1995.

[44] J. Garofolo, E. Voorhees, C. Auzanne, V. Stanford, B. Lund, “1998 TREC-7 Spoken Document Retrieval Track Overview Results,” *Proc. TREC-7*, 1998.

[45] J. Garofolo, C. Auzanne, E. Voorhees,, W. Fisher, “1999 TREC-8 Spoken Document Retrieval Track Overview Results,” *Proc. TREC-8*, 1999.

[46] J. Garofolo, E. Voorhees, C. Auzanne, V. Stanford, “Spoken Document Retrieval: 1998 Evaluations and Investigation of New Metrics,” *Proc. of the ESCA ETRW workshop*, Cambridge, UK, 1999.

[47] P. Gelin and C. Wellekens, “Keyword spotting enhancement for video soundtrack indexing”, *Proc. ICSLP ’96*, 1996, v. 2.

[48] P. Gelin, C. Wellekens, “Keyword spotting for video indexing”, *Proc. ICASSP ’96*, 1996.

[49] V. Goel, W. Byrne, “Task Dependent Loss Functions in Speech Recognition: Application to Named Entity Extraction,” *Proc. ESCA Tutorial and Research Workshop for Accessing Information in Spoken Audio*, Cambridge, UK, 1999.

[50] A. Hauptmann, H. Wactlar, “Indexing and Search of Multimodal Information”, *Proc. ICASSP ’97*, 1997.

[51] S. Johnson, P. Jourlin, G. Moore, K. Sparck Jones, P. Woodland, “The Cambridge University Spoken Document Retrieval System,” *Proc. ICASSP ‘99*, Phoenix, Arizona, 1999.

[52] K. Ng, V. Zue, “Phonetic Recognition for Spoken Document Retrieval,” *Proc. ICASSP ‘98*, 1998.

[53] M. Siegler, A. Berger, M. Witbrock, A. Hauptmann, “Experiments in Spoken Document Retrieval at CMU,” *Proc. TREC-7*, 1998.

[54] M. Siegler, M. Witbrock, S. Slattery, K. Seymore, R. Jones, and A. Hauptmann, “Experiments in Spoken Document Retrieval at CMU”, *Proc. TREC-6*, 1997.

[55] M. Siegler, M. Witbrock, “Improving the Suitability of Imperfect Transcriptions for Information Retrieval from Spoken Documents,” *Proc. ICASSP ‘99*, Phoenix, Arizona, 1999.

[56] E. Voorhees and J. Garofolo, “1997 Text retrieval conference spoken document retrieval track”, *Proc. TREC-6*, 1997.

[57] S.J. Young, *et. al.*, “Acoustic indexing for multimedia retrieval and browsing,” *Proc. ICASSP ‘97*, 1997.

Informedia-like Systems

[58] M. Brown, J. Foote, G. Jones, K. Sparck Jones, S. Young, "Automatic content-based retrieval of broadcast news", *Proc. ACM Multimedia '95 Conference*, 1995, pp. 35-43.

[59] M. Christel, S. Stevens, H. Wactlar, "Informedia digital video library" *Proc. ACM Multimedia '94 Conference*, 1994, pp. 480-481.

[60] G. Jones, M. Brown, J. Foote, K. Sparck Jones, S. Young, "The Video Mail Retrieval Project: Experiences in Retrieving Spoken Documents", *Intelligent Multimedia Information Retrieval*, AAAI Press / The MIT Press, 1997.

General Theory

[61] A. Dempster, N. Laird, D. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm", *Journal of the Royal Statistical Society,* 1977, vol. 39, no. 1, pp. 1-38.

[62] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[63] A. Leon-Garcia, *Probability and Random Processes*, 1989.

[64] L. Rabiner, J. Huang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

Data Resources

[65] *Houghton Mifflin Primary Dictionary*, 1986.

[66] Linguistic Data Consortium, http://www.ldc.upenn.edu/.

[67] M. Siegler, U. Jain, B. Raj, R. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News Audio," *Proc. DARPA Speech Recognition Workshop*, 1997.

[68] National Institute of Standards and Technology, Software available publicly at http://www.nist.gov/speech/software.htm

[69] L. Wall, T. Christiansen, R. Schwartz, *Programming Perl*, 2nd edition, O'Reilly, 1996.