Finding Topic Words for Hierarchical Summarization (DRAFT)

Dawn Lawrie, W. Bruce Croft, and Arnold Rosenberg
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

ABSTRACT

Hierarchies have long been used for organization, summarization, and access to information. In this paper we define summarization in terms of a probabilistic language model and use the definition to explore a new technique for automatically generating topic hierarchies by applying a graph-theoretic algorithm, which is an approximation of the Dominating Set Problem. The algorithm efficiently generates terms according to a language model. We compare the new technique to previous methods proposed for constructing topic hierarchies including subsumption and lexical hierarchies, as well as words found using TF.IDF. Our results show that the new technique performs as well as or better than these other techniques.

1. INTRODUCTION

Multi-document summarization is a research question that has gained much attention in the past couple of years. There has been a lot of work on generating natural language summaries for multiple documents, but this is feasible only for a very small number of documents. In this paper we are interested in summarization for a larger set of documents, such as a retrieved set or perhaps a collection of e-mails. In such an environment, rather than using natural language, one could design summaries based on single words or phrases. Because the amount and variability of the text in the documents, such a summary can be shorter while at the same time touching on a greater portion of the text.

We believe that finding topic terms (terms that can identify main themes in the document set) and relating these terms through the use of a hierarchical structure is a succinct way to construct a multi-document summary. The reason that the hierarchical structure is so powerful is that people find it intuitive, and it is commonly used such as in the Library of Congress, Yahoo![12], and MeSH (Medical Subject Headings) as well as in newspapers.

SIGIR'01, September 9-12, New Orleans, Louisiana, USA..

There are a number of examples of building hierarchies from terms of a document set using heuristic techniques. One example is subsumption hierarchies[8], which find term dependencies by calculating conditional probabilities of pairs of terms. A term that is dependent on another term is said to be subsumed by it. Another example is lexical hierarchies[1], which are created by identifying all phrases in a document set and finding the most frequent single words that occur in those phrases. These words become the top level of the hierarchy, while the other words in the phrases can be found at subsequent levels. These are reasonable techniques for constructing topic hierarchies and have produced fairly good results, but there is room for improvement. The main goal of this work is to develop a formal basis for the construction of topic hierarchies. We propose a technique based on a probabilistic model of the vocabulary that uses the Dominating Set Problem for graphs to choose topic terms by considering their relation to the rest of the vocabulary used in the document set.

One of the challenges faced in multi-document summarization in general, and topic hierarchies in particular, is the difficulty of evaluation. Evaluating our new technique is even more difficult because the probabilistic model uses conditional probabilities of terms. These conditional probabilities are approximated using the text of the document by considering variable window sizes, which means an optimal window size must be found. Because of this, we limit our evaluation to the top level of the hierarchy and compare it to previous techniques as well as to the top terms chosen by TF.IDF, which has been a popular technique for weighting and selecting terms [7].

In the following section, we present a more detailed description of previous heuristic techniques for creating topic hierarchies. In Section 3 we describe the probabilistic model developed to create topic hierarchies. In Section 4, we give a comparative example of the first level of the different topic hierarchies and the terms selected by TF.IDF. In Section 5, we evaluate the top level of the hierarchy. Finally, we conclude with future work.

2. HEURISTIC TECHNIQUES TO CREATE TOPIC HIERARCHIES

2.1 Subsumption Hierarchies

One method used to create a topic hierarchy is through the use of subsumption[8]. Subsumption is a means of associating terms so that the hierarchy reflects the topics covered by the documents. This association is defined by the following two conditions[1]:

$$P(x|y) \geq 0.8 \text{ and } P(y|x) < P(x|y).$$

Thus x subsumes y if the windows in which y occurs are a subset, or nearly a subset, of the windows in which x occurs. A window could be an entire document or it may be smaller.

Subsumption requires choosing a set of candidate terms. Sanderson and Croft[8] use all single words or phrases that occur in at least two documents. Conditional probabilities are calculated for all word pairs. Once all the individual subsuming relationships are found, the hierarchy is constructed in a bottom-up fashion. Because the relationships expressed in the hierarchy are transitive, a subsuming relationship (a, c) is redundant and therefore eliminated if a subsumes b and b subsumes c.

2.2 Lexical Hierarchies

Another way to create a hierarchy is by using the hierarchical structure of frequently occurring phrases. Creating such a hierarchy has been explored by many researchers including [5] and [1]. Both of these studies rely on frequently occurring words within phrases or noun compounds of a document set to expose the topics of that document set. Anick and Tipirneni[1] introduce the *lexical dispersion hypothesis* which states that "a word's *lexical dispersion* – the number of *different* compounds that a word appears in within a given document set – can be used as a diagnostic for automatically identifying key concepts of that document set."

Once the phrases are identified, they are divided into groups based on the terms that appear in the phrases. The lexical dispersion of each term can then be calculated. Anick and Tipirneni studied the effects of ranking the candidate terms based on lexical dispersion and found that in order to study the dispersion of a term throughout the document collection, it is also necessary to examine the number of documents that involve phrases using a particular term. Otherwise, a long document that uses the term a large number of times could make that term seem like a much better candidate than it actually is. As a rule, Anick and Tipirneni's technique ranked terms based on the number of documents that contributed at least one phrase if the dispersion level exceeded five phrases. The remainder were ranked by dispersion.

The hierarchy is constructed in a top-down method. Once the high level terms are chosen, the phrases contributing to its selection are examined and other words appearing in the phrases are ranked by the number of documents in which the phrase occurs. A third level exists when multiple phrases contain the terms in the previous two levels, and so on.

3. PROBABILISTIC MODEL FOR TOPIC HIERARCHIES

The goal of this work is to construct topic hierarchies for summarization, which means the hierarchy can be viewed as a summary. In this context a *summary* consists of terms that are strongly predictive of the rest of the vocabulary. This is essentially a language model view of a summary. A user would be able to use such a summary to predict occurrences of other terms. A *topic term* is one of the predictive terms in the summary. The top level of a hierarchy is a set of topic terms for the entire vocabulary. The secondary level consists of topic terms that cover the same vocabulary as its parent, thus exposing subtopics of the top level topic. This definition can be re-applied recursively for many levels.

Subsumption and lexical hierarchies are both partial summaries because they identify terms that can predict a portion of the vocabulary. We used them to determine the characteristics which should be present in a new technique: (1) top level terms co-occur with many different terms, and (2) lower-level terms are dependent on upper-level terms. A third characteristic imposed by the definition of a summary is that the topics have maximal coverage, so they can predict all of the vocabulary.

In order to fulfill the first characteristic, one must know what a co-occurrence is. The two previous techniques disagree on this point. In a subsumption hierarchy, terms co-occur as long as they occur within a few hundred words of each other. The lexical hierarchy requires that terms occur within the same adjective-noun (lexical) compound to co-occur. For the third characteristic, a decision must be made about what is the vocabulary of the document set. In subsumption hierarchies all non-stopword single words and phrases that occur in at least two documents are considered the vocabulary. In lexical hierarchies only single words appearing in a lexical compound are part of the vocabulary.

One way of making the definition operational is to capture the predictive nature of words in an entropy framework. Entropy is used as a measure of uncertainty about the vocabulary. By developing an algorithm that minimizes conditional entropy, we hoped to identify topic terms that would reduce one's uncertainty about unknown vocabulary. The weakness in this intuition is that conditional entropy values both negative and positive information equally. The highest conditional entropy occurs when a term has conditional probability zero with all terms, or when a term has conditional probability one with all terms. The first case is negative information because of the certainty that the term is unrelated to the vocabulary. The second case is positive information because the term is related to everything. Because a term is never related to every term and is rarely completely dependent on many terms, conditional entropy favors terms that occur very infrequently, even when smoothing is used to give some small probability of occurring with all terms. These terms violate the first characteristic of the summary which says that they should co-occur with many terms.

To avoid the problems with the entropy-based model, we decided to take a different approach. We use the conditional probabilities to create a probabilistic language model of the vocabulary. By recasting the language model as a graph, we can apply a graph-theoretic algorithm to find the set of terms that have maximal predictive power and coverage of the vocabulary. The graph consists of vertices that represent the terms and edges that are weighted by the conditional probabilities in the language model. Thus, our problem can be restated as the search for a set of topic term vertices that satisfies two conditions: (1) The graph must

[1] The threshold 0.8 was determined empirically.

be fully connected, indicating that topics cover the vocabulary no matter how few or many topic terms we are willing to allow, and (2) the conditional probability must be maximized. This is the Dominating Set Problem (DSP)[3] for graphs. Since DSP is NP-hard in its full generality, we develop a greedy approximation to find the topic terms for a single level of the hierarchy. The solution is implemented recursively in order to generate a complete hierarchy. A more in-depth discussion of each step follows.

3.1 Creating a Language Model

Before creating a model of the vocabulary, the candidate topic terms and vocabulary terms must be defined. For example, candidate topic terms could be defined as only those terms that are found in lexical compounds, as in the lexical hierarchy, or candidate topic terms could be restricted to those words found in the query and terms used by a query expansion algorithm in the context of retrieval to focus the hierarchy on relevant documents. This is similar to the way Sanderson and Croft[8] favor query terms and those found by Local Context Analysis[11] when constructing subsumption hierarchies. Restrictions may also be placed on the vocabulary the hierarchy should cover, such as excluding stopwords or requiring terms to occur at least twice in the documents. The experiments shown in this paper use the same set of terms for candidate topic terms and vocabulary, namely, the set of terms that occur in at least two documents; they exclude numbers and stopwords. These limitations are very similar to those of subsumption without the added knowledge of which terms are similar to the query.

Once the candidate topic terms and vocabulary are determined, the language model can be computed. The model is composed of all conditional probabilities $\mathbf{P}(A|B_x)$ where A is a candidate topic term and B is a vocabulary term; $\mathbf{P}(A|B_x)$ is computed as the number of instances in which A is x or fewer terms away from B, divided by the number of times B occurs. The conditional probability is computed directly rather than by using Bayes's Rule because the terms are actually dependent variables. If a counting method were used to compute the joint probabilities, the probability would depend on the size of the window, while the probability of a term, $\mathbf{P}(A)$ is not so dependent, making the two probabilities incompatible.

3.2 Interpreting the Model as a Graph

A graph is formed by considering each candidate topic term and vocabulary term as a vertex. This means each candidate topic term will actually be split into two vertices, one that represents it as a topic and the other that represents it as a vocabulary term. An edge exists between A and B if the probability $\mathbf{P}(A|B_x)$ is positive. This probability is used as the weight of the edge, which we call the *affinity* between two terms. However, for the dominating set problem, vertex weights are required rather than edge weights. We compute the vertex weights by summing all edges that are connected to that vertex. We can know use this bipartite graph to selected topics.

3.3 Greedy Approximation of DSP

Our premise is that the likelihood of A's being a topic term for B increases as the conditional probability, $\mathbf{P}(A|B_x)$, increases. From the graph, we want to find a set D, which is a minimum set of topics for the document set. This is a variant of the Dominating Set Problem for graphs: Given a graph $G = (V, E)$ and vertex weights w_v for all $v \in V$, find a subset of vertices $D \subseteq V$ so that every $u \in V - D$ there is a $v \in D$ for which $\{u, v\} \in E$ and such that $\sum_{v \in D} w_v$ is minimized[3]. In our work we actually want to maximize the sum of the vertex weights in D.

```
DSPapprox($G, CTT, k$)
(1)  $VT = V - CTT$
(2)  $D = \emptyset$
(3)  $VocabCovered = \emptyset$
(4)  $thresh = \mathbf{mean}(w_e(CTT, VT))$

(5)  foreach $c \in CTT$
(6)      $w_v(c) = \sum_{v \in VT} w_e(c, v)$

(7)  while ($VocabCovered \neq VT$ and $|D| < k$)
(8)      $d = argmax_{c \in CTT} w_v(c)$
(9)      $vCovered = d_v$ where $v \in d_v$ if $w_e(d, v) \geq thresh$
(10)     $D = D \cup d$
(11)     $CTT = CTT - d$
(12)     $VocabCovered = VocabCovered \cup vCovered$
(13)     foreach $v \in vCovered$
(14)         foreach $c \in CTT$
(15)             $w_v(c) = w_v(c) - w_e(c, v)$

(16) return $D$
```

Figure 1: Greedy Approximation of the Dominating Set Problem. It requires as inputs G (the graph), CTT (the candidate terms), and k (the maximum number of topics desired). The algorithm returns the topics, D, chosen which will be a complete or partial dominating set of the vocabulary.

Our heuristic solves the DSP in the topic-vocabulary affinity graph via the greedy approach of the algorithm **DSPapprox** depicted in Figure 1. This algorithm takes as inputs the graph, which consists of all vertices, edges, and edge weights (which are used to compute the vertex weights); the candidate topic terms, which are that portion of the vertices representing the candidate topic terms; and a number k that provides a cut-off for the number of topics requested. In the first line of Figure 1 we ide0ntify the vertices that represent the vocabulary. We then initialize D, the set that will hold the vertices chosen as topics and $VocabCovered$, the set that will hold all vertices that are connected to a vertex in D by an edge. Because we are trying to find true topic terms rather than just to cover the vocabulary, the mere existence of an edge is not sufficient proof that the candidate topic term is truly a topic for a particular vocabulary term. We test for validity by imposing a threshold. However, since the relatedness of documents vary, we choose a document-set-dependent threshold. For this paper we use the mean of all affinities because the mean is commonly used as a threshold; in future work we plan to experiment with other thresholds.

In the fifth and sixth lines of Figure 1, the vertex weights are calculated. These represent the sum of all the edges leading into a particular vertex. Since vocabulary terms will never be chosen as topic terms, it is necessary to calculate only

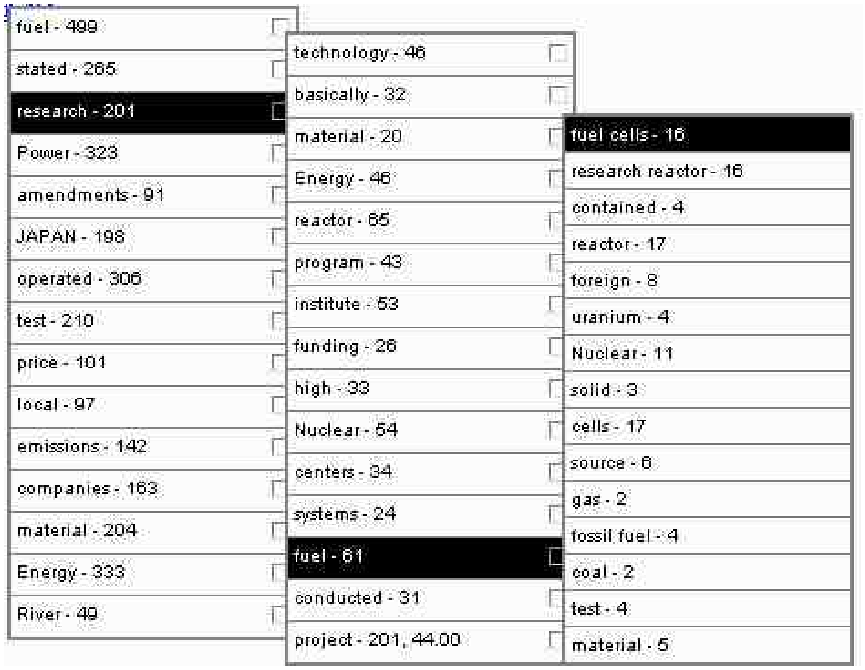

Figure 2: A Dominating Set hierarchy created for TREC query 319: New Fuel Sources, where x=5,2,1.

the weights for candidate topic term vertices.

In line eight of Figure 1, we choose the heaviest vertex, d, in the set of candidate topic terms to be a member of the set D. We then determine the set of vertices adjacent to d that are covered by the topic term, by using the threshold that was calculated. The reason that edges with weights that are less than the threshold are part of the overall weight of the vertex but are not used to determine which vocabulary terms are covered is that the accumulation of infinitesimal weights allow one to distinguish topics from the terms they point to by breaking the symmetries inherent in the affinity measure.

In order to ensure that the second topic selected has adjacencies with different terms, we adjust the weights of the vertices by subtracting the weights of edges that link candidate terms to vocabulary terms covered by d. The algorithm loops through, picking the heaviest vertex each time. At each step, the new heaviest vertex is added to the set D. We continue to augment D until either all the vocabulary vertices are in the set of covered vocabulary, or we accumulate k topic terms.

3.4 Creating the Hierarchy

Algorithm **DSPapprox** creates the top level of the hierarchy. In order to create subsequent levels, a language model is computed for each level. This models only the terms used in close proximity to the topic terms at the higher levels, and enables us to construct a hierarchy of topics, subtopics, sub-subtopics, and so on. The language model for the second level of the hierarchy is created using conditional probabilities of the form $\mathbf{P}_{C_y}(A|B_x)$, where A is the possible topic term which occurs within x or fewer terms of B, the vocabulary term as before. However, the parent term C must be with y or fewer terms of A to be considered a valid occurrence of the topic term A. By changing the allowable distance between terms, we can control how closely terms are related at different levels of the hierarchy. Once the probabilistic model is constructed, it can be turned into a graph, and the topic terms can be selected by **DSPapprox**.

3.5 Efficiency

DSPapprox is a very efficient algorithm. Given a vocabulary size of n, t candidate topic terms, and a goal of selecting k topics, the algorithm performs in $O(ktn)$ time. In contrast, the entropy-based algorithm mentioned in section 3 performs in $O(kn^3)$ where the Big-O is hiding a number of time intensive computations as well many more steps in the initialization part of the algorithm before topic terms can be selected.

4. EXAMPLE RESULTS

Evaluating automatically generated hierarchies is a particularly difficult task. Since summaries are created with users in mind, a user study is the most intuitive form of evaluation. However, user studies generally yield ambiguous results whose significance is difficult to evaluate. Recently, there have been a few interesting forms of automatic evaluation for single document summaries. In [10] the keywords found automatically were compared to the keywords named by the author of the particular document. In [2], the Open Directory Project was used, which utilizes human-generated summaries. Unfortunately, these evaluations cannot be adapted to a multi-document summary because no human generated summaries exist.

When evaluating multi-document summaries, many researchers

Subsumption		Lexical		TF.IDF		Dominating Set, x=1	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
fuel	499	fuel	499	94	124	fuel	499
boron	11	energy	323	state	256	Nuclear	286
nuclear energy	84	power	308	time	279	technology	264
energy policy	57	operate	305	fuel	499	stated	265
Nuclear Policy	56	new	302	nuclear	284	operated	306
solar	54	source	300	1994	125	program	183
power system	49	nuclear	280	require	249	research	201
energy technologies	47	state	256	service	115	reactor	238
neutron	44	plant	254	company	159	requires	249
energy conservation	43	require	249	govern	224	figure	76
high temperature	43	generate	245	amend	91	companies	163
new energy	42	electric	244	country	187	source	301
high level waste	40	reactor	235	system	213	time	285
Gaseous	39	part	227	000	8	International	196
Technology Agency	39	govern	224	japan	189	systems	214

Table 1: Lists the topics terms and number of documents those terms occur in for the top level of subsumption, lexical, TF.IDF, and the Dominating Set created using window size of one for TREC query 319.

have developed system dependent evaluations that evaluate individual parts of the system. For example, many of these summaries make use of clustering [6][9], so the quality of the cluster is useful for the evaluation. Our proposed summaries are quite different, so this approach to evaluation cannot be adapted.

In this paper we will present both qualitative and quantitative evaluations. This section shows an example of the type of hierarchy the DSP technique creates, and then manually compares the top levels of terms chosen for several window sizes of the DSP technique with the terms chosen as the top level of the subsumption and lexical hierarchies, as well as the top TF.IDF terms for the documents. In the following section we adapt the evaluation approach from [4] to get general performance measures for the top levels of the hierarchies.

4.1 Example Hierarchy

The multi-document sets that we are summarizing in this paper are the retrieved set for the TREC queries 301 to 350. The documents retrieved come from TREC volumes 4 and 5. Each retrieved set consists of five hundred documents. The example that follows is the hierarchy created from the documents retrieved for query 319, which is about new fuel sources. The hierarchy we create is not intended to be a summary for the query, but rather a summary of the documents, which ideally will expose topics related to the query as well as those that are unrelated. In the hierarchy shown in Figure 2, a couple of examples of exposing unrelated topics are the two documents about a health strike that fall under the topic 'amendments' and four documents about waste management under the topic 'River'.

We created a three-level hierarchy and asked for 15 topics at each level. The probabilistic model used to selected the top level was created by a window size of five (x=5) which is actually a window that is centered on the term and includes the five preceding and five succeeding terms. Although some of these terms are ignored because they are stopwords, numbers, or appear only in a single document, they are still used when determining the terms in the window. Once all the conditional probabilities were computed, the mean was found to be 0.0126, which is much smaller then the value required for subsumption. Figure 2 shows that the terms chosen for the top level are very general, which is what we expected when selecting topic terms that cover a large portion of the vocabulary. All five hundred documents can be found in this hierarchy.

The second level of the hierarchy computes conditional probabilities based on a window of size two. However, since this is the second level, an occurrence of a topic term is only counted when the parent term occurs within a window of size five. The mean value for the topic chosen to be the children of "research" in Figure 2 was 0.1088, which shows that the vocabulary is more uniform than at the top level. This increase in the mean is due both to the requirement that the parent be close and to the narrowing of the window size.

At the third level the conditional probabilities are based only on a window of size one, so the topics chosen are more closely related to their parents as well as being more closely related to the vocabulary that they cover. The mean for this level was 0.5748 for the topics chosen to be the children of (research→fuel). At this level of the hierarchy, the vocabulary that the topics cover is only a subset of the original vocabulary because not all terms will occur in a valid window. At the third level, a valid window is one where the parent occurs within a window size of two and the grandparent within one of five.

Figure 2 also shows how the terms become more specific at deeper levels of the hierarchy. Although it requires 59 topic terms to completely cover the vocabulary with a window size of five, the hierarchy in Figure 2 does a good job of identifying some of the topics.

4.2 Query 319: New Fuel Sources

In this section we compare the top levels of hierarchies created by different techniques from the retrieved set for TREC query 319, which asks, "What research is ongoing for new fuel sources." This is a fairly cohesive group of documents, 106 of which were judged to be relevant to the query. Tables 1 and 2 list the topics selected and the numbers of

Dominating Set, x=2		Dominating Set, x=5		Dominating Set, x=50		Dominating Set, x=100	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
fuel	499	fuel	499	fuel	499	fuel	499
research	201	stated	265	amendments	91	price	101
stated	265	research	201	price	101	amendments	91
operated	306	Power	323	research	201	samples	66
Power	323	amendments	91	samples	66	public	173
amendments	91	JAPAN	198	public	173	Power	323
plant	254	operated	306	Nuclear	286	high	250
program	183	test	210	impact	112	stated	265
reactor	238	price	101	uranium	159		
JAPAN	198	local	97	optical	11		
companies	163	emissions	142	show	107		
requires	249	companies	163	lived	65		
systems	214	material	204				
plutonium	186	Energy	333				
local	97	River	49				

Table 2: Lists the terms and number of documents those terms occur in for Dominating Sets created using window sizes of 2, 5, 50, and 100 for TREC query 319.

documents in which each topic occurs. As one can see, the topics selected by subsumption are much more closely related to the query topic than the others because it uses knowledge of the query to select them. However, these topics are so specific that there are very few subtopics, which is not a good trait in a hierarchical summary. The other techniques all choose some terms related to the topic and others which more generally capture the topics of the document set. Another noticeable difference among the topics is the number documents in which each term is found. Subsumption finds terms that divide up the documents into much smaller groups than the others. The lexical hierarchies chooses topics that are in many more documents. The smallest group contains 224 documents, which is nearly half of the set.

4.3 Query 317: Unsolicited Faxes

Tables 3 and 4 show the top levels of the hierarchies created using the retrieved set for TREC query 317: “Have regulations been passed by the FCC banning junk facsimile (fax)? If so, are they effective?” The retrieval for this set was quite poor. Only 14 of the 500 documents have been judged relevant to the query. This fall in retrieval performance is quite noticeable in the subsumption topics where the terms are much more general and also more similar to the terms chosen by other methods. This example shows that the subsumption and large window sizes for the Dominating Set choose some of the same terms to be topic terms. For example ’Markey’ is observed in DSP x=100 and subsumption. The large windows of DSP also pick up more off-query-topic words such as ’Armenia’, ’Browning’, and ’tobacco’.

5. EVALUATION

A quantitative analysis of the top level of the hierarchies follows. First, we evaluate the hierarchies’ performance on a retrieval oriented task. In this task the goal is to identify the topic term or terms, which contain all relevant documents and as few non-relevant documents as is possible. Each technique is evaluated based on how well it performs over all queries. Second, we evaluate the overlap of the terms chosen at the top level. We do this by stemming the topics and counting the number that two techniques have in common.

5.1 Comparing Techniques using Relevance

In this evaluation, we find the average number of documents read for each relevant document in the hierarchy, which is very similar to the evaluation in [4]. For example, if the hierarchy contains 20 relevant documents and 80 documents must be read before the 20 are found, then the average is 4 for the particular hierarchy, which we will call the *score* of the hierarchy. Because we do not want to have to model the order in which one might read the documents, the policy used to determine the number of documents read is that once a topic is selected, all documents attached to the topic must be read. Secondly, since documents can occur in multiple groups, a document is counted only once even if it is encountered multiple times.

Once each hierarchy has a score, we use ANOVA analysis to compare the different techniques and Tukey’s Honest Significant Difference at p=0.05 to find where there are significant performance differences in the techniques. For these experiments we looked at the effect that different numbers of topics have on the results. In this analysis we divided the techniques into a number of different groups, since differences become less clear as more techniques are added to a single comparison and as the differences among the techniques increase. Figures 3 and 4 show the results for two of the ANOVA analyses performed. Figure 3 compares the previous techniques to three very different Dominating Sets. When there are ten or more topics, the large window sizes for the Dominating Set do not perform as well as the previous techniques, although the difference is not significant. Figure 4 shows that as the number of topics is decreased, Dominating Set performs better relative to the other techniques. In fact in this comparison, the Dominating Set x=1 performs significantly better than subsumption, lexical, and TF.IDF topic terms, no matter how many topic terms are used.

When comparing different window sizes for the Dominating Set, generally smaller windows yield better performance. When looking at the smallest six, Dominating Sets with windows of 1 and 2 were not significantly different; however, x=1 was significantly better than all other window sizes. When looking at the set whose neighbors jumped by dis-

Subsumption		Lexical		TF.IDF		Dominating Set, x=1	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
fcc	203	time	302	state	218	services	331
bill	198	service	291	service	327	state	249
fax	195	new	273	time	302	Federal	271
telecommunications	160	communicate	259	america	173	Street	114
consumer	135	federal	258	house	187	market	220
legislation	130	company	245	company	247	contacts	111
d mass	130	call	242	govern	189	Computers	149
message	120	telephone	238	amend	124	Information	249
transmission	90	commission	234	work	185	company	250
markey	89	system	233	page	125	economics	66
advertiser	83	office	227	1994	149	Rep	48
rep	81	operate	219	1993	134	FCC	203
Facsimile	74	market	216	bill	198	taxes	54
ban	65	part	210	act	198	fax	194
dialers	63	bill	197	commission	262	bill	199

Table 3: Lists the topics terms and number of documents those terms occur in for the top level of subsumption, lexical, TF.IDF, and the Dominating Set created using a window size of one for TREC query 317.

Dominating Set, x=2		Dominating Set, x=5		Dominating Set, x=50		Dominating Set, x=100	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
services	331	services	331	services	331	services	331
state	249	state	249	fax	194	fax	194
Federal	271	FCC	203	FCC	203	Bonds	35
market	220	fax	194	Bonds	35	time	321
Information	249	chairman	117	bill	199	FCC	203
president	133	telephones	245	University	32	MARKEY	89
time	321	Information	249	time	321	tobacco	14
room	119	bill	199	ARMENIAN	1	Utilities	35
fax	194	market	220	messages	120	mail	117
AMERICAN	196	Street	114	sets	195		
FCC	203	time	321	law	141		
amended	124	president	133	Councilman	2		
office	231	Communications	305	selling	89		
rules	201	Administration	123	world	130		
bill	199	work	186	Browning	27		

Table 4: Lists the terms and number of documents those terms occur in for Dominating Sets created using window sizes of 2, 5, 50, and 100 for TREC query 317.

tances of 20, Dominating Sets of 20, 40, 60, 80, and 100 were all equivalent, which means that once the window size has increased to more than 20 the performance at this task is about the same.

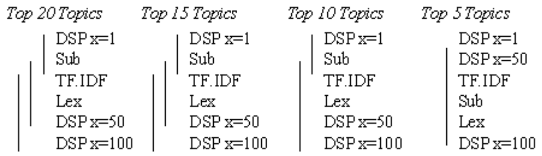

Figure 3: The ANOVA analysis for Subsumption, Lexical, TF.IDF, and the Dominating Set with window sizes of 1, 50, and 100. The bars to the left indicate where there is no significant difference between techniques. The Dominating Set with x=1 always had the highest mean score independent of the number of topics; however, only in the case of 5 topics was it significantly better than the second highest performing technique.

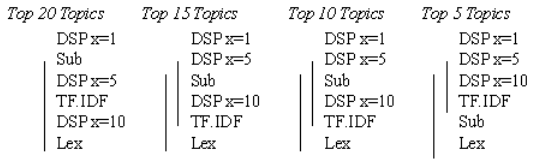

Figure 4: The ANOVA analysis for Subsumption, Lexical, TF.IDF, and the Dominating Set with window sizes of 1, 5, and 10. The Dominating Set with x=1 always has the highest mean score independent of the number of topics, and is significantly better than the second highest technique.

5.2 Measuring Overlap

Since the Dominating Set with a window size of x=1 performed the best, we compared the terms selected for each query to all the other techniques. Figures 5 and 6 show how many terms different techniques have in common with DSP x=1 using box plots. These show that the small windows of the Dominating set are most similar to DSP x=1. However, the terms selected by the lexical technique have more in common with DSP x=1 than with DSP using larger window sizes including 50 and 100 has with DSP x=1. This is not surprising because the major difference between lexical and DSP x=1 is the vocabulary used. DSP x=1 essentially uses a bigram model rather than well defined phrases. When comparing DSP x=1 to subsumption, it is interesting that the percentage of common terms increases as the number of topics decrease. This means that the two techniques are ranking the same terms very highly, but disagreeing on less prominent topics.

When comparing DSP x=1 to other Dominating Sets, overlap decreases as window sizes increase when the top 15 topics are considered. However, when the number of topics is decreased to 5, there is no longer any difference between the overlaps as shown in Figure 6.

6. CONCLUSIONS AND FUTURE WORK

The ANOVA analysis shows that the Dominating Set technique for finding topic words performs better than previous techniques used to select topic terms at the top level of the hierarchy. Comparing the overlap showed that there is some agreement about what terms should be chosen. The inspection of the two queries illustrated that the Dominating Set technique is finding general topic terms for the highest level of the hierarchy, and focuses on more specific aspects at lower levels.

In the future, we will continue work on the Dominating Set technique to find the optimal parameters for the different levels of the hierarchy, as well as experimenting with different thresholds in order to ascertain coverage. We will then be able to compare complete hierarchies to the performance of other hierarchies.

7. ACKNOWLEDGMENTS

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the National Science Foundation under grant number CCR-00-73401, and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] P. Anick and S. Tipirneni. The paraphrase search assistant: Terminological feedback for iterative information seeking. In M. Hearst, F. Gey, and R. Tong, editors, *Proceedings on the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 153–159, 1999.

[2] A. Berger and V. Mittal. A system for summarizing web pages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 144–151, 2000.

[3] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Wlt Freeman and Company, 1979.

[4] D. Lawrie and W. Croft. Discovering and comparing topic hierarchies. In *Proceedings of RIAO 2000 Conference*, pages 314–330, 2000.

[5] C. Nevill-Manning, I. Witten, and G. Paynter. Lexically-generated subject hierarchies for browsing large collections. *International Journal on Digital Libraries*, 2(2+3):111–123, 1999.

[6] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL 2000 workshop on Automatic Summarization*, 2000.

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

[8] M. Sanderson and W. B. Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 206–213, 1999.

[9] G. Stein, T. Strzalkowski, G. B. Wise, and A. Bagga. Evaluating summaries for multiple documents in an interactive environment. In *LREC*, 2000.

[10] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM conference on Digital Libraries*, pages 254–255, 1998.

[11] J. Xu and W. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, 1996.

[12] YAHOO. Yahoo. www.yahoo.com.

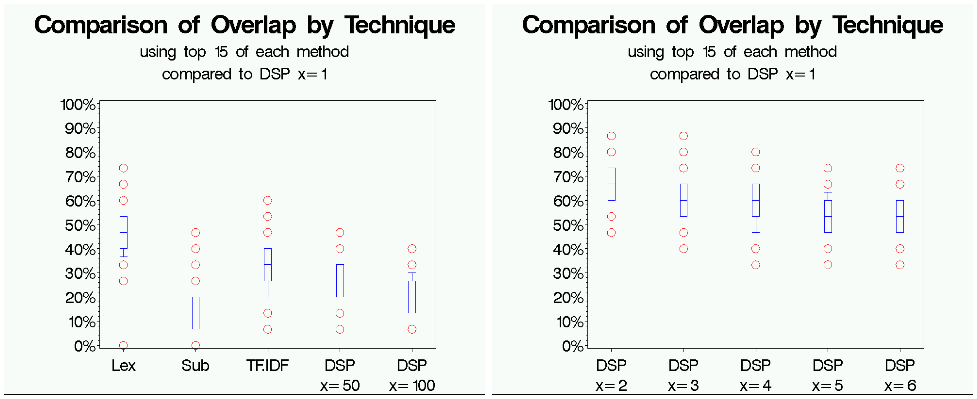

Figure 5: Illustrates the overlap of the top 15 topics between Lexical, Subsumption, TF.IDF, and a number of different variants of the Dominating Set. For each method a box plot represents the middle 50% of the overlaps ranging over the queries. The whiskers go down to the 20th percentile and up to the 80th percentile. The circles represent where points fall outside the range.

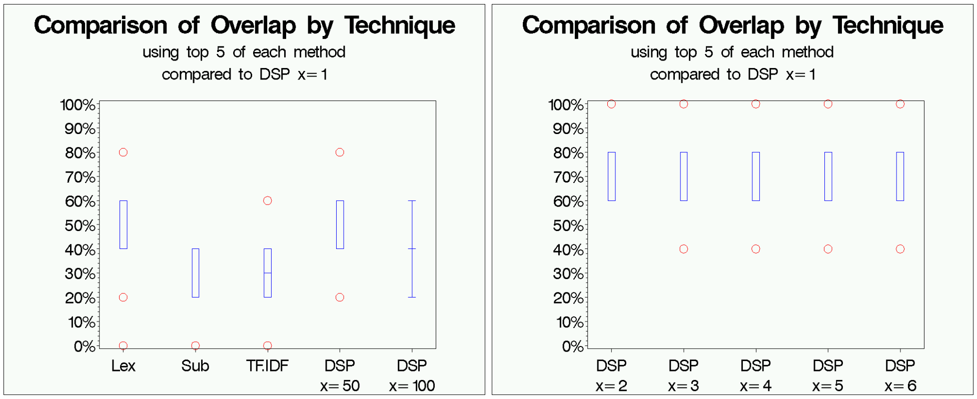

Figure 6: Illustrates the overlap of the top 5 topics between Lexical, Subsumption, TF.IDF, and a number of different variants of the Dominating Set using box plots.