Using Word Position in Documents for Topic Characterization

Reem K. Al-Halimi
School of Computer Science
University of Waterloo
Waterloo, Canada
ralhalim@uwaterloo.ca

Frank W. Tompa
School of Computer Science
University of Waterloo
Waterloo, Canada
fwtompa@uwaterloo.ca

Technical Report cs-2003-36

Abstract

In this report we show how to use the position of words in a text for characterizing the topic of the text, and compare our method to measures that use frequency statistics that are independent of word position. We show that word position information produces words that are more suited for characterizing topics and at the same time relies on a vocabulary size that is as little as 10% of the size used by the other measures.

1 Introduction

It is generally accepted within the text retrieval community that the words used in documents relate to the documents' content, and that documents with similar content will tend to use a similar vocabulary. Therefore, text retrieval systems index and compare documents by the words they contain. The quality of these index words is usually measured by how well the words can discriminate a topic or document from all other topics or documents [23, 19, 8]. In these systems a good index word can be any sequence of characters that occurs often in one type of documents and rarely in the other types, even if it is weakly relevant to content. For example, the *Earnings and Earnings Forecasts* category in the Reuter's database [18] contains the term &*lt* in place of the left bracket at the end of company names. Therefore, &*lt* is a good discriminator of that category against other categories in the database [16]. However, even if &*lt* may be a good discriminator for the *Earnings and Earnings Forecasts* category, it is not relevant to the meaning of the category. Although such discriminators are appropriate as category markers, they lack essential content relevance information that is needed for tasks such as document visualization and text summarization.

Several researchers have attempted to evaluate index words by their content. Damerau, for example, uses word frequency to extract *domain-oriented vocabulary* which he defines as

> a list of content words (not necessarily complete) that would characteristically be used in talking about a particular subject, say *education*, as opposed to the list of words used to talk about, say *aviation*. [8]

However, his evaluation method does not guarantee that the words are *domain-oriented*. It only shows that the selected words are used more often in the domain's documents. Therefore, good discriminators can easily be accepted as topic words by this test if they happened to be used more often in one domain than in the other domains tested. But a good discriminating word is not necessarily a content word.

For better content words some researchers have turned to the distribution of topics within a text. Documents in this case are no longer viewed as *bags-of-words* where the order of their constituent words is ignored. Rather, text is viewed as a collection of topics, and words and their position in the text are used to reflect these topics. The longer the discussion of a topic, the larger the segment that contains its words and the more important the topic with respect to the document. Minor topics, on the other hand, are discussed in smaller segments of text. Bookstein, Shmuel, and Raita [3] use this behavior of content-bearing words to extract good index words that are likely to be useful in satisfying users' requests. Their experiments show that such information on word occurrence improves the quality of words selected for indexing when compared to pure inverse document frequency.

Through this method Bookstein *et al.*identify content words, but they do not show how to use the method to characterize the topic of a whole document or which words are indicative of that topic.

Katz [14] notes that although content words are likely to repeat in close proximity to each other, those that are treated heavily and continuously in the text will occur across the length of the text. We speculate that these intensely treated content words are strongly related to the main topic of the document and are thus good topic words.

In the remainder of this paper, we show how to use a word's pattern of occurrence in a topic's documents in measuring the word's relevance to the topic. We also show that these words, called *topic words*, correspond more closely to manually selected keywords than words chosen using traditional indexing techniques. This supports our claim that topic words are better identifiers of the topical content of documents.

2 Topic Relevance

A word w is strongly relevant to topic t if w reflects a concept that can be discussed as a subtopic of t. For example, *belief networks* and *agents* are strongly relevant to the topic *Artificial Intelligence*, and *corpora* and *discourse analysis* are strongly relevant to the topic *Computational Linguistics*, but *the* and *conductor* are less relevant to either topic. Figure 1 shows a list of concepts

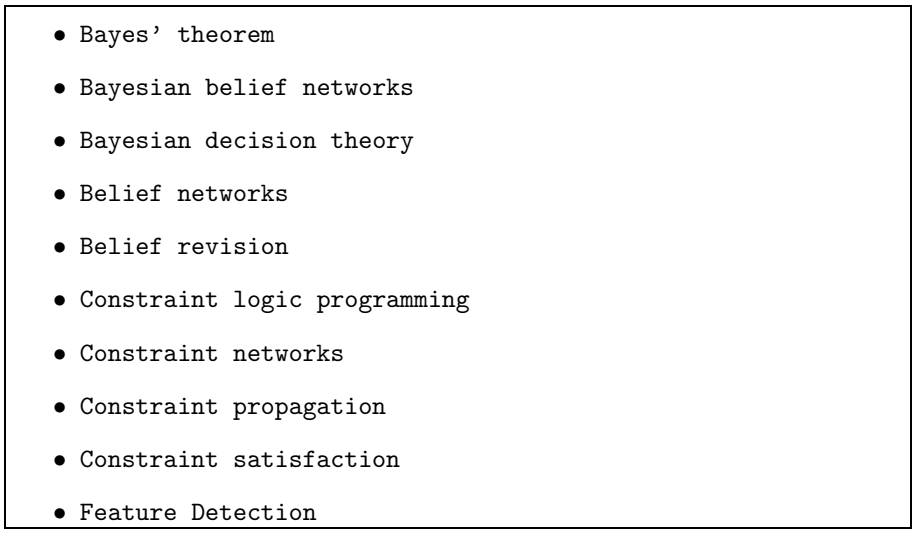


Figure 1: Concepts that are topically relevant to *Artificial Intelligence*

that are strongly relevant to *Artificial Intelligence*, taken from the index of the Encyclopedia of Artificial Intelligence [21].

Document keywords may also be viewed as topic words. Keywords indicate the main topics of the document so the set of keywords used to describe a topic's documents acts as a partial set of topic words for the topic discussed in the document. These keywords can either be found in a preset keyword field in the document, or they can be recognized through some visual features throughout the document. InfoFinder [15], for example, extracts keywords (called topic phrases) from documents using a set of heuristics based on 'visually significant features' such as italics and document structure. The keywords of a topic's documents are used to build decision trees that reflect the topic's content.

As we mentioned earlier, topically relevant words are Damerau's *domain-oriented vocabulary* [8]. Damerau tests two different approaches that extract domain-oriented vocabulary from the body of plain text documents. In the first he sorts the list of all words in the domain documents by their frequency, eliminates words bearing little content such as *the* and *it* (called stopwords) from the list, and finally trims the list size to a predetermined constant by removing the lowest frequency words. In the second approach he creates two domain lists: for the first list he extracts from a dictionary all words whose label is that of the domain, and for the second list he indexes the words used in the domain documents. He then creates the final domain list from words in common between these two lists. Given a previously unseen set of documents for a domain, Damerau found that more of the domain's list appeared in the domain's documents than words from most other domains' lists. However, this acceptance test does not guarantee that the words are *domain-oriented*. It only shows that the selected words are used more often in the domain's documents.

Therefore, words like *she* and *her* can easily be accepted as topic words by this test if they happened to be used more often in one domain than in the other domains tested. In this case *she* and *her* may be good discriminators of that domain, but they are not topically relevant words. Therefore, although topic words are by definition acceptable discriminators for their topics, not all discriminators are necessarily topic words.

With this definition of relevance as a domain-oriented vocabulary, topic characterization becomes a representation of the distribution (in a non-statistical sense) of topic concepts in the topic's documents. Furthermore, the variation in relevance values reflects the variation in the strength of topic relevance to the various concepts across the text.

The ability to measure the relevance of a topic to a word implies some knowledge of the word and the topic. In information retrieval this knowledge is usually acquired by analyzing some sample documents and the topics to which they belong. Assume we have access to a representative sample of plain text documents for each topic t in the set of all possible topics T, and that each document is represented by the words it contains in the order they occur in the document. In this report we will call this set of sample documents the database of training documents, or simply the database. Our task is to recognize strongly relevant words for each topic from the representative sample. In the remainder of the report we will discuss and compare several different measures of relevance.

3 Measuring Topic Relevance

Topic words are clearly associated with their topics. Otherwise, it would be hard to argue that they are "characteristically" used in discussing the topic. To measure this association we use *Pointwise Mutual Information* (PMI). PMI is an extension of the information theoretic measure *Mutual Information* [16]. It has been used for many different retrieval tasks including feature extraction [26] and word concordance discovery [5]. PMI measures the likelihood of observing two events simultaneously as opposed to observing either event separately. This measure is defined as follows:

$$I(x,y) = log\frac{p(x,y)}{p(x)p(y)} \tag{1}$$

where:

- $p(x,y)$ is the probability of observing x and y together in the database.
- $p(x)$ ($p(y)$) is the probability of observing x (y) separately.

Within the context of topic relevance, x is the topic, y is the word whose relevance to x is of interest, and $I(x,y)$ reflects the relative likelihood of using a word y when discussing topic x as compared to using either independently, or, since $I(x,y) = I(y,x)$, the relative likelihood that an observed instance of word y refers to topic x as compared to using either independently.

In order to measure the PMI between a word and a topic, their probability of co-occurrence $p(w,t)$ should be estimated, as well as their probabilities of occurring separately $p(w)$ and $p(t)$. The probability of occurrence of a word is usually estimated using some observable statistic of that word such as the number of documents in which it occurs (document frequency), or the word's total number of instances in the database (term frequency).

In what follows we will discuss how document frequency and term frequency can be used to measure the relevance of a topic to a word, and we compare the effectiveness of these two statistics to two other word statistics.

3.1 Measuring Topic Relevance using Document Frequency

Document frequency (DF) is the number of documents in which a word occurs. Using DF our PMI measure of the relevance of word w to topic t becomes

$$\begin{aligned} M^{DF}(w,t) &= log\frac{p(w,t)}{p(w)p(t)} \\ &= log\left(\frac{\frac{DF(w,t)}{|db|}}{\frac{DF(w,db)}{|db|} * \frac{|t|}{|db|}}\right) \\ &= log\frac{DF(w,t) * |db|}{DF(w,db) * |t|} \\ &= log\frac{DF(w,t)}{DF(w,db)} + log\frac{|db|}{|t|} \end{aligned} \tag{2}$$

where:

- $DF(w,t)$ is the number of topic t documents containing the word w.
- $DF(w,db)$ is the number of database db documents containing the word w.
- $|t|$ is the number of documents in the database whose main topic is t.
- $|db|$ is the number of documents in the database.

Since the term $log\frac{|db|}{|t|}$ is independent of words for a fixed topic t and database db, $M^{DF}(w,t)$ in Equation 2 compares words under t based on the ratio $\frac{DF(w,t)}{DF(w,db)}$. Words that occur in significantly more documents of t than in the rest of the database will be weighed more heavily than those occurring in relatively fewer documents of t than the rest of the database.

Measuring topic relevance of a word by the number of documents in which it occurs assumes that words that are not frequently used in a variety of documents usually occur in a document if they are strongly relevant to a main topic of that document, and that a word that is strongly relevant to a topic t will have a higher proportion of the documents in which it appears falling under t. For

example, a word that only occurs in documents of t will have a higher weight than another word where only some of its documents belong to t.

We find the basic assumption in this measure troublesome. In particular, the measure will assign a high weight to singletons (words occurring only once in the document) such as misspellings and other infrequent and irrelevant words. For instance, mentioning the word *treaty* in this report does not mean the word is strongly relevant to the topic of the report. Yet the above measure will count this report as evidence of the relevance of the word to the main topic of the text.

One way to filter out some of these irrelevant words is to discard documents where the word is a singleton, as we shall see next.

3.2 Measuring Topic Relevance using Modified Document Frequency

As we mentioned in the previous section, the problem with using pure DF in measuring topic relevance is that it exaggerates the importance of single occurrences in a document. To overcome this drawback we suggest discarding from the DF count those documents where the word occurs only once. In this case our topic relevance measure becomes:

$$\begin{aligned} M^{\widetilde{DF}}(w,t) &= log\frac{\widetilde{DF}(w,t) * |db|}{\widetilde{DF}(w,db) * |t|} \\ &= log\frac{\widetilde{DF}(w,t)}{\widetilde{DF}(w,db)} + log\frac{|db|}{|t|} \end{aligned} \tag{3}$$

where

- $\widetilde{DF}(w,t)$ is the number of topic t documents containing more than one occurrence of the word w.
- $\widetilde{DF}(w,db)$ is the number of database db documents containing more than one occurrence of the word w.
- $|t|$ and $|db|$ are the number of documents in t and db respectively.

The new measure assumes that the probability of two occurrences of an irrelevant word is quite low, and that those words occurring at least twice anywhere in the document may be strongly relevant to the main topic of the document. These assumptions are made by Katz [14], who argues that when a word is relevant to the topic of the document it occurs in a *burst* of repetitions. Church also uses $\widetilde{DF}(w,db)$ in one of his adaptation measures [4], where he shows that content words[1] tend to occur at least twice in documents to which they are

[1]Manning and Schütze define *non-content* words informally as "words that taken in isolation .. do not give much information about the contents of the document" [16]. Note that although a content word is usually relevant to the topic being discussed in the document, it does not need to be.

strongly relevant.

Based on the above assumptions, if the proportion of topic t documents containing more than one occurrence of a word is high, then the measure assumes the word is strongly relevant to t.

Replacing document frequency with $\widetilde{DF}$ alleviates the singleton-word problem. But $\widetilde{DF}$, as well as DF, assumes an independence of the number of times a word is used within the document from the degree of relevance of the word to the document's topic. For both statistics a word repetition of 10 times in one document is as good as repeating the same word twice only in that same document. Yet some researchers assert that this is not valid. Katz for example states that

> the total number of observed occurrences of the content word or phrase in the document ought to be a function only of the degree of relatedness of the concept named by the word to the document or, in other words, of the intensity with which the concept is treated. [14]

Luhn also hypothesized that the frequency of a word in a document is a strong indicator of the word's significance in the text [23].

3.3 Measuring Topic Relevance using Term Frequency

Many researchers in the information retrieval community use term frequency TF as an indicator of the importance of a word to a document. We can carry this concept of frequency as an indicator of importance towards measuring topic relevance by plugging TF into Equation 1 above:

$$\begin{aligned} M^{TF}(w,t) &= log\frac{TF(w,t) * ||db||}{TF(w,db) * ||t||} \\ &= log\frac{TF(w,t)}{TF(w,db)} + log\frac{||db||}{||t||} \end{aligned} \tag{4}$$

where

- $TF(w,t)$ is the number of occurrences of w in topic t documents.
- $TF(w,db)$ is the number of occurrences of w in the database db.
- $||t||$ and $||db||$ are the number of terms in t and db respectively.

Equation 4 assumes that strongly relevant words are those that occur more frequently in topic t than in the whole database. Damerau used essentially this measure in determining domain-oriented vocabulary [8] as discussed above, and he uses a function similar to Equation 4 to extract 2-word domain phrases for a set of pre-specified domains, based on the ratio of the frequency of the phrase within the domain to its frequency in the whole database [9]. The assumption here is that good domain phrases will tend to occur more often on average in the domain's documents than in the whole database.

3.4 Incorporating Relative Word Positions in Measuring Topic Relevance

Whether using term frequency or document frequency, the measures proposed so far assume a bag of words document view, where the total number of occurrences in the topic documents and in the database is sufficient to describe the contents of the topic regardless of where the words occur. The bag of words view is incomplete for our notion of text as an interweaved collection of topics. The position of words in a document can be useful in understanding the document's content.

We view text as a collection of topics, with the main topic(s) spanning the length of the text. Since words are the atomic units describing document content, we expect topic words to span across the text as well. Some of these topic words will occur in small segments of the text, while others will repeat throughout the text. Bookstein *et al.* [3] use this behavior of content-bearing words to extract good index words that are likely to be useful in satisfying users' requests. The authors apply goodness measures of such words that compare the word's occurrence behavior in the text to the expected random occurrence. They define two occurrence behaviors: the word's tendency to clump by repeating in close proximity in a single textual unit such as a paragraph, and the word's tendency to occur at least once in several consecutive textual units in a document. Their experiments show that such information on word occurrence improves the quality of words selected for indexing when compared to pure inverse document frequency. The experiments also indicate a link between content-bearing words and these words' tendency to clump.

Although their method identifies content words, the authors do not show how to use the method to characterize the topic of a whole document or which words are indicative of that topic.

Katz [14] notes that although content words are likely to repeat in close proximity to each other, those that are treated heavily and continuously in the text will occur across the length of the text. We speculate that these intensely treated content words are strongly related to the main topic of the document and are thus good topic words. The challenge here is to identify words that repeat across the text, and evaluate them based on their tendency to span the length of text in a topic's documents.

To identify words that repeat across the text we turn to a second measure introduced by Church [4]. Church uses the notion of word spread to show that content words *adapt*, i.e. their probability of occurrence changes based on the lexical content of the document. Church divides each document into two halves: the first half is called the *history*, and the second is called the *test*. He shows that when a content word appears in the history, its probability of occurring in the test segment rises significantly.

If a word spans the length of the text, then it will occur in both halves of the document. Adopting Church's idea of segmenting a document into two halves, our topic relevance measure becomes

$$
\begin{aligned}
M^{DF_2}(w,t) &= log\frac{p(w,t)}{p(w)p(t)} \\
&= log\left(\frac{\frac{DF_2(w,t)}{|db|}}{\frac{DF_2(w,db)}{|db|} * \frac{|t|}{|db|}}\right) \\
&= log\frac{DF_2(w,t) * |db|}{DF_2(w,db) * |t|} \\
&= log\frac{DF_2(w,t)}{DF_2(w,db)} + log\frac{|db|}{|t|}
\end{aligned} \tag{5}
$$

where:

- $DF_2(w,t)$ is the number of topic t documents containing the word w *in both halves of the document.*
- $DF_2(w,db)$ is the number of database db documents containing the word w *in both halves of the document.*
- $|t|$ is the number of documents in the database whose main topic is t.
- $|db|$ is the number of documents in the database.

Under this measure, topically relevant words are those that occur in both halves of the document in topic t significantly more often than in both halves of the database documents.

The idea of segmenting a document into two halves can be easily generalized into any number n of segments. In this case words are said to spread across the document if they occur in all n segments. Our measure of topic relevance becomes

$$
M^{DF_n}(w,t) = log\frac{DF_n(w,t)}{DF_n(w,db)} + log\frac{|db|}{|t|} \tag{6}
$$

where:

- $DF_n(w,t)$ is the number of topic t documents containing the word w *in all n segments of the document.*
- $DF_n(w,db)$ is the number of database db documents containing the word w *in all n segments of the document.*
- $|t|$ is the number of documents in the database whose main topic is t.
- $|db|$ is the number of documents in the database.

DF_n restricts the frequency count to the number of documents where the word occurs in all n segments. Words that never occur in all n segments of any document in the database have a frequency $DF_n(w,db)$ of 0. Such words

are assumed to be not topically relevant to any topic t in the database and are therefore excluded from the $M^{DF_n}(w,t)$ vocabulary for all topics t. The immediate effect of using the DF_n frequency count is a vocabulary size smaller than that used by M^{DF}, M^{TF}, and $M^{\tilde{DF}}$. The higher the value of n, the more words are excluded, and the larger the reduction in the size of the vocabulary. But this reduction is acceptable only if it does not harm the effectiveness of the M^{DF_n} measure in recognizing topic words. In what follows we will look at the effectiveness of M^{DF_n} in extracting topic words for several different values of n and compare that to the other three topic relevance measures.

4 Evaluation

In Section 3 we proposed four different functions to measure the relevance of a word to a topic. All measures use the PMI formula from Equation 1, with different definitions of $p(w,t)$, $p(w)$, and $p(t)$ for a given word w and topic t. Up until now we have been deliberately ignoring the question of what constitutes a topic. This is because the discussion has been general enough to apply to any topic one may think of, be it *politics* or *"today's lunch."* However in order to compare the topic relevance measures experimentally, we must simplify the concept of *topic* so that it can be easily captured and quantified. For evaluation purposes we define a topic as a predefined subject or category, such as the classes in Yahoo!'s classification hierarchy [24], the subject classes of the ACM Computing Classification System [1], or the classification used in the Reuter's database [18].

For this evaluation we use the CoRR database [6] and the classification system used by that database. For our purposes, CoRR has the advantage of longer documents (all having several pages) which are not found in other, more widely used databases such the Reuter's database (most having a couple of paragraphs only). Documents longer than a few paragraphs are essential for testing the effectiveness of the M^{DF_n} measure which is based on segmenting each document into many small sections.

Given a predetermined set of categories, and a database of manually classified documents, we can now generate topic words and evaluate the four measures.

4.1 The CoRR Database

The CoRR database [6] is an online repository for research in computer science. The database consists of theses, technical reports, conference papers, and journal papers from the last decade. Documents range in length between 5 pages and around 250 pages, but are on average about 13 pages long. They are mainly in LaTeX format, with a few pdf, and some ps and html files.

Each document in the CoRR database has been classified by the paper's authors under one or more of the pre-determined 34 categories listed in the Appendix.

Category	Description	size
CL	Computation and Language	185 documents
LO	Logic in Computer Science	90 documents
AI	Artificial Intelligence	82 documents
CC	Computational Complexity	67 documents
CG	Computational Geometry	58 documents
DS	Data Structures and Algorithms	43 documents
PL	Programming Languages	36 documents
SE	Software Engineering	30 documents
LG	Learning	30 documents
DC	Distributed, Parallel, and Cluster Computing	28 documents
CE	Computational Science, Engineering, and Finance	19 documents
NI	Networking and Internet Architecture	16 documents

Table 1: The CoRR database categories used in this evaluation

Our version of the database consists of documents submitted between January 1998 and June 2001 for a total of 1151 documents, mostly in LaTeX format. The LaTeX documents were converted to text using a version of detex [10] that was modified to ignore text preceding the $\backslash begin\{document\}$ command, as well as ignoring abstracts and footnotes. A few of the pdf files were converted using Adobe's pdf2txt. In total, 824 text files were converted successfully. Many of the 34 categories contain very few documents. Documents that belong exclusively to one or more of these small categories were removed from the database leaving 736 documents. The remaining categories and their sizes in number of documents are shown in Table 1. Some of these categories are still quite small, but they are useful in understanding the effect of category size on the quality of the extracted vocabulary.

4.2 Preprocessing

The total vocabulary in our version of the CoRR database is $53,877$ unique words. Vocabulary words are case-insensitive sequences of alphanumeric characters consisting of at least one letter. Abbreviations containing a dot interleaved with the alphanumerics are accepted, as well as words containing an underscore, as in the sequence w_t.

Two pre-processing steps are widely used when indexing databases: stopword removal and word stemming. Stopword removal ignores words that are generally accepted as content-free such as *the* and *it*, and stemming removes morphological inflections from words.

Many researchers opt to remove stopwords using a preset stopword list for efficiency and effectiveness [20] [25] [13]. However, these stopword lists vary from one database to another and from one language to another, so it is desirable

that topic relevance measures be able to identify these non-content words as such regardless of the language or database used. Stopword removal also raises a concern when using the DF_n frequency counts: removing words from the body of a document will disrupt the position of the remaining words in the document thus disrupting the DF_n frequency counts. The approach we choose to deal with this change in DF_n frequencies depends on our view of the role of word position: if the importance of a word position is affected by the word's content then stopword positions are of little importance and can be safely ignored. If, on the other hand, word position in a document is a rough reflection of the word's sentence position then all word positions are important, including those of stopwords, and words that occur in some segment of the document before stopword removal should still occur in this same segment after stopword removal. In this case, before removing stopwords their positions should be held by some null word that can safely be discarded after all DF_n frequencies are counted.

Stemming is another common preprocessing step. It reduces words with varying morphological inflections to one common representation. Thus, *runs* and *running* will both be represented by the stem *run*. This process has been reported to help compress an index by up to 50% [11]. Stemming has also been used to increase the number of documents found by search systems although there is no strong evidence of its effectiveness in discovering documents that satisfy the users' request [11].

There are many different approaches to stemming from simple table look-up and predefined stemming rules to methods that attempt to discover stems statistically with no prior knowledge of any stemming rules. None of these approaches has been shown to outperform the others in terms of precision or retrieval effectiveness, but they differ in their compactness and simplicity. One of the most compact is the Porter stemmer [11] [2], which gradually reduces a word to its simplest form by iteratively looking for the longest suffix it can remove based on a set of predefined rules [17].

Thus we have four possibilities for representing documents' words: with or without stopwords and with or without stemming. To explore the effects of stopword removal and stemming on the precision of our topic relevance measures, we created four independent indices for the CoRR database. The first index (SIMPLE) is the simplest one with no stopwords removed and no stemming performed. The second index (STEM) does not remove stopwords, but replaces words by their stems using the Porter stemmer prior to frequency count. The third index (STOP) removes stopwords found on the SMART stopword list [22], but does not stem the remaining words. While the fourth version of the CoRR index (S&S) removes stopwords as was done for the STOP index and stems the remaining words as was done in the STEM index. We have opted to replace stopwords in the STOP and S&S indices by a null word in order to preserve the words' original positions in the database.

Because the resulting topic word lists for two indices consist of stemmed words while the STOP and SIMPLE indices may include several different morphological variants of the same stem, we subsequently convert the unstemmed topic word lists of STOP and SIMPLE into stemmed lists of topic words as fol-

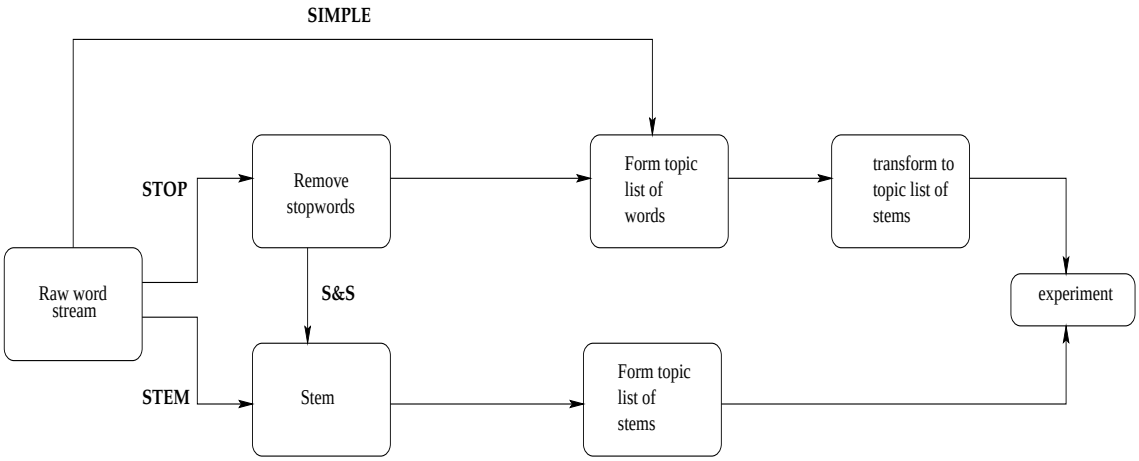


Figure 2: The steps involved in creating each of the indices used in the evaluation experiments

lows: Each word in the list is stemmed using the Porter stemmer and assigned the unstemmed word's weight. When several variants of the same stem occur in the list, only the first occurrence of the stem is preserved thus giving it the highest relevance weight of all its variants. This removes from the unstemmed lists variations of the same word and produces homogeneous topic word lists that can be cross-compared easily. The creation process of the different indices is shown in Figure 2.

It is important to note that stemming after topic word list generation is different from stemming prior to index creation. When words are stemmed before they are indexed, their frequency counts will include the occurrence of all morphological variants of the word in the database. But when stemming is done after the initial topic word lists are created, a stem will be assigned the highest topic relevance weight of its variants where *each variant is weighed separately.*

Now that we have our final topic word lists, we turn to the method used to evaluate the relevance measures.

4.3 Creating the Baseline

At the beginning of this report we define topic words for a category. The definition highlights the criterion by which to judge our topic relevance measures: a word that is identified by a relevance measure is acceptable as a topic word if the word is used by humans in discussing that topic. According to this criterion we must evaluate each topic relevance measure by comparing it against a manually selected list of topic words for each category. The most obvious source for these words are the keyword fields in the CoRR papers. These fields are usually assigned by the authors to briefly describe the main contents of the

paper. Thus, for a paper that is classified under *Artificial Intelligence* for instance, its keywords will describe issues discussed under *Artificial Intelligence*. The collection of all keywords used in all the *Artificial Intelligence* papers could then form the list of manually selected topic words for the *Artificial Intelligence* category. Unfortunately, very few papers in the CoRR database contain the keyword field.

Luckily, since our goal is to compare the effectiveness of the topic relevance measures in capturing what humans consider topic words, any database discussing the category is a potential source for the manual topic word list. Furthermore, seeking the target topic word list from a source other than our CoRR database has the advantage that the words are more likely to be database independent. Thus the success of a topic relevance measure in finding words from this ideal list is an indication of the measure's success in recognizing vital topic words independently of the database. Also, since the ideal list is extracted from an external source, the success of our topic relevance measures in finding some of these words shows that topic words are universal for a topic and that given a sufficient amount of data about a topic, the topic words found by these measures are in fact good representations of the topic.

One source of an ideal list is the index of the Encyclopedia of Artificial Intelligence [21] which provides a comprehensive list of artificial intelligence concepts. Some online journal indices also provide topic words for many different categories. These indices are easily searchable, and such large databases reduce the chances of missing good topic words for any category in the index. The most accessible index we found for the range of topics in the CoRR database was the Computer and Information Systems Abstracts index from the Cambridge Scientific Abstracts (*CSA database*) [7]. The online index is a database of 329,660 records from 850 serials. Each record contains several fields describing a serial's article. The fields include, among others, the *title* of the article, the *author*, the *classification* field which contains the list of general classes under which the article falls, the *descriptors* field which contains a list of closed vocabulary items describing the category of the article, and the list of *identifiers* which are open vocabulary terms found by human indexers to be strongly relevant to the contents of the article. Figure 3 shows a sample CSA record.

We assume that identifiers that are believed to be relevant to an article are also relevant to the categories to which the article belongs. The identifiers are used in discussing those categories and are therefore the equivalent to what we have been calling topic words. Thus, given a large enough list of identifiers for each category, we expect good topic words to appear on this list. To create such a list for a category C we collect a large number of CSA records that are classified under category C and extract the identifiers used to describe the papers indexed in these records. We will call this *the identifier list* for C. The more of these identifiers a topic relevance measure can discover, the better the measure.

But identifier lists contain phrases as well as individual words. Since we deal with single words only, each identifier phrase is broken into its constituent words. Stopwords are then removed from the resulting identifier list and the

TI: Title On the hardness of approximate reasoning

AU: Author Roth, Dan

AF: Author Affiliation Harvard Univ, Cambridge, MA, USA

SO: Source Artificial Intelligence [ARTIF INTELL], vol. 82, no. 1-2, pp. 273-302, 1996

IS: ISSN 0004-3702

PB: Publisher ELSEVIER SCIENCE B.V., AMSTERDAM, (NETHERLANDS)

AB: Abstract Many Al problems, when formalized, reduce to evaluating the probability that a propositional expression is true. In this paper we show that this problem is computationally intractable even in surprisingly restricted cases and even if we settle for an approximation to this probability. We consider various methods used in approximate reasoning such as computing degree of belief and Bayesian belief networks, as well as reasoning techniques such as constraint satisfaction and knowledge compilation, that use approximation to avoid computational difficulties, and reduce them to model-counting problems over a propositional domain. We prove that counting satisfying assignments of propositional languages is intractable even for Horn and monotone formulae, and even when the size of clauses and number of occurrences of the variables are extremely limited. This should be contrasted with the case of deductive reasoning, where Horn theories and theories with binary clauses are distinguished by the existence of linear time satisfiability algorithms. What is even more surprising is that, as we show, even approximating the number of satisfying assignments (i.e., 'approximating' approximate reasoning), is intractable for most of these restricted theories. We also identify some restricted classes of propositional formulae for which efficient algorithms for counting satisfying assignments can be given.

LA: Language English

PY: Publication Year 1996

PT: Publication Type Journal Article

DE: Descriptors Approximation theory; Probability; Computational methods; Constraint theory; Algorithms; Problem solving

ID: Identifiers Approximate reasoning; Bayesian belief networks; Model counting problems; Propositional languages; Horn theory

CL: Classification C 723.4 Artificial Intelligence; C 921.6 Numerical Methods; C 922.1 Probability Theory; C 721.1 Computer Theory (Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory)

SF: Subfile Computer and Information Systems Abstracts

AN: Accession Number 0225588

Figure 3: A Sample Record From the CSA's Computer and Information Systems Abstracts database

CoRR Category	CSA Equivalent	CSA Field	#CSA Records
AI	Artificial Intelligence	classifiers	21,910
PL	Programming Languages	classifiers	8,940
LO	formal languages and grammars; mathematical logic; formal logic; formal languages; theorem proving; computational logic; lambda calculus; set theory; temporal logic; verification; correctness proofs; logic programming;	classifiers descriptors	7,124
CC	Computational Complexity	descriptors	9,016
CG	Computational Geometry	descriptors	2,257
SE	Software Engineering	descriptors	4,428
CL	Natural Language Processing	descriptors	810
DS	Data Structures; Algorithms	descriptors	45,053
NI	Computer Networks; Communication Networks; Network Protocols; Wide Area Networks; Packet Networks	descriptors	12,218

Table 2: CoRR Category to CSA category mapping and the number of CSA records under each mapped CoRR category

remaining words are stemmed (using the Porter stemmer) and sorted by the number of times the stem is used (term frequency).

In order to use the CSA database for evaluating our topic relevance measures, we still have to attend to one more issue: the classification hierarchy defined by the CSA database is quite different from the categories defined by the CoRR database, but most of the twelve categories in the CoRR database can be mapped to ones in the CSA database. One CoRR category cannot be mapped to an acceptable equivalent, while two had too few documents for a reliable evaluation. The mappings for the remaining nine out of the twelve CoRR category areas are shown in Table 2. Some of the CoRR categories are equivalent to CSA classes used in the classification field, such as *Artificial Intelligence* which maps the *C 723.4 Artificial Intelligence* class, and *Programming Languages* which maps the *C 723.1.1 Computer Programming Languages* class; while other more specific classes can only be mapped to descriptors: *Computational Linguistics* for example is mapped to the descriptor *Natural Language Processing*. Two categories cover more than one class or descriptor in the CSA database. For example *Networking and Internet Architecture* covers many different descriptors in the CSA database each pertaining to some aspect of the category such as *Wide Area Networks*, *Network Protocols*, and *Computer Networks*.

In summary, to create the ideal topic list for each of the nine CoRR categories

CoRR Category	# CSA Records	Target List Size
AI	2,357	2,400
PL	2,159	2,081
LO	2,000	2,574
CC	2,159	2,463
CG	2,210	2,576
SE	2,381	2,301
CL	783	1,153
DS	2,500	2,583
NI	2,151	2,106

Table 3: The number of CSA records collected to create the target list for each of the nine CoRR categories, and the number of words in the resulting target list.

shown in Figure 2 we collected a total of 18,700 CSA records, each satisfying the CSA equivalent of at least one CoRR category. Table 3 shows the exact number of records collected for each of the nine CoRR categories. We then extracted the identifiers used in the identifier fields of all CSA records in each category. Identifier phrases were broken into single words, words found on the SMART stopword list [22] were removed, and the remaining words were stemmed using the Porter stemmer [17]. The number of words in each category's identifier list is shown in Table 3.

Once the identifier lists were created, each list was sorted by the number of occurrences of the identifier stem. This sorting de-emphasizes words that are rarely used to discuss the category or which may have been used to discuss additional categories to which the indexed paper belongs. Table 4 shows the ten most frequent stems in *AI*'s identifier list. The final identifier list is pruned down to the top n words to form the target topic word list. We experimented with target lists of size $n = 100$, 200, 300, 400, 500, 1000, and 2000 words.

4.4 Evaluation Method

The target lists provide us with the basis for comparing the effectiveness of each of our topic relevance measures. Each of these lists contains a sample set from the space of words used to discuss the list's corresponding category. The measure that consistently finds more of these stems, especially when more of the stems are found near the start of the topic word list, is deemed to be more effective in extracting topic words than the other measures. In the remainder of this section *target list* refers to the list of identifiers (i.e. stems) presumed to represent a topic, and *topic word list* and *word list* refers to the list of words that has been extracted by a topic relevance measure for a category. Each word list is sorted in decreasing order by topic relevance value.

The aim in these experiments is to evaluate the relative effectiveness of

Identifier Word Stem	Example Identifier Phrase
neural	Neural Networks
network	Neural Networks
learn	Cooperative Learning Method
system	Multi-agent Systems
model	Recurrent Fuzzy Neural Model
algorithm	Learning Algorithms
function	Horn Function
base	Knowledge Based Systems
method	Cooperative Learning Method
fuzzi	Fuzzy Rules

Figure 4: Ten most frequent word stems in the AI identifier list along with example phrases containing these identifier words

our topic relevance measures in extracting words humans view as topic words. We would also like to see the effect of stemming and stopword removal on a measure's word list.

To compare the effectiveness of the topic relevance measures we follow standard procedure in information retrieval experimentation and use the average interpolated 11-point precision-recall curves [12]. Precision at any point in the sorted word list is defined as the ratio of target words found until that point in the list to the number of words seen so far, while recall at any word in the sorted list is the ratio of the number of target words found up to this point in the sorted word list to the total number of identifiers in the target list. The 11-point precision-recall curves show the precision at 11 recall points from 0 to 1 with increments of 0.1. Specifying the recall values allows us to compare the results for different topic word list sizes. Since some of these recall points may not exist in our sorted list, we use interpolated 11-point precision-recall curves (p-r curves). In the interpolated curves the precision at each one of the recall points is the highest precision found at recall greater than or equal to the current recall point. The average interpolated 11-point precision-recall curves (average p-r curves) are calculated over all 9 CoRR categories for each topic relevance measure to compare the average effectiveness of each measure.

4.5 Results

In this section we study the effectiveness of the topic relevance measures M^{TF}, M^{DF}, $M^{\tilde{DF}}$, and M^{DF_n} for n = 2,4,6,8,10,12,14,16 using the four indices SIMPLE, STEM, STOP, and S&S. The p-r curves for each of these measures were produced using target lists comprising a maximum of 100, 200, 300, 400, 500, 1000, and 2000 word stems. Recall that these lists are a sample set of topic words taken from a different database that covers topics similar to those in the CoRR database. Therefore, we do not expect any of the measures to reach high precision or recall values. However, if a measure consistently succeeds in dis-

covering more target words than the other measures, then by comparison this measure is a better topic relevance measure.

The average p-r curves for each of the seven list sizes using the SIMPLE index are shown in Figures 5– 11. Each curve in these figures is labeled by the type of frequency used in the topic relevance measure associated with the curve. For example the curve for M^{TF} is labeled as TF. All seven lists produce a similar pattern: the more traditional measures M^{TF}, M^{DF}, and $M^{\tilde{DF}}$ have a lower precision than those of M^{DF_n}. This pattern becomes more apparent as we increase the target list size. Measures that are based on a pure count of occurrences, M^{TF} and M^{DF}, have the lowest precision values. Accounting for the minimum frequency of the term in calculating its document frequency in $\tilde{DF}$ improves precision but even $M^{\tilde{DF}}$ falls behind M^{DF_n} at all recall levels. M^{DF_2} is a further improvement over $M^{\tilde{DF}}$, which indicates that *how* the term occurs in a document influences the term's relevance to the contents of the document. This is corroborated further by the improvement in precision when using M^{DF_n} for $n > 2$. The figures also show that the precision may start to degrade at very high values of n (e.g. $n = 12$, 14, and 16) while the maximum recall will still decrease.

Interestingly, the increased precision for M^{DF_n} over the more traditional measures is coupled with a large decrease in vocabulary size. While M^{TF} and M^{DF} produce a total vocabulary of $54,381$ words in the SIMPLE index, M^{DF_4} extracts a vocabulary of $6,834$ words only. This is 12.6% of the vocabulary size of M^{TF} and M^{DF}. The vocabulary sizes of all measures tested are shown in Table 4 for the SIMPLE and STOP indices. As we have mentioned in Section 3.4, this reduction is to be expected since there are more words with non-zero DF and TF values than those with non-zero DF_n values in the database.

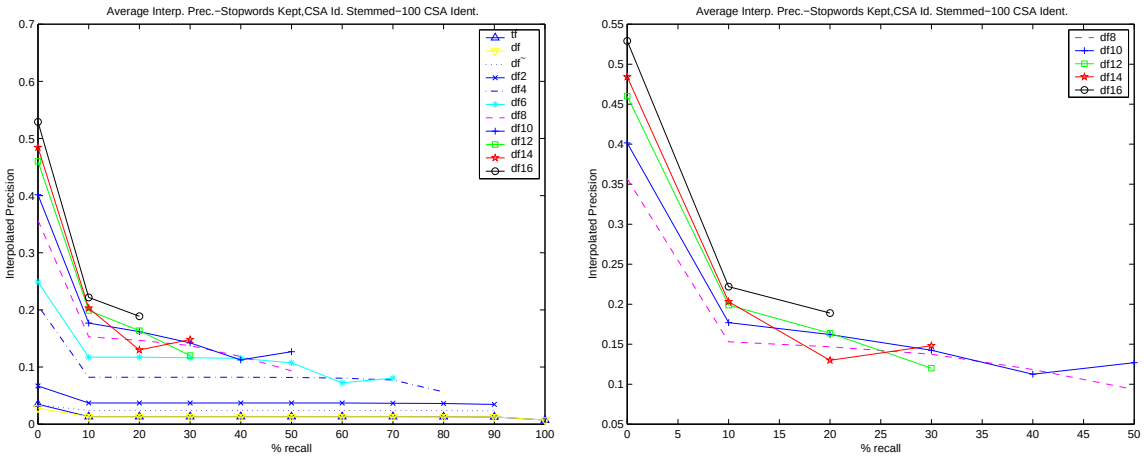
The smaller vocabulary size does have its drawback: fewer words mean a smaller chance of finding all the target words on our word list. This translates into a recall that is less than 100%. But by using reasonable values for n such as 4, 6, or 8 we improve precision by at least 30% over TF and DF without significant loss in recall.

Word lists from the other three indices produce very similar p-r graphs, which shows that the relative effectiveness of the topic relevance measures is not influenced by whether stemming or stopword removal is performed, as long as the same pre-processing functions are done for all measures uniformly. This is not to say that stemming and stopword removal have no effect on the precision of topic relevance measures. In fact we found that stopword removal improves precision while stemming degrades it. Figures 12 and 13 show the p-r graphs for each M^{DF_n} for $n = 2, 4, 6, 8, 10, 12, 14, 16$ against the top 500 target words for topic words from the SIMPLE, STOP, STEM, and S&S indices. These figures clearly show that for $n \geq 2$ stopword removal produces the highest precision while stemming always generates the least precise word lists.

The negative effect of stemming might be due to several factors: first, it could be that the genre of technical writing tends to repeat words in the same form. Also, the stemmer used might be too aggressive. The Porter stemmer

Measure	SIMPLE Voc.	STOP Voc.
$M^{DF_{16}}$	$3,111$	$2,849$
$M^{DF_{14}}$	$3,459$	$3,163$
$M^{DF_{12}}$	$3,855$	$3,542$
$M^{DF_{10}}$	$4,352$	$4,016$
$M^{DF_{8}}$	$5,400$	$5,020$
$M^{DF_{6}}$	$6,834$	$6,423$
$M^{DF_{4}}$	$9,793$	$9,357$
$M^{DF_{2}}$	$15,921$	$15,447$
M^{DF}	$26,964$	$26,478$
M^{DF}	$54,381$	$53,877$
M^{TF}	$54,381$	$53,877$

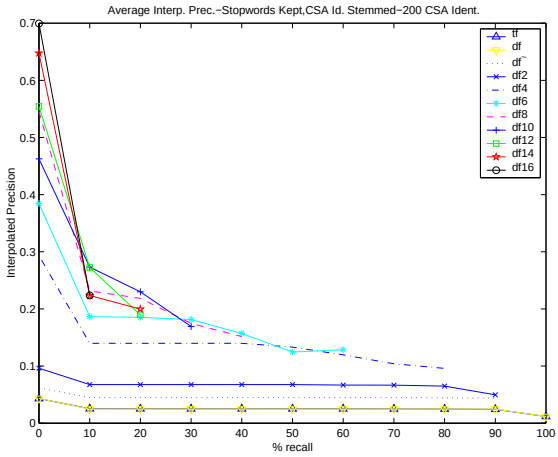
Table 4: Size of the CoRR database index used by each topic relevance measure in the SIMPLE and STOP indices.



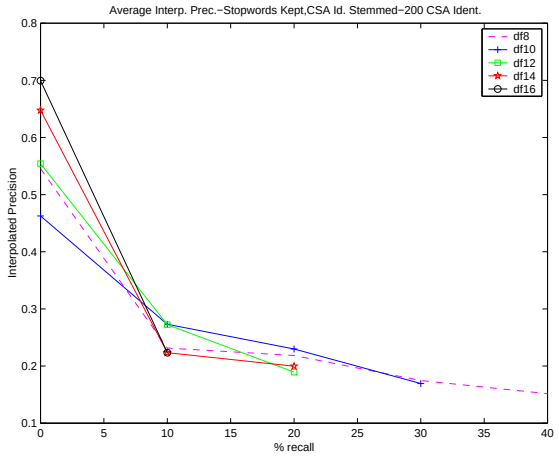
(a) The full p-r curve

(b) The curve starting at 0.1 recall

Figure 5: The average interpolated 11-point average precision-recall using a 100-word target list

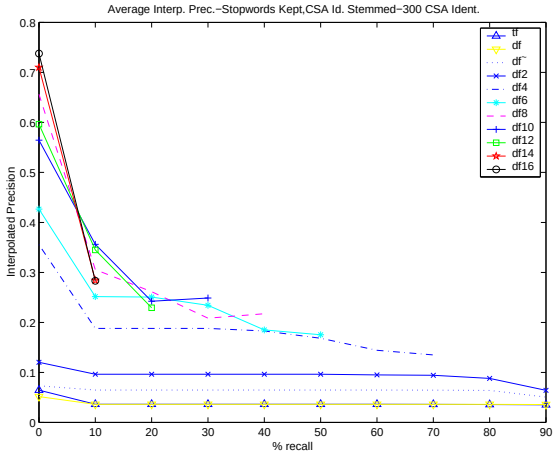


(a) The full p-r curve

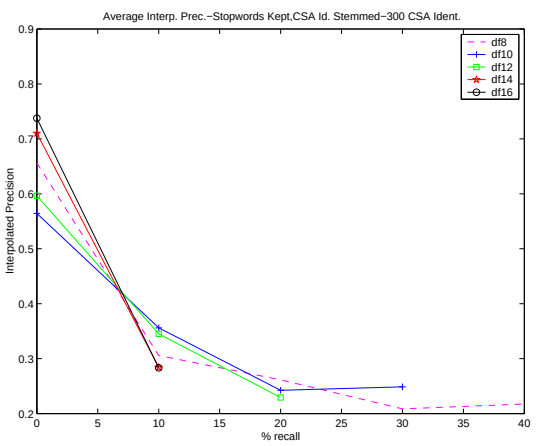


(b) The curve starting at 0.1 recall

Figure 6: The average interpolated 11-point average precision-recall using a 200-word target list

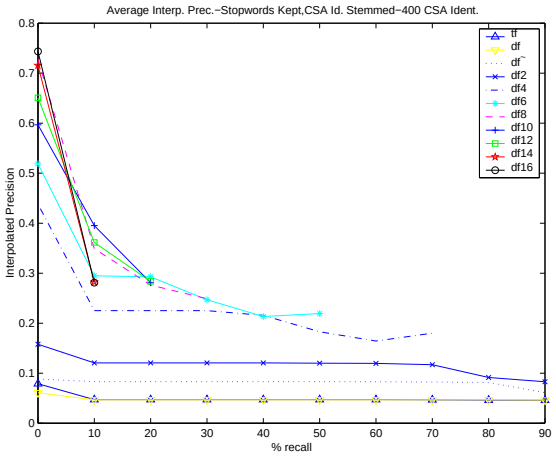


(a) The full p-r curve

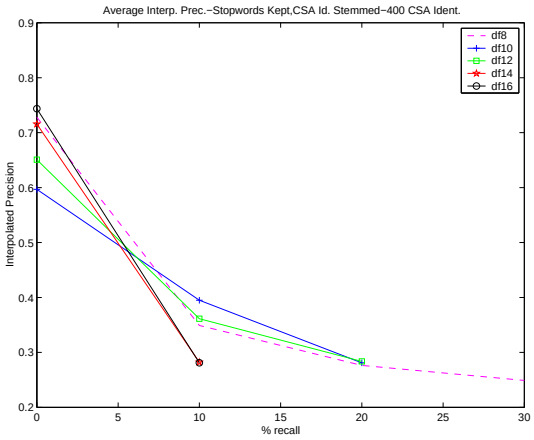


(b) The curve starting at 0.1 recall

Figure 7: The average interpolated 11-point average precision-recall using a 300-word target list

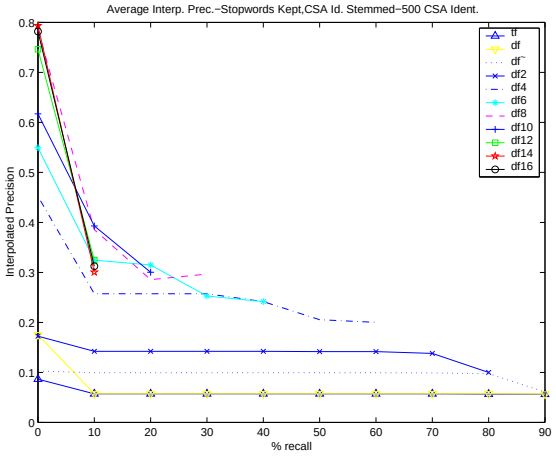


(a) The full p-r curve

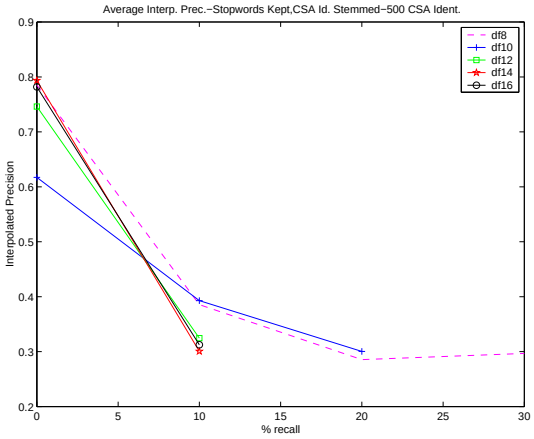


(b) The curve starting at 0.1 recall

Figure 8: The average interpolated 11-point average precision-recall using a 400-word target list

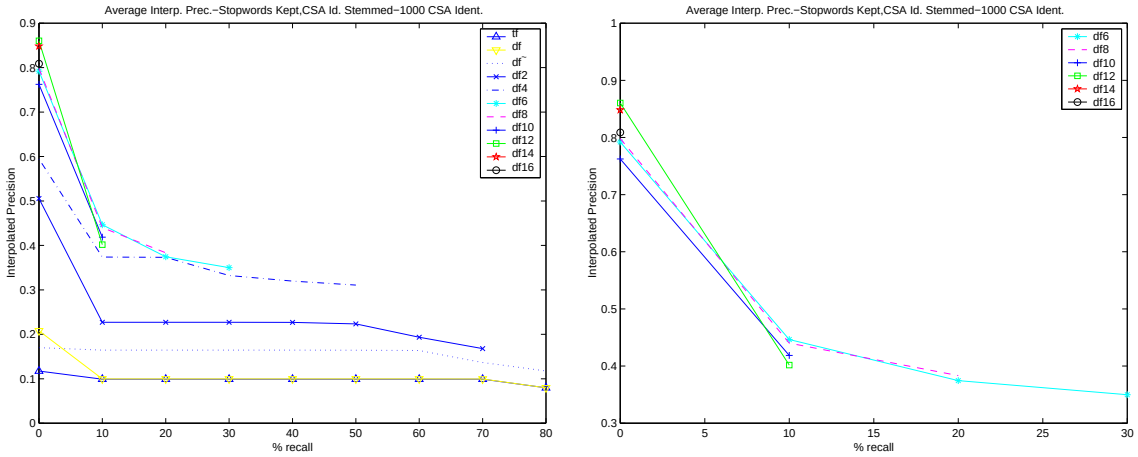


(a) The full p-r curve



(b) The curve starting at 0.1 recall

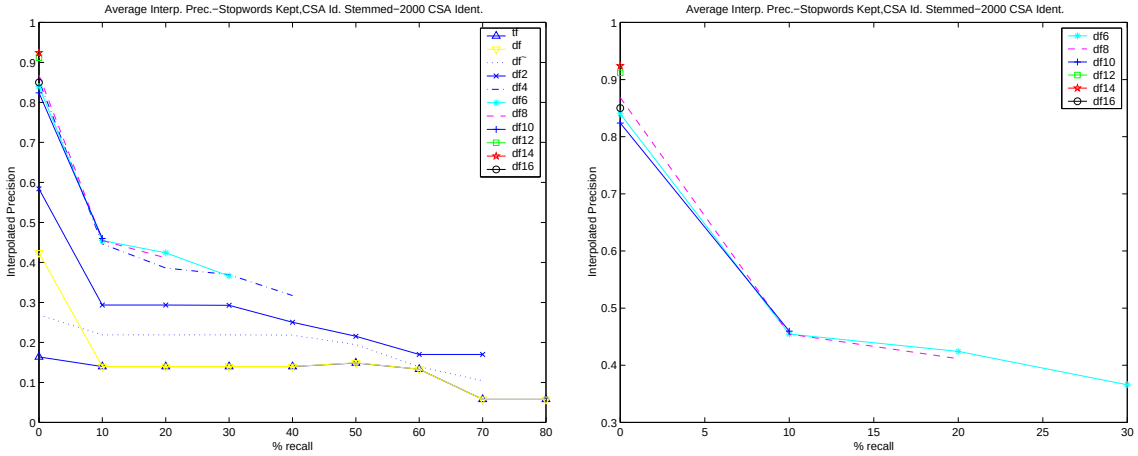
Figure 9: The average interpolated 11-point average precision-recall using a 500-word target list



(a) The full p-r curve

(b) The curve starting at 0.1 recall

Figure 10: The average interpolated 11-point average precision-recall using a 1000-word target list



(a) The full p-r curve

(b) The curve starting at 0.1 recall

Figure 11: The average interpolated 11-point average precision-recall using a 2000-word target list

reduces words to the smallest stem possible. This clusters together words that do not have a common meaning (e.g. *factory* and *factorial*) thus over-emphasizing incorrect words. A weaker stemmer that does limited stemming might have a more positive effect on precision.

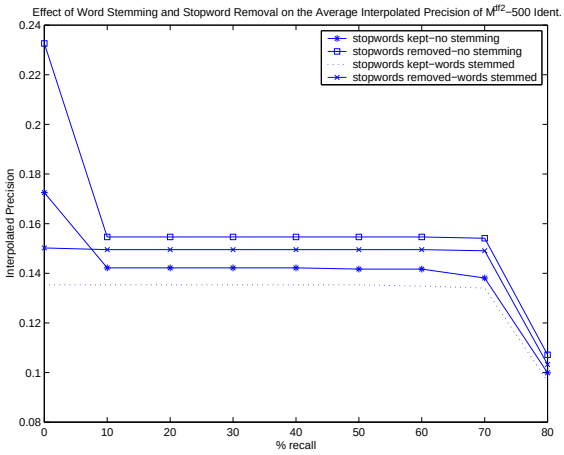
The effect of stopword removal on precision is rather surprising. Stopwords are expected to behave in a similar manner across topics, thus getting a low topic relevance weight. This seems to be the case for the most frequent stopwords, such as *the* and *in*, which are usually at the bottom of the sorted topic word list. But there are also stopwords that are more common within some topics than others. An example is *behind* which is near the top of M^{DF_4}'s word list for categories AI and CL, but is not considered a topic word for any other category.

The positive effect of stopword removal is also evident in Figures 14 and 15. In these figures we compare the unstemmed word lists obtained from the STOP and SIMPLE indices against the *unstemmed* target list which is created in the same manner as the target list we have been using so far, except that we skip the stemming step and sort CSA identifier words by their total frequency in their unstemmed form.

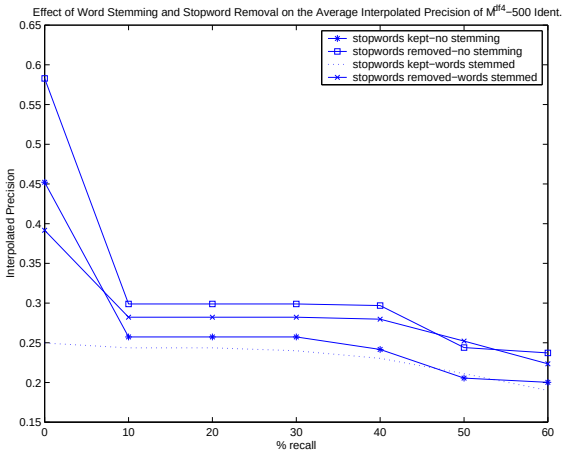
5 Conclusion

In this report we define topic relevance and distinguish topically relevant words from traditional information retrieval index terms. We then present four different measures of topic relevance, each using a different word frequency statistic. The first three of these measures are based on the document as a bag of words. These measures use frequency statistics that are independent of word position in the text to assess the word's topic relevance. The other measure M^{DF_n} is based on our view of text as a sequence of topic indicators. Under such a view the position of the word in the document can be useful in understanding the document's content. This measure evaluates the topic relevance of a word by how it spreads out in the document. We show that M^{DF_n} is more effective in selecting good topically relevant words than the other three measures, and that medium values of n, namely $n = 4, 6$, and 8, produce the best topic word lists. Moreover, these three measures generate vocabulary sizes that are between 10% and 20% of the size used by the other more traditional measures. The evaluations also indicate that stopword removal improves the precision of the topic words extracted whereas stemming degrades it.

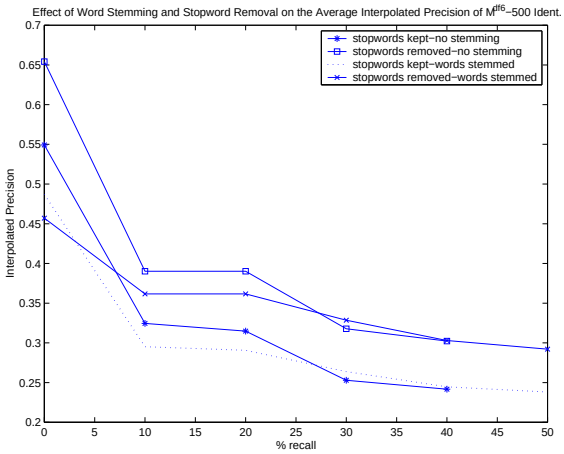
Although we use a preset stopword list for stopword removal in the evaluation, it is preferable to be able to remove stopwords automatically with no reference to a preset list. One possible approach is to sort index words by their pure DF_n database frequency and remove the top words from this sorted list. For example, if we plan to use M^{DF_4} to weigh topic words, we can start by sorting the SIMPLE index words by $DF_4(w, db)$ and removing the top 10% of the words from the index. Figure 16 compares the sorted pure frequency lists to SMART's stopword list. The plots in this figure are the traditional 11-point precision-recall plot *with SMART's stopword list as the target vocabulary.* These



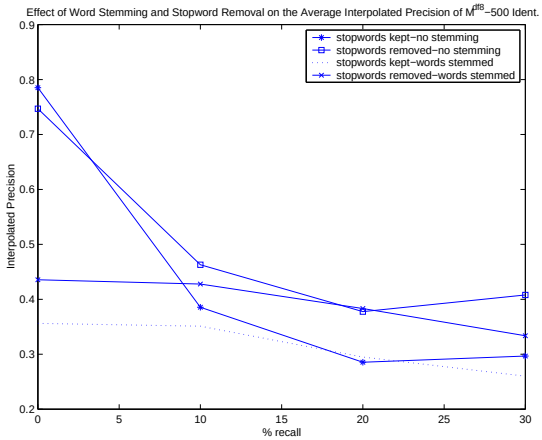
(a) M^{DF_2}



(b) M^{DF_4}

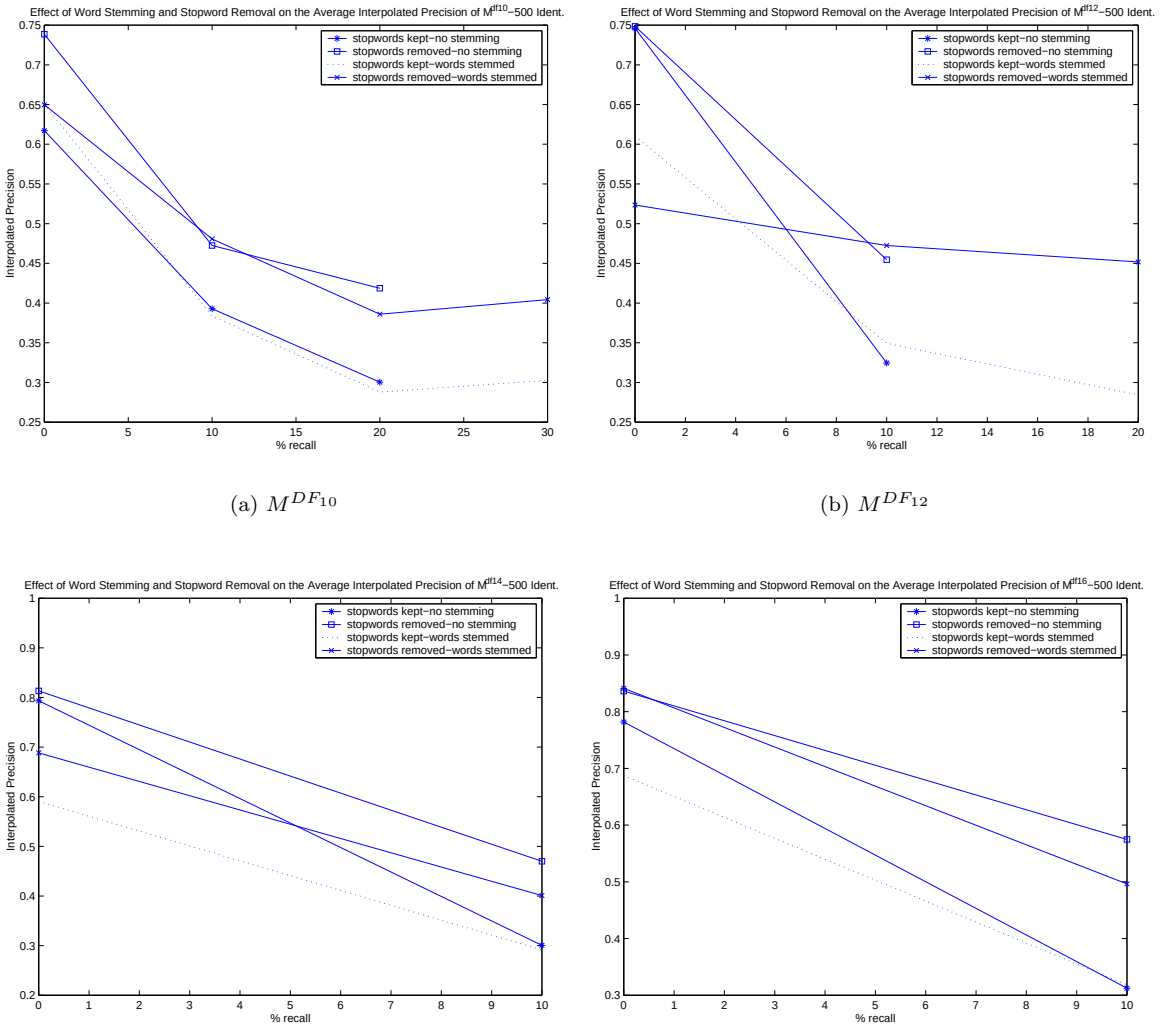


(c) M^{DF_6}



(d) M^{DF_8}

Figure 12: The average precision of the word lists generated by the M^{DF_2}, M^{DF_4}, M^{DF_6}, and M^{DF_8} measures using the SIMPLE, STOP, STEM, and S&S indices.



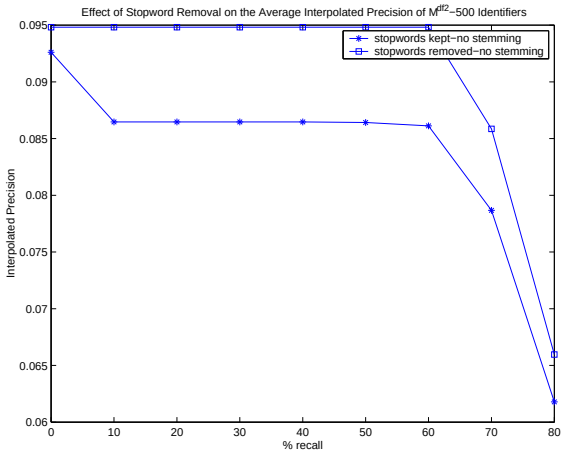
(a) $M^{DF_{10}}$

(b) $M^{DF_{12}}$

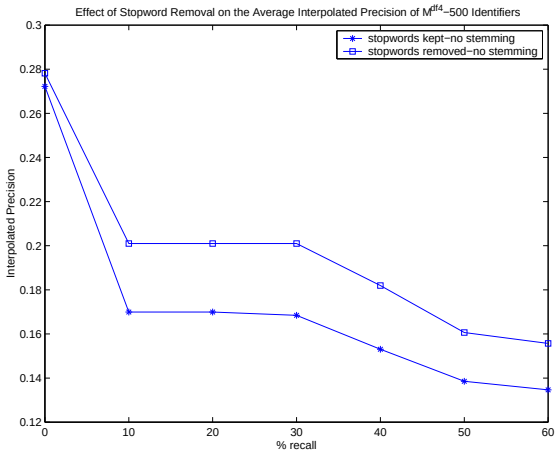
(c) $M^{DF_{14}}$

(d) $M^{DF_{16}}$

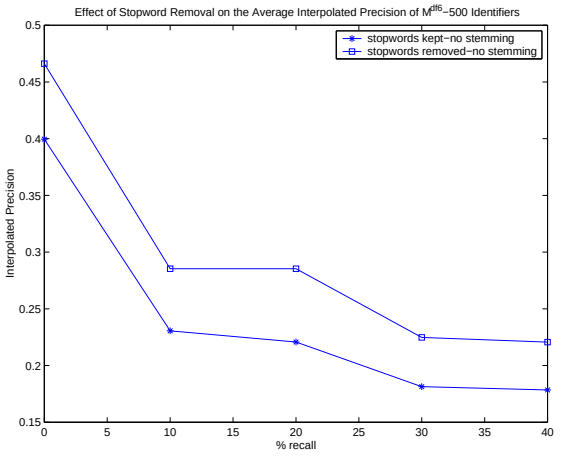
Figure 13: The average precision of the word lists generated by the $M^{DF_{10}}$, $M^{DF_{12}}$, $M^{DF_{14}}$, and $M^{DF_{16}}$ measure using the SIMPLE, STOP, STEM, and S&S indices.



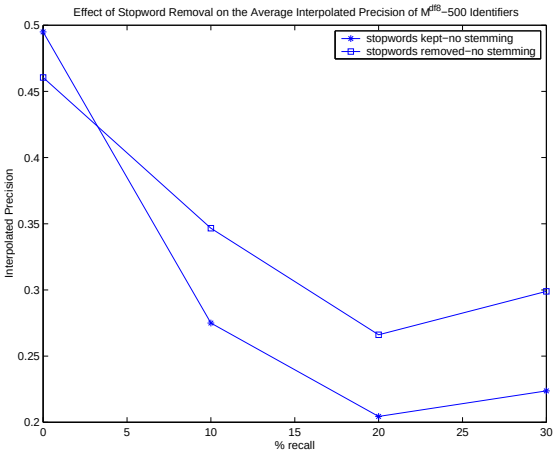
(a) M^{DF_2}



(b) M^{DF_4}

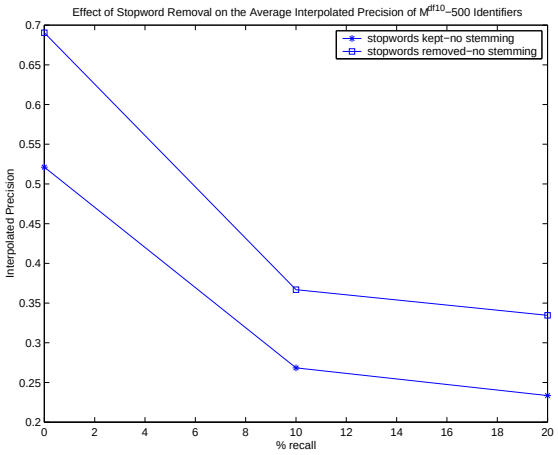


(c) M^{DF_6}

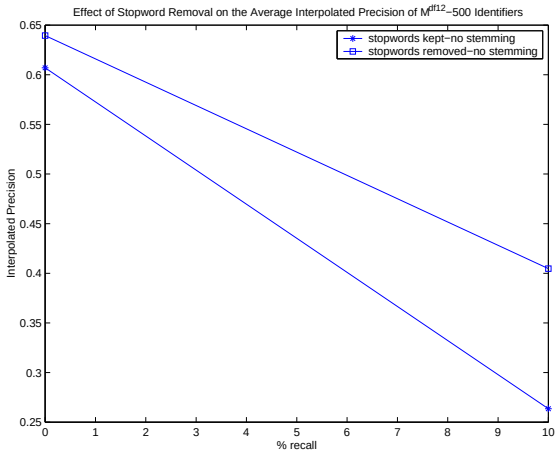


(d) M^{DF_8}

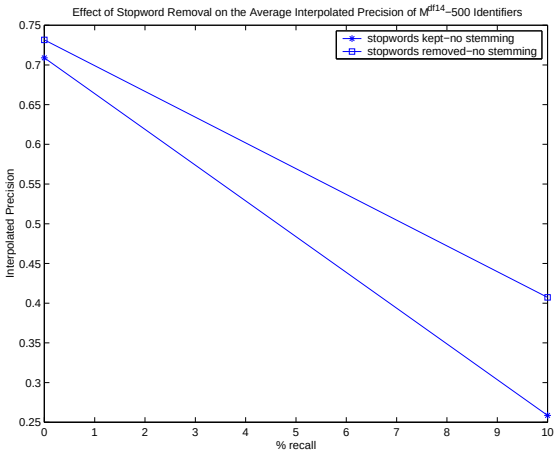
Figure 14: The average precision of the word lists generated by the M^{DF_2}, M^{DF_4}, M^{DF_6}, M^{DF_8} measures using the SIMPLE and STOP indices versus the *unstemmed* ideal list.



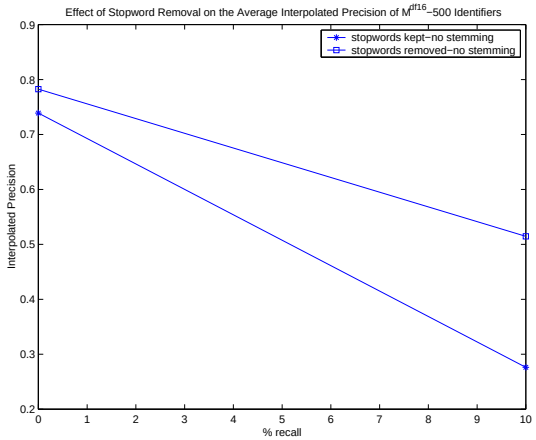
(a) $M^{DF_{10}}$



(b) $M^{DF_{12}}$



(c) $M^{DF_{14}}$



(d) $M^{DF_{16}}$

Figure 15: The average precision of the word lists generated by the $M^{DF_{10}}$, $M^{DF_{12}}$, $M^{DF_{14}}$, and $M^{DF_{16}}$ measures using the SIMPLE and STOP indices versus the *unstemmed* ideal list.

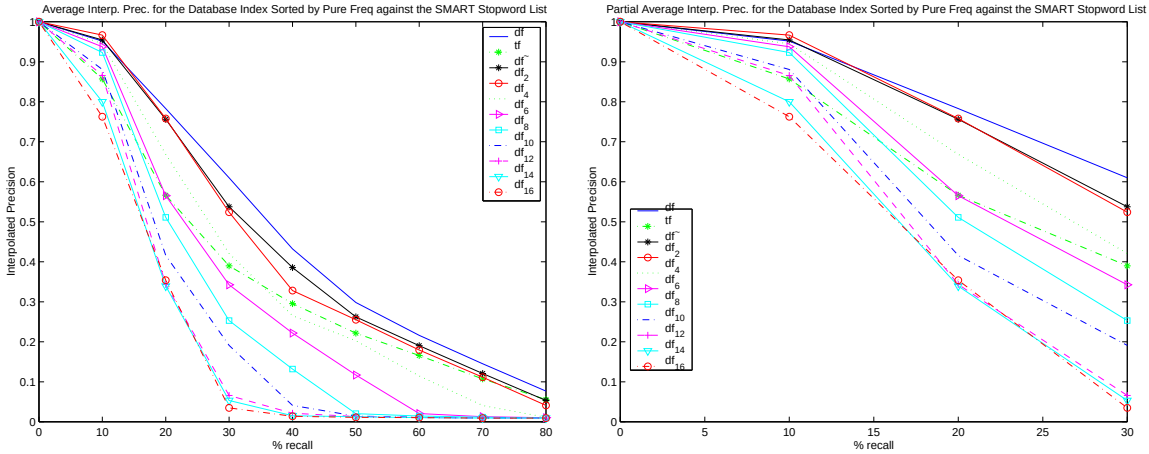


Figure 16: The precision of the SIMPLE index words sorted by their frequencies against the SMART stopword list.

plots show that a high proportion of the top words in any of the sorted lists are stopwords found on the SMART stopword list. Therefore, removing these words will likely improve the precision of our M^{DF_4} measure without losing too many non-stopwords from the list. This approach is similar to the adhoc feature selection method mentioned by Yang and Pedersen [26], where they report that removing words with the highest DF values is one of the most effective feature selection methods for text categorization. We leave further studies along these lines for future work.

In our evaluations we replaced stopwords by some null term to preserve their positions in the text. For completeness, it would be interesting to further study the effect of simply removing stopwords without preserving their position.

Another aspect we would like to investigate is the effect of a stemmer that is less aggressive than the Porter stemmer used in our evaluations. A weaker stemmer might reduce the number of words incorrectly represented by the same stem when an aggressive stemmer is utilized, but at the same time recognize words with slightly varying morphological formats, such as plurals, which would otherwise be treated as different words when no stemming is performed.

It is also interesting to explore the effect of relaxing our spread measure to allow for documents in which a word appears in m out of n segments for DF_n, for some preset value of $m < n$. This would recognize that the major topic of a paper could be interrupted in one or two places for a tangential discourse.

References

[1] The 1998 ACM computing classification system. http://www.acm.org/class/1998/ccs98.html.

[2] Baeza-Yates, R., and Ribeiro-Nero, B. *Modern Information Retrieval*. ACM Press, N.Y., 1999.

[3] Bookstein, A., Klein, S. T., and Raita, T. Clumping properties of content-bearing words. *Journal of the American Society of Information Science 49*, 2 (1998), 102–114.

[4] Church, K. Empirical estimates of adaptation: The chance of two Noriega's is closer to p/2 than p2. In *Coling* (2000), pp. 173–179.

[5] Church, K., and Hanks, P. Word association norms, mutual information, and lexicography. *Computational Linguistics 16*, 1 (1990), 22–29.

[6] Computing research repository (CoRR). http://arxiv.org/archive/cs.

[7] Computer and information systems abstracts. http://www.csa3.com/csa/ids/databases-collections.shtml.

[8] Damerau, F. J. Evaluating computer-generated domain-oriented vocabularies. *Information Processing and Management 26*, 6 (1990), 791–801.

[9] Damerau, F. J. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management 29*, 4 (1993), 433–447.

[10] DeTeX. http://www.cs.purdue.edu/homes/trinkle/detex/.

[11] Frakes, W. B. Stemming algorithms. In *Information Retrieval: Data Structures and Algorithms*, W. B. Frakes and R. Baeza-Yates, Eds. Prentice Hall, Inc., Englewood Cliffs, NJ, 1992, pp. 131–160.

[12] Harman, D., Ed. *Common Evaluation Measures* (Gaithersburg, Md, 2001), Department of Commerce, National Institute of Standards and Technology.

[13] Joachims, T. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning* (Chemnitz, DE, 1998), C. Nédellec and C. Rouveirol, Eds., no. 1398, Springer Verlag, Heidelberg, DE, pp. 137–142.

[14] Katz, S. M. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering 2*, 1 (1996), 15–60.

[15] Krulwich, B., and Burkey, C. The infofinder agent: Learning user interests through heuristic phrase extraction. *IEEE Expert 12*, 5 (1997), 22–27.

[16] Manning, C. D., and Schütze, H. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

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

[18] The Reuters-21578 text categorization test collection. http://www.research.att.com/ lewis/reuters21578.html.

[19] Salton, G. *Dynamic Information and Library Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

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

[21] Shapiro, S., Ed. *Encyclopedia of Artificial Intelligence*. John Wiley, New York, NY, 1992.

[22] The SMART stopword list. ftp://ftp.cs.cornell.edu/pub/smart/english.stop.

[23] van Rijsbergen, C. J. *Information Retrieval*. 2nd ed., Butterworths, 1979.

[24] Yahoo! http://www.yahoo.com.

[25] Yang, Y., and Liu, X. A re-examination of text categorization methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)* (1999), pp. 42–49.

[26] Yang, Y., and Pedersen, J. O. A comparative study on feature selection in text categorization. In *Proc. 14th International Conference on Machine Learning* (1997), Morgan Kaufmann, pp. 412–420.