Finding Topic Words for Hierarchical Summarization

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 chooses 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 the top TF.IDF terms. Our results show that the new technique consistently performs as well as or better than these other techniques. They also show the usefulness of hierarchies compared with a list of terms.

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[3, 9]. 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 number of the topics.

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![19], and MeSH (Medical Subject Headings)[8].

There are a number of examples of building hierarchies from terms of a document set using heuristic techniques. One example is subsumption hierarchies[13], 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 language models used to generate the hierarchies must also be evaluated. Since the language model captures term relatedness information, the window size that best captures dependencies must be determined. 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 [12].

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[13]. Subsumption is a means of associating terms so that the hierarchy reflects the topics covered by the documents. This association is defined by the

SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA. Copyright 2001 ACM 1-58113-331-6/01/0009...$5.00.

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[13] 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 [10, 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 a 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 fashion. 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.

From this point of view, 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. 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 this entropy-based model, we decided to take a more direct approach. We have theorized that topic terms are good at predicting other terms. Conditional probability is a measure that can be used to estimate how well one term predicts another. Most generally this approach will find topic terms by identifying a set of terms that have high conditional probability with many other terms in the vocabulary

In order to implement this 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 be fully connected, indicating that topics cover the

[1] The threshold 0.8 was determined empirically.

vocabulary no matter how few or many topic terms we are willing to allow[2], and (2) the conditional probability must be maximized. This is the Dominating Set Problem (DSP) for graphs. Since DSP is NP-hard in its full generality[4], 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[13] favor query terms and those found by Local Context Analysis[18] 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. We exclude numbers and stopwords from the vocabulary. 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 $\mathbf{P}_x(A|B)$ where A is a candidate topic term and B is a vocabulary term; $\mathbf{P}_x(A|B)$ 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 there is not a straightforward way to compute $\mathbf{P}_x(B)$. This also means the language model must be recomputed for each level in a hierarchy.

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 $\mathbf{P}_x(A|B)$ is non-zero. 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 now use this bipartite graph to selected topics.

3.3 Greedy Approximation of DSP

Our premise is that the likelihood that A is a topic term for B increases as the conditional probability, $\mathbf{P}_x(A|B)$, 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 for 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[4]. 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)  $VocabDominated = \emptyset$
(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 ($VocabDominated \neq VT$ and $|D| < k$)
(8)     $d = argmax_{c \in CTT} w_v(c)$
(9)     $vDominated = d_v$ where $v \in d_v$ if $w_e(d, v) \geq thresh$
(10)    $D = D \cup d$
(11)    $CTT = CTT - d$
(12)    $VocabDominated = VocabDominated \cup vDominated \cup d$
(13)    foreach $v \in vDominated$
(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 topic 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 identify the vertices that represent the vocabulary. We then initialize D, the set that will hold the vertices chosen as topics and $VocabDominated$, 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 dominate the vocabulary, the mere existence of an edge is not sufficient proof that the candidate topic term is 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.

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 the weights for candidate topic term vertices.

[2] We relax this requirement in the actual implementation where terms are selected until the maximum number is reached or the vocabulary is dominated. This means that some hierarchy levels are not true dominating sets.

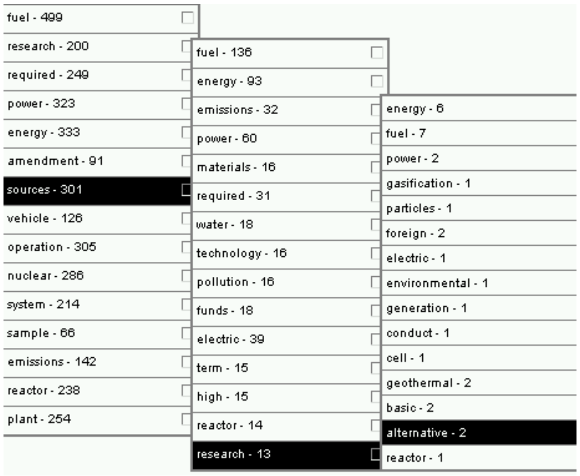

Figure 2: A Dominating Set hierarchy created for TREC query 319: New Fuel Sources, where the window size is 5 for the top level, is 2 for the second level, and is 1 for the third level (x=5,2,1).

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 dominated 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 dominated is that the accumulation of infinitesimal weights allows one to distinguish topics from the terms they dominate 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 to vocabulary terms dominated 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 dominated 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 $\mathbf{P}_{x,C_y}(A|B)$, 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 Analysis of the Algorithm

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 above 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 ascertain[5]. Especially because we believe our model needs further refinement, the hierarchies are not yet ready to evaluated by users.

Recently, there have been a few interesting forms of automatic evaluation for single document summaries. In Witten *et al*[17], the keywords found automatically were compared to the keywords named by the author of the particular document. In Berger and Mittal[2], the Open Directory Project was used, which utilizes human-generated summaries. Unfortunately, these evaluations cannot be adapted to this type of multi-document summary because no comparable human generated hierarchies exist for the document sizes we are interested in.

When evaluating multi-document summaries, many researchers have developed system dependent evaluations that evaluate individual parts of the system. For example, many of these summaries make use of clustering [11, 14], 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. Instead we use evaluation metrics that we developed for the evaluation of subsumption and lexical hierarchies[7].

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 introduce a new evaluation of the terms selected at the top level using a measure of predictiveness and adapt our previous evaluation approach to determine general performance measures for the top levels of the hierarchies compared to hierarchies with two levels.

4.1 Example Hierarchy

The multi-document sets that we are summarizing in this paper are retrieved sets for the TREC queries 301 to 350[16]. The sets consist of five hundred retrieved documents from TREC volumes 4 and 5. 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

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	energy	333
nuclear energy	84	power	308	time	279	power	323
energy policy	57	operate	305	fuel	499	nuclear	286
Nuclear Policy	56	new	302	nuclear	284	sources	300
solar	54	source	300	1994	125	technology	262
power system	49	nuclear	280	require	249	reactor	238
energy technologies	47	state	256	service	115	plutonium	186
neutron	44	plant	254	company	159	required	249
energy conservation	43	require	249	govern	224	rules	106
high temperature	43	generate	245	amend	91	research	200
new energy	42	electric	244	country	187	vehicle	126
high level waste	40	reactor	235	system	213	testing	210
Gaseous	39	part	227	000	8	plant	254
Technology Agency	39	govern	224	japan	189	materials	204

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

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
energy	333	research	200	amendment	91	amendment	101
power	323	required	249	components	93	components	93
research	201	power	323	sources	301	time	285
required	249	energy	333	contained	173	sources	301
reactor	238	amendment	91	energy	333	energy	333
plant	254	sources	301	time	285	issue	212
sources	301	vehicle	126	industrial	156		
vehicle	126	operation	305	services	135		
plutonium	186	nuclear	286	electric	261		
operation	306	system	214	issue	212		
testing	210	sample	66	trees	23		
amendment	91	emissions	142				
technology	263	reactor	238				
program	186	plant	254				

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

health strike that fall under the topic 'amendments' and two documents about muffler prices under the topic 'vehicle'.

We created a three-level hierarchy and asked for 15 topics at each level. The language model with a window size of five (x=5) was used to selected the top level terms. This window 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, we determined the mean to be 0.0173, 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 the vocabulary. All five hundred documents can be found in this hierarchy.

The second level of the hierarchy uses a language model with a window of size two. However, since this is the second level, the language model in based only on the text that is in the window size of five surrounding the parent term. The mean value for the topics chosen to be the children of "sources" in Figure 2 was 0.2891, which shows that the vocabulary has more dependencies 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 0.7796 for the topics chosen to be the children of (sources→research), shown in Figure 2. 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, the language model requires that 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 96 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 documents that have at least one occurrence of the topic term. As one can see, the topics selected by subsumption are much more closely related to the query topic than the other examples because subsumption favors phrases and other unambiguous terms. However, most of 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. For example, 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. The terms selected by Dominating Set x=100 indicate that the document set is indeed a retrieved set. A search engine believes that these documents are likely to be relevant and all of the terms chosen to describe this set excluding ‘time’ and ‘purchase’ have a fairly clear relationship to the query. As the window size decreases, it becomes more obvious why the documents have not been judged relevant.

5. EVALUATION

A quantitative analysis of the top levels of the hierarchies follows. First we introduce an evaluation that measures how well the top-level terms chosen to be part of the hierarchy predict the general vocubulary in the documents. To do this, we calculate the Expected Mutual Information Measure (EMIM)[15] between the set of topic terms and set of all non-stopwords occurring at least twice in the document set. We use ANOVA to determine where there are significant differences in performance. Second, we evaluate the hierarchies' performance on a retrieval oriented task. In this task we score each hierarchy based on the average number of documents that must be read in order to read all relevant documents. This basically finds out how well the hierarchy breaks the documents into clusters of relevant documents. We use ANOVA again to determine how well each technique performs over all queries. Third, 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 Evaluating Predictiveness

In order to find out how well the set of topic terms chosen by individual algorithms predicts the rest of the vocabulary, we calculate EMIM, which measures the extent to which the distributions of the two sets deviates from stochastic independence. The formulation of EMIM follows:

$$I(T, V) = \sum_{t \in T, v \in V} P(t, v) \log \frac{P(t, v)}{P(t)P(v)},$$

where T is the set of topic terms and V is the set of non-stopwords that occur at least twice. In order to calculate the joint probability, we use the formulation in Lavrenko and Croft[6]:

$$P(t, v) = \sum_{d \in D} P(d)P(t|d)P(v|d),$$

where D is the set of documents.

Once EMIM has been calculated for each set of topic terms, 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 ANOVA analyses performed. Figure 3 compares subsumption, lexical, and TF.IDF to three very different Dominating Sets. For sets of 5 topic terms, Dominating Set performs better than the other techniques, although the difference is not always significant. For sets of 10, 15, and 20 topic terms, subsumption is significantly better than all other techniques. Subsumption's poor performance when there is only 5 topics is most likely due to the specificity of the topic terms subsumption identifies. However, with more terms the subsumption terms predict the general vocabulary very well. In the larger groups of terms, the Dominating Sets still perform better than the lexical approach and TF.IDF. ANOVA analysis, whose results are not shown, indicate that windows sizes between 10 and 50 usually perform significantly better than very small window sizes and large window sizes. We selected two Dominating Sets, x=20 and x=30, to find if there is actually any significant differences among the Dominating Sets, the lexical approach, and TF.IDF. Figure 4 show that for all groups of terms, the Dominating Sets perform significantly better than the lexical approach, and TF.IDF.

5.2 Comparing Techniques using Relevance

In this evaluation, we find the average number of documents read per relevant document in the hierarchy, which is very similar to our evaluation in Lawrie and Croft [7]. This is calculated by dividing the number of relevant documents by the number read, which we call the *score* of the hierarchy. We use a greedy algorithm to model which documents a user would read during an exhaustive search. This algorithm models the best possible user by choosing topics that have the highest concentration of relevant documents. Because we do not want to have to model the order in which one might read the documents, the policy is to select a topic and then read all documents attached to the topic. Secondly,

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	market	220
fax	195	new	273	time	302	form	149
telecommunications	160	communicate	259	america	173	Street	115
consumer	135	federal	258	house	187	companies	250
legislation	130	company	245	company	247	rules	201
d mass	130	call	242	govern	189	stations	131
message	120	telephone	238	amend	124	bill	199
transmission	90	commission	234	work	185	Communications	305
markey	89	system	233	page	125	Director	108
advertiser	83	office	227	1994	149	fax	194
rep	81	operate	219	1993	134	Commission	262
Facsimile	74	market	216	bill	198	contact	115
ban	65	part	210	act	198	telephone	245
dialers	63	bill	197	commission	262	industry	244

Table 3: Lists the topics terms and number of documents whose terms occur in for the top level of subsumption hierarchy, lexical hierarchy, 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
telephone	245	telephone	245	fax	194	fax	194
market	220	Commission	262	Commission	262	time	321
Commission	262	States	182	time	321	FCC	203
States	182	market	220	bill	199	House	187
companies	250	Office	231	House	187	regulations	194
Office	231	FCC	203	information	249	purchase	107
Act	197	time	321	license	102	phone	145
Street	115	bill	199	months	168		
form	149	Act	198	required	265		
Communications	305	fax	194	pass	132		
required	265	Street	115				
Chairman	117	Division	81				
fax	194	required	265				
time	321	American	196				

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

since documents can occur in multiple groups, a document is counted only once even if it is encountered multiple times. It is unknown whether a user would choose the same topic terms, so we are comparing optimal performances.

Once each hierarchy has a score, we again use ANOVA analysis with Tukey's Honest Significant Difference at p=0.05. In this evaluation we are interested in both the performance of different techniques and also if a hierarchy gives any advantage to finding relevant documents. In Figure 5, we compare two-level hierarchies created using subsumption, the lexical approach, and Dominating Set to lists of the top level topics of these hierarchies and TF.IDF. When more than ten topics are 2-level Dominating Set hierarchies perform the best, although not significantly better than other 2-level hierarchies. As the number of terms allowed decreases, the single level subsumption hierarchy does increasingly well and is actually ranked higher than the 2-level DSP for five topcis, although the difference is not significant. Figure 6 focuses the analysis on just the top level of the hierarchies. Here it can be seen that subsumption is often significantly better than DSP and that the lexical approach and TF.IDF are also better than DSP, but not significantly so.

5.3 Measuring Overlap

Since the Dominating Set with a window size of x=20 performed well on the predictiveness evaluation, we compared these terms selected for each query to all the other techniques. Figure 7 shows how many terms different techniques have in common with DSP x=20 using box plots. These show that windows close to 20 for the Dominating Set are most similar to DSP x=20, which is not surprising since those language models would be most similar. A comparison of DSP x=20 to subsumption reveals that the differences observed in the qualitative analysis hold true over all queries. These two techniques have the least in common when judged by the terms each select.

6. CONCLUSIONS AND FUTURE WORK

The predictive evaluation shows that the Dominating Set does consistently well at choosing terms that are predictive of the document set vocabulary, and that large 2-level hierarchies do well in the retrieval oriented task. These two evaluations provide a partial picture of the quality of the hierarchies produced. Indeed subsumption could be viewed as the overall best technique. The problem is that subsump-

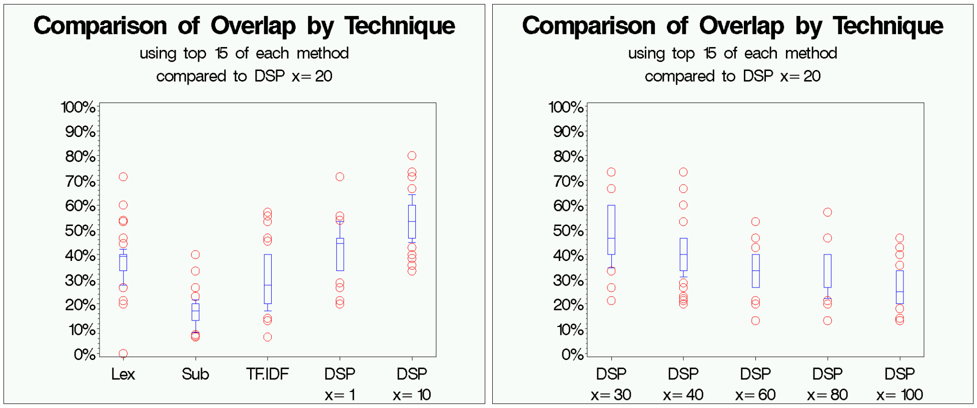

Figure 7: Illustrates the overlap of the top 15 topics between Lexical, Subsumption, TF.IDF, and a number of different variants of the Dominating Set. For each method a box represents the middle 50% of the overlaps ranging over the queries. The whiskers go down to the 20th percentile and up to the 80th percentile. The circles represent where points fall outside the range. DSP of similiar window sizes have the most overlap.

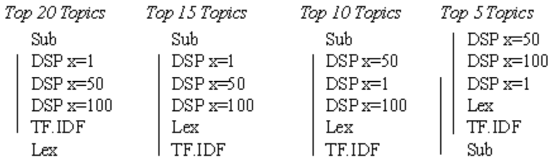

Figure 3: The ANOVA analysis of predictiveness for Subsumption, Lexical, TF.IDF, and the Dominating Set with window sizes of 1, 50, and 100. The bars to the left indicate where there is no significant difference between techniques. The Dominating Sets always had higher mean EMIM independent of the number of topics than the lexical approach and TF.IDF, and in the case of 5 topics was it significantly better than subsumption.

tion is inconsistent. The quality of retrieval seems to have a great effect on the hierarchy, which means one can be less certain of the behavior in a non-retrieval setting. However, there are ways to modify the Dominating Set so that the terms chosen would be more similar to those of subsumption by adjusting parameters and changing the language models used by the Dominating Set.

In the future, we will focus our attention on better understanding the types of terms that should occur in a good hierarchy. As indicated above, we will begin by experimenting with different language models and settings of parameters. We will also develop a more complete set of evaluations, including user studies.

We are working towards a definition of an optimal hierarchy. This definition will include analysis of both the language model and the hierarchies produced from the language model. We plan to determine which language models are best at different levels of the hierarchy, and by comparing the Dominating Set approach to a newly developed entropy based approach, determine the best method for selecting terms.

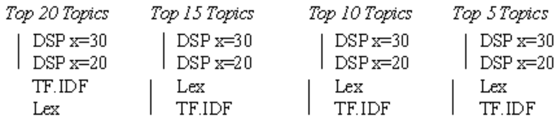

Figure 4: The ANOVA analysis of predictiveness for Subsumption, Lexical, TF.IDF, and the Dominating Set with window sizes of 20 and 30. The Dominating Sets have significantly higher EMIM than both lexical and TF.IDF.

7. ACKNOWLEDGMENTS

We would like to thank Victor Lavrenko, Chandu Ravela, Stephen Cronen-Townsend, and James Allen for their insightful discussions pertaining to this work.

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, in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912, and in part by the Advanced Research and Development Activity in Infomation Technology (ARDA) under its Statistical Language Modeling in Information Retirieval Research Program, contract number MDA904-00C-2106. 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.

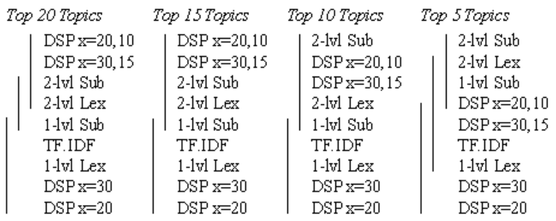

Figure 5: The ANOVA analysis of relevancy for Subsumption, Lexical, TF.IDF, and the Dominating Set for single level and 2-level hierarchies. When more than 10 topics are included, the 2-level Dominating Set peforms best, but not significantly better. Single level subsumption does extremely well and surpasses the 2-level DSP where there are only 5 topics.

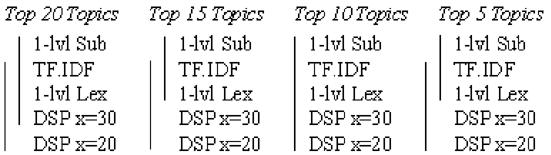

Figure 6: The ANOVA analysis of relevancy for Subsumption, Lexical, TF.IDF, and the Dominating Set with window sizes of 20 and 30. Subsumption is significantly better than DSP and both TF.IDF and the lexical approach are ranked higher than DSP no matter the number of topcis.

[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] J. Carbonell and J. Goldstein. Use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

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

[5] M. Hearst. User interfaces and visualization. In R. Baeza-Yates and B. Riberio-Neto, editors, *Modern Information Retrieval*, pages 257–323. ACM Press Series, 1999.

[6] V. Lavrenko and W. Croft. Relevance-based language models. In *Proceedings on the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page to appear, 2001.

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

[8] H. Lowe and G. Barnett. Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. *Journal of the American Medical Association*, 271(4):1103–1108, 1994.

[9] K. McKeown, J. Klavans, V. Hatzivzssiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 453–460, 1999.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] C. van Rijsbergen. *Information retrieval.* Butterworths, London, second edition, 1979.

[16] E. M. Voorhees and D. K. Harman, editors. *The Sixth Text REtrieval Conference (TREC-6).* Department of Commerce, National Institute of Standards and Technology, 1997.

[17] 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.

[18] 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.

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