2001

LANGUAGE MODELS FOR HIERARCHICAL SUMMARIZATION (PROPOSAL FOR DISSERTATION)

Dawn Lawrie
University of Massachusetts Amherst

LANGUAGE MODELS FOR HIERARCHICAL SUMMARIZATION (PROPOSAL FOR DISSERTATION)

A Dissertation Outline Presented

by

DAWN LAWRIE

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2001

Computer Science

LANGUAGE MODELS FOR HIERARCHICAL SUMMARIZATION (PROPOSAL FOR DISSERTATION)

A Dissertation Outline Presented

by

DAWN LAWRIE

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

Donald Fisher, Member

Arnold Rosenberg, Member

W. Bruce Croft, Department Chair
Computer Science

ABSTRACT

LANGUAGE MODELS FOR HIERARCHICAL SUMMARIZATION (PROPOSAL FOR DISSERTATION)

MAY 2001

DAWN LAWRIE

A.B., DARTMOUTH COLLEGE

M.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Hierarchies have long been used for organization, summarization, and access to information. In this proposal we define summarization in terms of a probabilistic language model and use the definition to explore new techniques for automatically generating topic hierarchies. One technique applies a graph-theoretic algorithm, which is an approximation of the Dominating Set Problem. Another technique uses an entropy-based approach to choose topic terms. Both techniques efficiently select terms according to a language model. We compare the new techniques to previous methods proposed for constructing topic hierarchies including subsumption and lexical hierarchies, as well as words found using TF.IDF. Our preliminary results show that the new techniques perform as well as or better than these other techniques. We plan to evaluate the two techniques further through user studies as well as computer simulations. We will also develop a demo for better interaction with users.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

This proposal presents methods of automatically generating a summary for a large group of documents. It is particularly difficult to present a summary for hundreds of documents, especially when there is no assumption that the documents cover the same topics. Recently there has been a large body of research on developing natural language summaries, which try to string together sentences or sentence fragments from the text[3, 4] or generate sentences to form a summary[2, 20]. When dealing with a few hundred documents, such a solution is impractical because of the myriad of topics one expects to find. Instead, we focus on term based summaries which can be much more compact than a natural language summary.

Unlike traditional abstract-like summaries, which at some level can act as a substitute for the original text, our term-based summaries are intended to act as a guide to which documents are useful and to give the user a general sense of the topics covered in the documents. Such a summary could be created for a user's e-mails. From the summary, one would learn the topics of the e-mails expressed through keywords and most likely, the people who write about particular topics. If the user were looking for e-mails on a specific topic, the summary could help identify which e-mails would most likely contain the specific topic.

There are three main challenges related to the automatic creation of term-based summaries. The first is the selection of terms. The second is the presentation of these terms in a coherent fashion to the user, and the third is the evaluation of the summaries in a meaningful way.

In order to automatically generate a term-based summary, we must first determine how to identify which terms should be part of the summary. We propose using a probabilistic model of the language used in the text of the documents, which is described in Lawrie *et al.*[14]. The model incorporates probabilities of individual terms in the text and conditional probabilities of pairs of terms. Through the language model we have an abstract view of the text which makes it easier to find the terms that convey the most information about the text as whole. We present two different methods that use the language model to determine which terms are best at conveying the topics in the documents. The first method uses the model to construct a weighted graph which then determines the terms that dominate the rest of the vocabulary by using a greedy approximation to the dominating set problem[14]. The second method looks at the problem from a compression point of view to find the topic terms. It selects terms that minimize the average code length required to express the entire vocabulary in the documents through an entropy equation.

Once the topic terms are selected, they need to be conveyed to the user. There are many examples of listing the terms by frequency or another metric[5, 9]. The main problem with a list is that it provides no indication of how the terms in the list are related. Instead, we propose to use a hierarchy because of the information the structure provides about how terms relate to one another.

Finally, the issue of evaluation must be addressed. Evaluating the hierarchies is a particularly difficult problem because we cannot directly compare our hierarchies to what a human might generate. The manually produced hierarchies that exist are very general such as the Yahoo! hierarchies[27] or MeSH[16]. Instead, we have focused on evaluations that can be simulated by a computer, which provides a good metric for comparing hierarchies. This method of evaluation has shown the usefulness of clustering in heuristic approaches to generating a hierarchy [13], and that the language model approaches are as good as or better than the heuristic approaches at identifying terms that break documents into relevant and non-relevant groups [14]. However, this type of evaluation does not answer the

question of whether the information contained in the hierarchy conveys the topics of the documents and whether it is a useful tool for information-related tasks such as browsing and finding relevant documents.

This research makes three general contributions:

- We will develop a definition of an optimal hierarchy for summarization and information related tasks. With such a definition, we will have concrete attributes that can be measured to determine which hierarchies are best.

- We will provide the first formal framework for choosing topic terms for the hierarchies. The advantage of having a formal framework is the existence of well-defined ways to modify the framework to make improvements.

- We will develop evaluation measures for hierarchical summarization using large document collections. In order to develop human-usable hierarchical summarizes, the current methods of evaluation must be expanded to encompass a broader range of attributes. These measures will be able to verify whether the hierarchies constructed using our formal framework are in fact optimal hierarchies.

In the remainder of the proposal, we discuss work related to summarization, topical hierarchies, visualization, and evaluation in Chapter 2. In Chapter 3, we summarize the research that has already been completed, and in Chapter 4 we present our plan of future work. In Chapter 5, we conclude with the research contributions of the thesis proposed.

CHAPTER 2

RELATED WORK

2.1 Summarization

Luhn began working on the problem of automatic summarization in the 50's[17]. Research on the topic has intensified in the past decade with the advent of the World Wide Web. Automatic methods of summarization have used three main approaches: linguistic (e.g. [19, 21]), statistical (e.g. [2, 3]), and combinations of the first two approaches (e.g. [6]).

Most of this work focuses on single document summaries created by extracting sentences or sentence fragments. Kupiec *et al.*[11] is one example of this work. They use a training set of documents with hand-selected document extracts to develop a classification function that estimates the probability a given sentence is included in a summary. Sentences are ranked according to this probability, and a user-specified number of top-ranked sentences are selected.

An example of generating summaries is Berger and Mittal[2]. They create a language model of the document, select terms that should occur in a summary, and then combine it with a trigram language model to generate readable summaries.

Multidocument summarization has mainly focused on the similarity among a group of documents [3, 20]. Carbonell *et al.*[3] finds passage similarity using Maximum Minimal Relevance for multiple documents on the same topic. They organize the top-ranked sentences in their original order within the documents. McKeown *et al.*[20] extract features from the document set. They then determine which themes are similar in order to identify phrases that should occur in the summary. Finally, a sentence generator is used, which

orders the phrases by the earliest date they occurred and employs a language generator with complete grammar rules to combine the phrases. Mani and Bloedorn[18] incorporate some of the differences between a pair of documents in their summaries. To do this they represent each document as a graph, where terms are nodes and edges correspond to semantic relationships between terms. They use spreading activation to find nodes that are semantically related to the topic. Activated graphs of two documents are matched in order to find a graph corresponding to similarities and differences between the pairs. The system outputs the set of sentences containing the shared terms and the set of sentences covering the unique terms. The first two examples focus on the similarity among documents, and they can present concise summaries of at most 25 documents. However, in the third example where differences are included, only two documents are used in the summary. In order to summarize the similarities and differences of many more documents, a term-based summary is advantageous because it can remain concise while covering the diversity of topics.

The research cited above all focus on generating some sort of natural language summary that acts in the same way as an abstract. Term-based summaries also exist. An example of a single document term-based summary is the list of keywords that appear at the beginning of articles. Kea [26] is an example of such a single document summarizer. It uses machine learning to identify features that are characteristic of keywords, and the trained system selects keywords for documents where the author did not give any. An example of multidocument term summaries is a hierarchy which will be discussed further in the following section.

2.2 Topical Hierarchies

The work on topical hierarchies can by divided into two main sets based on the way that documents are grouped together or clustered. One body of research has used traditional clustering algorithms [25] to induce a hierarchical structure in the documents[5, 28, 9].

Membership in a particular group is determined by the presence of a number of features, but no one feature in particular must be present, which is referred to as polythetic clustering. The other body of research makes use of monothetic clustering[1, 23, 13, 14]. In this case, documents are grouped together because of the presence of a particular feature. The advantage of monothetic clusters is that such clusters are very easy to describe. The particular feature required for membership in the cluster is a satisfactory label. In general, the labels of polythetic clusters are a list of top-ranked features. Invariably, some of the documents that are part of the cluster have none of the features that are mentioned in the label and in a sense are surprise elements, since a user would have no knowledge of their presence.

A thorough review of topic hierarchies that use monothetic clustering is described in Subsections 2.3 and 2.4 of Lawrie and Croft[13].

Of the polythetic clustering approaches, Scatter/Gather is the most well known[5]. It clusters documents into five groups and labels the clusters with the top eight terms. A user is expected to pick one or two clusters which are thought to contain relevant documents. The selected groups are reclustered and again labeled. The user continues this process of drilling down until a satisfactory group of documents is gathered. Yang *et al.*[28] extends Scatter/Gather to the task of topic detection and tracking. They cluster the entire corpus of a few thousand documents and use the top five ranked terms to describe the clusters. They envision that a complete system would allow the user to drill down to the particular level of granularity that met the user's need and then use a single document natural language summarizer to learn about the specific contents of the documents.

Hofmann[9] uses a different clustering algorithm based on an annealed version of the Expectation-Maximization algorithm to produce a hierarchical probabilistic clustering of the documents. Like Scatter/Gather and Yang *et al.*, documents are expected to fit into a single place in the hierarchy, which assumes that documents are about a single topic. Because of the probabilistic model used in clustering, the model can be used to summarize

a particular cluster rather than most frequent terms. Hofmann argues that these terms are more discriminating than the most frequent terms, but this description still suffers from the same problem of topics that appear at lower levels of the hierarchy but are not even hinted at in higher levels.

2.3 Visualization

There are several ways in which a hierarchy can be displayed to a user. The 2-dimensional methods include: a node-link graph diagram (e.g. [10]), embedded folders (e.g. [22, 15]), popup menus (e.g. [23, 13, 14]), and hyperlinks (e.g. [27]). There are also a few 3-dimensional methods of displaying hierarchies including Cat-a-cone[8] and the hyperbolic browser[12].

Each of the 2D methods has strengths and weaknesses. From a user's point-of-view, a hyperlink hierarchy is the weakest of the four because a user cannot see the siblings or ancestors when he reaches a dead-end. It also provides no way to keep track of many different places of interest at the same time [7]. However, the hyperlink visualization provides an intuitive way to incorporate documents and even short descriptions of the document, by placing the documents on the page along with the children. Popup menus are opposite of the hyperlink hierarchy. Where a hyperlink hierarchy shows one level of the hierarchy and the documents related to the level, a popup menu can easily show four or more levels of the hierarchy and all the siblings of ancestor nodes on a single screen. Levels are revealed instantly when the mouse is pointing to the parent, without any clicking. This allows the hierarchy to be explored very quickly, but occasionally it can be difficult to view a particular part of the hierarchy because items are so close together. Node-link graph diagrams and embedded folders fall somewhere between the previous two in their ability to display portions of the hierarchy on a screen. Node-link diagrams require that the arrows connecting nodes use part of the screen real estate, but such a display has more permanence than the popup menu. Neither the popup menu nor the node-link diagram offers an intuitive

way to display documents within the structure, so a separate view is required. Embedded folders offer a natural way to include the documents, as is done in the display of computer directory systems. The biggest drawback to using folders is that everything is displayed in a vertical column, so most of hierarchy will either be above or below the current view.

Some more advanced methods use 3D to display more of a hierarchy on a single screen. Cat-a- cones makes use of circles set on edge[8]. Labels are attached to the edges, like children on a Ferris Wheel, so that some categories will be in the foreground while others are obscured. Large segments of a hierarchy can be displayed, but only the siblings that are close to the label of interest will be easy to view. The hyperbolic browser focuses the attention of the user on those sections of the hierarchy that are physically close together[12]. As the levels of the hierarchy get farther away, the labels used to describe the attributes become smaller until they are unreadable. The user drags different parts of the hierarchy to the center of the display to view a particular section.

2.4 Evaluation

Evaluating the topic hierarchies is a very challenging task. Most evaluations of summaries have used component-wise methods as discussed in Section 4 of Lawrie *et al.*[14]. Thus far we have compared the hierarchies based on their ability to locate relevant documents as presented in Section 5 of Lawrie and Croft[13]. This particular metric scores the hierarchy based on the total number of documents and menus one must examine in order to find all relevant documents in the hierarchy with respect to the query. It turns out this metric is biased towards hierarchies that break the document sets into very small groups, so in the end the number of documents read greatly outnumbers the number of menus examined even when there are only small numbers of relevant documents. This means that if leaf nodes have 10 or fewer documents, the algorithm will be able to choose nodes that add only a few non-relevant documents to the score. Conversely, if the hierarchy has leaf nodes with 50 or more documents, many more non-relevant documents will contribute to the score. We

have also examined the similarity among hierarchies by calculating the overlap between two hierarchies, explained in Section 5 of Lawrie and Croft.

CHAPTER 3

RESEARCH SUMMARY

One of the main contributions of this work will be the development of two techniques that use language models to choose topic terms for a hierarchical summary. In Section 3.1, there is a overview of the Dominating Model. A longer discussion and experiments can be found in Lawrie *et al.*[14]. In Section 3.2, we present a entropy-based approach to selecting topic terms that tries to minimize the average code length required to express the vocabulary. We also include a preliminary evaluation of the Entropy Model that mirrors the one in Lawrie *et al*. This evaluation is included because the results have not been publisher anywhere else. Section 3.3 consists of an evaluation of two-level hierarchies created using the Dominating Model (DSP), subsumption, and the lexical approach. We have not yet completed this analysis for the Entropy Model. It is left for future work. Section 3.4 concludes with a summary of results.

3.1 Dominating Model

The Dominating Model captures two main characteristics observed in the subsumption and lexical hierarchies. Subsumption and lexical hierarchies both identify terms that have many dependents. In fact, subsumption defines a vocabulary term, v, to be a dependent of a topic, t, when $\mathbf{P}(t|v) \geq 0.8$. Topics are term that have many dependents and are not themselves dependents. In contrast, topic terms in lexical hierarchies are chosen because of the frequency of the term in phrases. The phrases are the dependents of the topic terms.

In order to express this in terms of the conditional probability $\mathbf{P}(t|v)$, a set V is defined to be the set of all phrases in the document set. Therefore,

$$\mathbf{P}(t|v) = \begin{cases} 1 & \text{if } t \text{ is a term in the phrase} \\ 0 & \text{otherwise} \end{cases}$$

The topic terms are those terms that maximize the function: $\sum_{v \in V} \mathbf{P}(t|v)$. Thus, both techniques find the terms using the same conditional probability. One way to combine the two approaches uses the following equation to find a set of terms, T, that maximizes dependents over the vocabulary, V:

$$\mathrm{S}(V^*) = \arg\max_{T} \left[\sum_{t \in T} \sum_{v \in V} \mathrm{p}(t|v) \right] \tag{3.1}$$

Equation 3.1 completely defines the lexical hierarchy. However, the second qualification of the subsumption is not addressed by the equation: the notion of choosing topic terms that have dependents from different parts of the vocabulary. To ensure this quality, let us define V_i to be the set of vocabulary terms dependent on t_i and V_T to be the union of all vocabulary sets. The independence criterion can be expressed in the following manner:

$$\forall t_i \in T, v_i \cap V_{T-i} = \emptyset \tag{3.2}$$

The Dominating Model combines Equations 3.1 and 3.2 in order to select topic terms that maximize dependents, while ensuring that the topics represent different parts of the vocabulary:

$$\mathrm{S}(V^*) = \arg\max_{T} \left[\left(\sum_{t \in T} \sum_{v \in V} \mathrm{p}(t|v) \right) \text{and} \left(\forall t_i \in T, v_i \cap V_{T-i} = \emptyset \right) \right] \tag{3.3}$$

The implementation of this model uses an approximation of the Dominating Set problem for graphs and is described in Lawrie *et al*.

3.2 Entropy Model

Information theory was first developed by Claude Shannon in the 1940s to address the problem of maximizing the amount of information that one can transmit over a noisy channel[24]. We extend this point-of-view to summarization. The goal of summarization can be thought of as explicitly communicating the smallest amount of data that allows the user to, in some sense, "know" the information contained in the documents. Since people are good at making associations, by revealing one term in the document set, a user will believe other terms are present as well. For example, when a user sees the term "tree", she thinks of plants that usually grow quite tall with a trunk of some sort, leaves, and roots.

When a user is presented with an unknown set of documents, she has no idea of what the topics of the documents are. We can model this uncertainty using a language model of general English, which means that the expectation of terms is the same for any set of documents. We approximate this model by using the language model of the document set we are summarizing. We use entropy, which is the average uncertainty of a single vocabulary term, to quantify the effort required to know the vocabulary. An entropy measure is used to find the best set of topic terms through a minimization algorithm. We begin with the traditional definition of entropy:

$$\mathrm{H}(V) = \sum_{v \in V} \mathrm{p}(v) \log_2 \frac{1}{\mathrm{p}(v)} \tag{3.4}$$

which is also the average code length required to express a vocabulary term, and can be thought of as a way of quantifying the user's effort to ascertain the vocabulary of a document set. Equation 3.4 can be interpreted as a weighted average where $\log_2 \frac{1}{\mathrm{p}(v)}$ is the length of the code of that vocabulary term and $\mathrm{p}(v)$ the proportion that the vocabulary term contributes to the overall average code length. In this model, when a term becomes a topic term, T_i, it is explicitly conveyed to the user, so that no bits are needed to express the particular term. This will also give the user some idea of what other vocabulary terms are present, so the code length of those terms should also be decreased. Which terms will a

user associate with a given topic term? We choose the terms by looking at how often a term, v_j, is associated with the topic term, which is modeled using the conditional probability, $\mathrm{p}(T_i|v_j)$. We assume that if the conditional probability is greater than the probability of v_j, the user's certainty about term v_j will increase. Otherwise it will remain the same. This yields the following formula for the average code length for a single topic term, T_i:

$$\mathrm{L}(V_{T_i}) = \sum_{v \in V} \mathrm{p}(v) \min \left[\log_2 \frac{1}{\mathrm{p}(v)}, \log_2 \frac{1}{\mathrm{p}(T_i|v)} \right]. \tag{3.5}$$

Equation 3.5 is the basis for the minimum average code length:

$$\mathrm{L}(V^*) = \arg\min_{T \in V} \left[\sum_{v \in V} \mathrm{p}(v) \min \left[\log_2 \frac{1}{\mathrm{p}(v)}, \log_2 \frac{1}{\mathrm{p}(t_1|v)}, \log_2 \frac{1}{\mathrm{p}(t_2|v)}, \ldots, \log_2 \frac{1}{\mathrm{p}(t_{|T|}|v)} \right] \right]. \tag{3.6}$$

In the implementation of the entropy model for finding topic terms, we use the same underlying language model of the document set. This language model is described in Section 3 of Lawrie *et al.*[14] and contains all the probabilities and conditional probabilities required for the calculation of the minimum average code length. The topic set is selected using a greedy approximation to Equation 3.6 by adding the topic that individually decreases the code length the most.

3.2.1 Example

In the following discussion, we compare hierarchies generated from language models where the conditional probabilities are based on a window size. For each hierarchy created using a language model, "$x = i$" or "$x = (i, j)$" follows the name of the technique. The $x = i$ refers to the size of the window used when creating the language model. These results attempt to establish the optimal language model for generating hierarchies as well as the best technique. Different depths in the hierarchy can be based on different window sizes, which is why $x = (i, j)$ is used for hierarchies with a depth of two.

Figure 3.1. A Minimum Average Code hierarchy created for TREC query 319: New Fuel Sources, where x=(5,2,1).

Min. Avg. Code, x=1		Min. Avg. Code, x=2		Min. Avg. Code, x=20		Min. Avg. Code, x=40	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
program	186	required	249	required	249	required	249
required	249	technolody	263	amendment	91	amendment	91
fuel	499	fuel	499	fuel	499	fuel	499
technology	263	amendment	91	Secretary	68	Secretary	68
amendment	91	program	186	technology	263	technology	263
States	221	nuclear	286	nuclear	286	nuclear	286
nuclear	286	States	221	research	201	research	201
services	135	services	135	House	69	tax	53
research	201	research	201	tax	53	inserting	35
energy	333	contained	173	program	186	program	186
House	69	Secretary	68	vehicle	126	power	323
cost	182	tax	53	services	135	vehicle	126
tax	53	energy	333	power	323	energy	333
system	214	Act	131	funds	84	services	135
based	221	cost	182	changes	189	changes	189

Table 3.1. Lists the topics terms and number of documents whose terms occur in the top level of the Minimum Average Code hierarchy using sliding windows of 1, 2, 20, and 40 for TREC query 319.

Figure 3.1 is one example of a hierarchy created using the Minimum Average Code (MAC) algorithm. MAC selects topic terms for a document set retrieved for TREC query 319 about New Fuel Sources. One of the most notable differences between this hierarchy and those created with the other techniques shown in Tables 1 and 2 of Lawrie *et al.* is that "fuel" is not the first topic term. This means that other terms are more consistently dependent on "required" and "amendment" than on fuel. In fact "fuel" is never the top ranked term when MAC is applied. However, the characteristics of the hierarchy in the figure are very similar to the characteristics of the hierarchy in Figure 2 of Lawrie *et al.* The menu under the topic "research" contains different topics as shown in Figure 3.1, but the sizes of the document clusters are very similar. The same can be said for the third level of the hierarchy.

Tables 3.1 and 3.2 show the terms chosen by the algorithm for a number of different language models. One of the most distinctive features of these sets of words is the similarity among them. There are only nine new terms in the seven lists introduced following MAC

Min. Avg. Code, x=50		Min. Avg. Code, x=60		Min. Avg. Code, x=80		Min. Avg. Code, x=100	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
required	249	required	249	required	249	required	249
amendment	91	amendment	91	amendment	91	amendment	91
fuel	499	fuel	499	fuel	499	fuel	499
Secretary	68	Secretary	68	technology	264	technology	264
technology	264	technology	264	Secretary	68	Secretary	68
nuclear	286	nuclear	286	nuclear	286	nuclear	286
research	201	research	201	research	201	energy	333
tax	53	program	186	program	186	program	186
program	186	tax	53	tax	53	tax	53
inserting	35	inserting	35	power	323	power	323
power	323	power	323	system	214	research	201
energy	333	energy	333	energy	333	Chairman	65
system	214	system	214	Act	131	system	214
vechicle	126	vechicle	126	Chairman	65	Act	131
services	135	Chairman	65	House	69	House	69

Table 3.2. Lists the topics terms and number of documents whose terms occur in for the top level of the Minimum Average Code length using sliding windows of 50, 60, 80, and 100 for TREC query 319.

x=1. In Tables 1 and 2 of Lawrie *et al*. there are twenty new terms in four lists introduced following DSP x=1, and there is much more variability in the smaller window sizes.

3.2.2 Initial Results

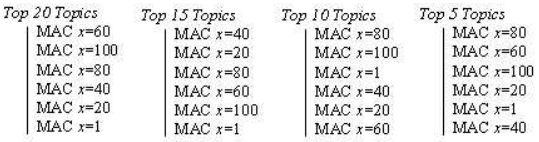

Figure 3.2. The ANOVA analysis of the hierarchy scores for Minimum Average Code with neighbors of 1, 20, 40, 60, 80, and 100. The bar to the left of the hierarchy types indicates where there is no significant differences among the types. In this figure there is no significant difference among the different terms chosen, although larger window sizes have slightly better mean scores.

We evaluate the top level of the hierarchies using two different quantitative techniques, which were also used in Lawrie *et al.*[14]. The first evaluation scores the hierarchy based on the number of documents read per relevant document in the document set. This evaluation

Figure 3.3. The ANOVA analysis for Subsumption, Lexical, TF.IDF, the Dominating Set with neighbors of 1, 50, and 100, and the Minimum Average Code with neighbors of 1, 50, and 100. 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. All three MAC examples appear in the second grouping and are not significantly different from Subsumption, TF.IDF, Lexical, and large window sizes of DSP.

assumes the user is interested in all relevant documents in the document set. The algorithm is explained in Section 5 of Lawrie and Croft[13], which is simplified here because we are only looking at the top level of the hierarchy. The second evaluation looks at the similarity between topic terms chosen, which we refer to as the overlap between two hierarchies.

Using hierarchy scores, we found no significant differences among the hierarchies created using the Minimum Average Code as shown in Figure 3.2. The means of hierarchies with larger window sizes were generally higher, but this can be explained by using Tables 3.1 and 3.2. The new terms that occur in hierarchies with larger window sizes occur in fewer documents. Since the size of the cluster plays an important role in the ability to choose clusters that add few non-relevant documents to the score, this result is not surprising. When comparing MAC to other techniques, only the Dominating Set where x=1 is significantly better, as shown in Figure 3.3. There are no significant differences in scores among subsumption, lexical, TF.IDF, MAC and DSP with large window sizes.

In Section 3.2.1 we noted that the terms selected were very similar across window sizes. In order to determine if this was true only for the particular query or a general phenomenon, we looked at the similarity between unary increments of window sizes: how similar MAC x=2 is to MAC x=1, how similar MAC x=3 is to MAC x=2, and so on up

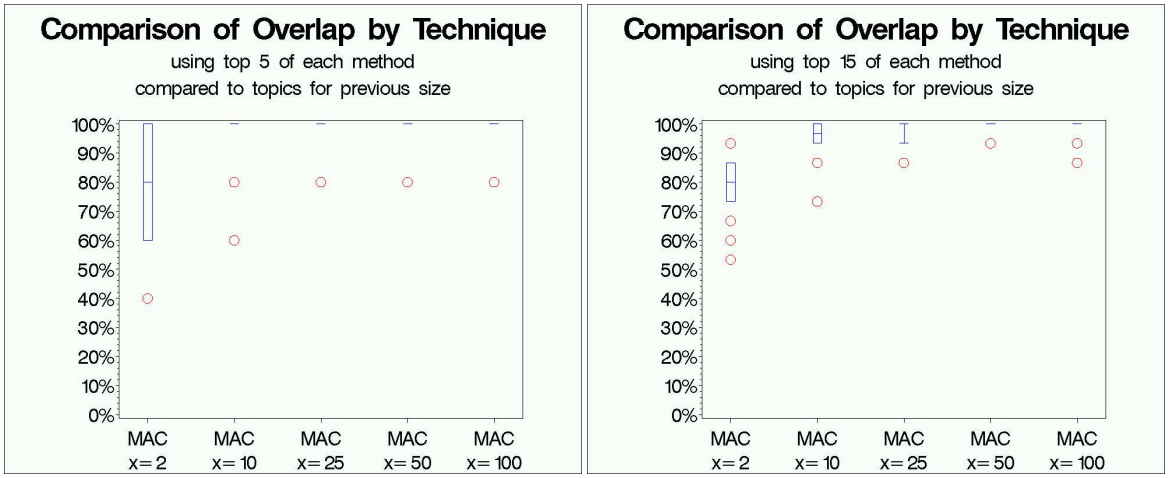

Figure 3.4. Illustrates the similarity of the top 15 and top 5 topics between the window size listed in the figure (e.g. x=2) and the window one smaller (e.g. x=1) for topics chosen by Minimum Average Code. For each method a box plot represents the similarities across all queries where the box is the middle 50% of the similarities. The whiskers go down to the 20th percentile and up to the 80th percentile. The circles represent points that fall outside the 20th to 80th percentile.

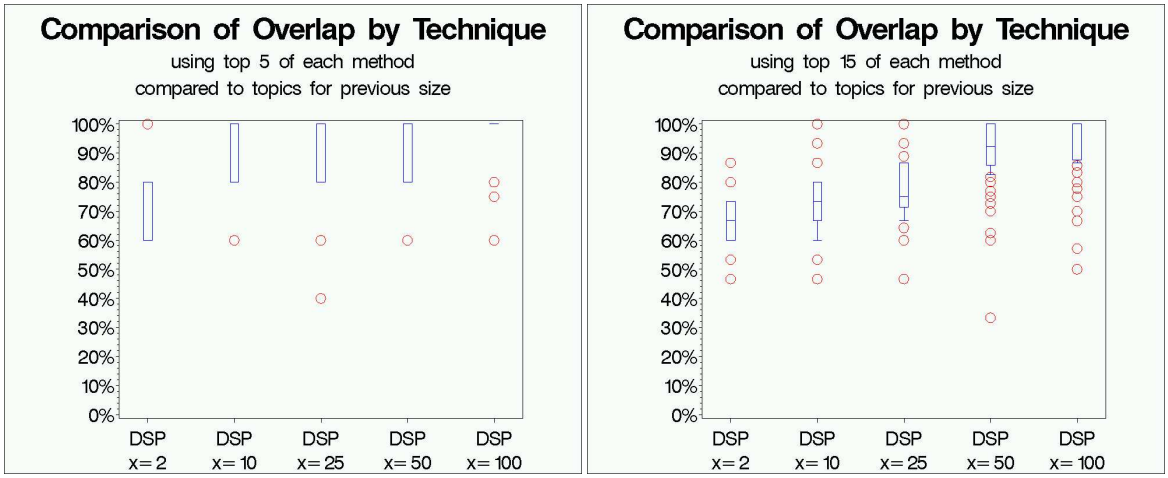

Figure 3.5. Illustrates the similarity of the top 15 and top 5 topics between the window size listed in the figure and the window one smaller for topics chosen by Dominating Set. In the figure showing 5 topic terms, there is a dot at 75% for DSP x=100. This is because for one query, DSP x=99 only selected 4 topics, and x=100 shared 3 of them giving a similarity of 75%.

to how similar MAC x=100 is to MAC x=99. We found that the similarity is very high as shown in Figure 3.4. The first five topics are almost always the same, except for small window sizes. When looking at the first fifteen topics, there is a little more variability, but as the window size increases, the similarity increases as well. In fact there is no window size for any topic where the similarity is less than 50%. In contrast, the Dominating Set has much less similarity between neighbors as is illustrated in Figure 3.5. Although the similarity increases as window size increases, it isn't as high as the corresponding window size for MAC. This is especially true when comparing the top 15 topics. In fact about 12% of the hierarchies have less than 50% of its topics in common with the preceding hierarchy.

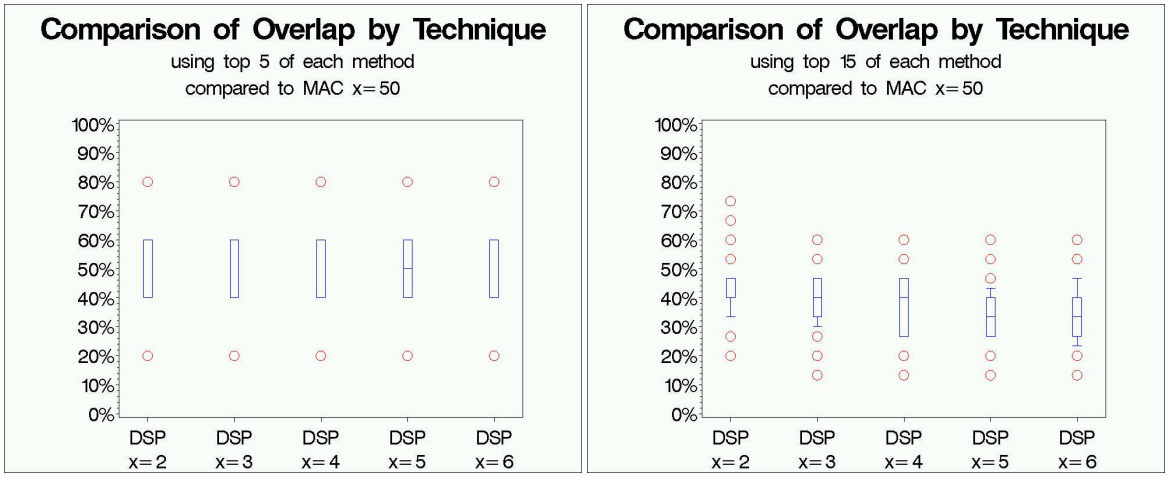

Figure 3.6. Illustrates the similarity of the top 15 and top 5 topics between MAC x=50 and terms chosen by the Dominating Set at small window sizes.

MAC was also compared to Subsumption, Lexical, TF.IDF, and DSP as shown in Figures 3.6 and 3.7. When comparing the top 5 topics of DSP with small windows to MAC x=50, on average about half of the terms are shared. When the top 15 topics are compared, the percent overlap decreases. For the other methods and DSP at larger window sizes, there is less similarity. In fact, occasionally none of the top five topic terms chosen by MAC x=50 are the same as the ones chosen by Subsumption, Lexical, and TF.IDF. MAC x=50 has less in common with these other techniques than DSP x=1 as shown in Figure 5 of Lawrie *et al.*, which means that MAC differs from previous techniques to a greater extent

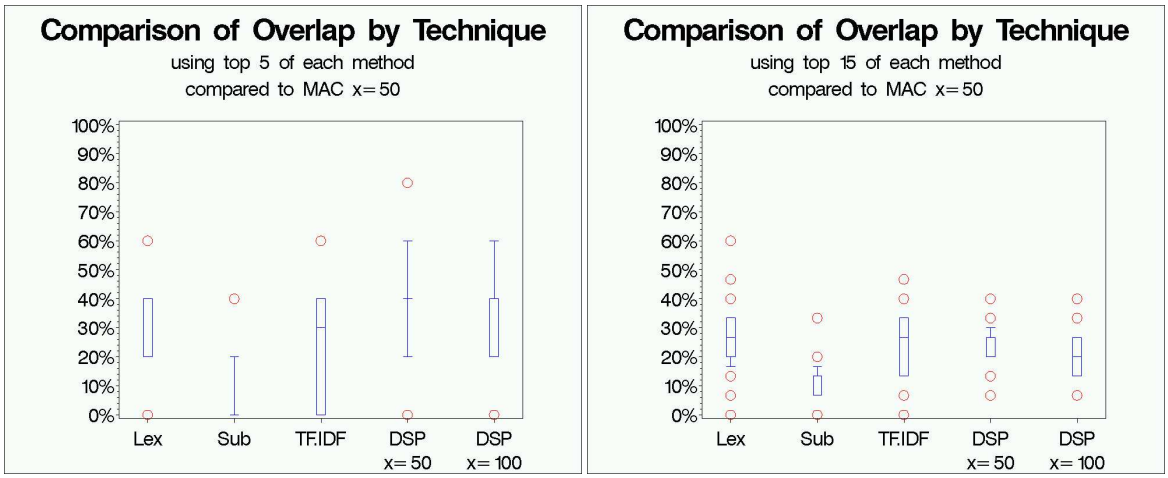

Figure 3.7. Illustrates the similarity of the top 15 and top 5 topics between MAC x=50 and Lexical, Subsumption, TF.IDF and terms chosen by the Dominating Set with large window sizes.

than DSP. The terms that MAC does have in common are just as likely to be found below the top five. This is very different from DSP where the average similarity increases when fewer topics are involved in the comparison.

3.3 Preliminary Evaluation of Hierarchies

We have applied the results of Lawrie *et al.*[14] to create hierarchies with a depth of two and used the same types of analysis as described above to evaluate the hierarchies. For the first level of the hierarchy, we chose window sizes of 5, 10, 15, and 20 since it seemed that the smaller window sizes performed better in our previous evaluation of DSP. These window sizes roughly correspond to a partial sentence to two or three sentences. For the second level of the hierarchy, we varied the window size between 1 and the parent's size.

Tables 3.3 and 3.4 show the subtopics for the term "research" for 8 different hierarchies. The smallest window size is (5, 2) where "research" was chosen for its ability to dominate vocabulary within a window size of 5 and the topic terms in the table for the ability to dominate vocabulary in a window size of 2. These hierarchies select very similar topic terms. Only 14 new terms are introduced in the seven lists following the first one and

DSP, x=(5,2)		DSP, x=(5,5)		DSP, x=(10,4)		DSP, x=(10,8)	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
technology	46	technology	46	technology	59	technology	59
basically	32	reactor	65	reactor	77	reactor	77
material	20	basically	32	Energy	70	Energy	70
Energy	46	Energy	46	material	41	funding	31
reactor	65	centers	34	systems	33	material	41
program	43	program	43	funding	31	project	53
institute	53	material	20	basically	34	fuel	89
funding	26	project	44	program	43	University	33
high	33	facility	36	fuel	89	Surveys	15
Nuclear	54	conducted	31	University	33	systems	33
centers	34	funding	26	facility	51	facility	51
systems	24	institute	53	high	43	centers	39
fuel	61	systems	24	project	53	methods	19
conducted	31	high	33	methods	19	program	43
project	44	area	27	production	41	institute	56

Table 3.3. Lists the topics terms and number of documents whose terms are subtopics of "research" for the Dominating Set hierarchies using sliding windows of (5,2), (5,5), (10,4), and (10,8). The first number in the pair is the window size that chose "research", and the second number is the window size that chose the terms listed in the table. Again these documents were retrieved for TREC query 319.

new terms are found at the bottom of the list, specifically in position 9 or higher. Another notable similarity is that all lists have ranked "technology" first, seven of the lists have ranked "reactor" second, and six of the lists have ranked "Energy" third. Not all of the subtopic lists are this closely related, but in general they do exhibit more similarities than the top level menus.

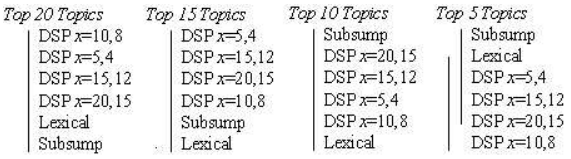

Figure 3.8. The ANOVA analysis for Subsumption, Lexical, TF.IDF, and the Dominating Set with neighbors of (5,4), (10,8), (15,12), and (20,15). The only significant difference occurs between Subsumption and DSP x=(10,8).

DSP, x=(5,2)		DSP, x=(5,5)		DSP, x=(10,4)		DSP, x=(10,8)	
terms	*# docs*	*terms*	*# docs*	*terms*	*# docs*	*terms*	*# docs*
technology	67	technology	67	technology	71	technology	71
reactor	87	reactor	87	reactor	91	reactor	91
Energy	75	Energy	75	Energy	85	Energy	85
fuel	101	systems	44	fuel	118	fuel	118
funding	34	fuel	101	material	55	material	55
material	51	centers	34	systems	48	program	50
systems	44	facility	54	program	50	facility	59
project	57	material	51	facility	59	systems	48
University	37	project	57	Power	74	funding	35
facility	54	Power	64	stated	49	study	43
Power	64	production	43	project	63	Power	74
high	50	credited	6	University	37	stated	49
operated	57	program	48	process	41	centers	42
National	39	area	40	water	41	basically	39
stated	46	Surveys	17	Nuclear	92	test	54

Table 3.4. Lists the topics terms and number of documents whose terms are subtopics of "research" for the Dominating Set hierarchies using sliding windows of (15,6), (15,12), (20,8), and (20,15).

Top 20 Topics	*Top 15 Topics*	*Top 10 Topics*	*Top 5 Topics*
DSP x=10,4	DSP x=15,6	Subsump	Subsump
DSP x=20,8	DSP x=5,2	DSP x=15,6	Lexical
DSP x=15,6	DSP x=20,8	DSP x=5,2	DSP x=5,2
DSP x=5,2	DSP x=10,4	DSP x=10,4	DSP x=10,4
Lexical	Subsump	DSP x=20,8	DSP x=15,6
Subsump	Lexical	Lexical	DSP x=20,8

Figure 3.9. The ANOVA analysis for Subsumption, Lexical, and the Dominating Set with neighbors of (5,2), (10,4), (15,6), and (20, 8). There are no significant differences found between any pairs of techniques.

The ANOVA results showed almost no significant differences between Subsumption, Lexical, and Dominating Set hierarchies with varying window sizes. The only significant difference is between Subsumption and DSP x=(10,8) when the top five topic terms are compared. The Subsumption and Lexical hierarchies are truncated at the second level. If this had not been done, Subsumption would have been significantly better than all the other hierarchies because the Subsumption hierarchies are quite deep and break the documents into very small clusters. The Lexical hierarchy's performance would not have changed as greatly because it has a maximum depth of three. Some of the ANOVA results are displayed

in Figures 3.8 and 3.9. From these results, the only thing we may conclude is that at a depth of two, DSP is not significantly worse than the prior two techniques for this task. Deeper hierarchies will reveal if DSP can perform better than Subsumption at choosing topic terms. Subsumption is shown to perform significantly better than lexical hierarchies in Lawrie and Croft.

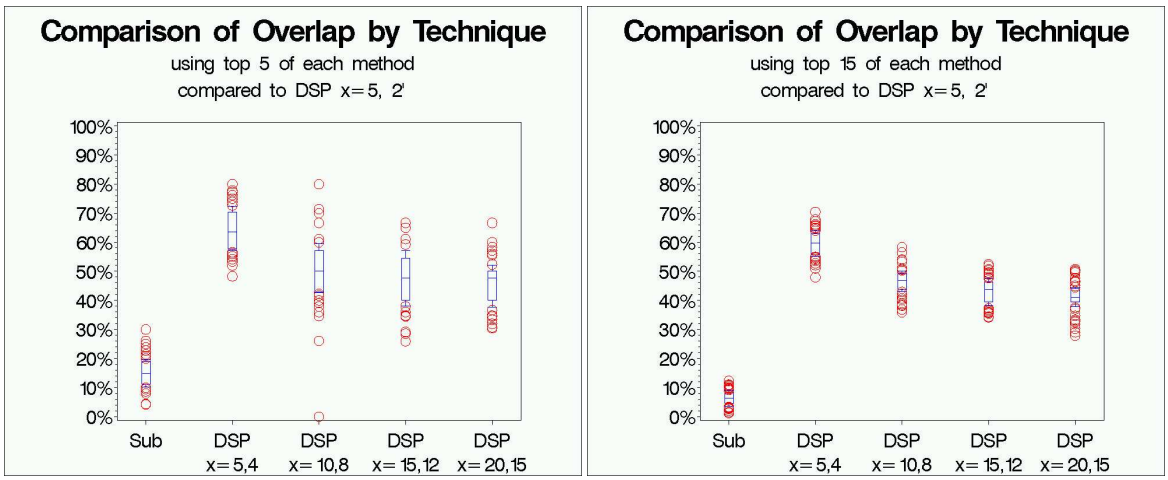

Figure 3.10. Illustrates the similarity of the top 5 and top 15 topics between DSP x=(5,2) and Subsumption, DSP x=(5,4), DSP x=(10,8), DSP x=(15,12) and DSP x=(20,15). DSP x=(5,4) has the greatest similarity because the first menus are exactly the same. The difference between Subsumtion and DSP x=(5,2) is more pronounced than when comparing the top levels.

The results of the similarity test magnify the differences observed in the comparisons of the single level hierarchies. Figure 3.10 and 3.11 show the overlap between DSP x=(5,2) and Subsumption, Lexical, and seven different DSP hierarchies. Both Lexical and Subsumtion have little similarity to DSP. When looking at similarity between different DSP, hierarchies that use the same language model for the first level exhibit more similarity. The comparison of hierarchies with five terms per level to fifteen reveals that the same terms tend to occur in the top 5 topics.

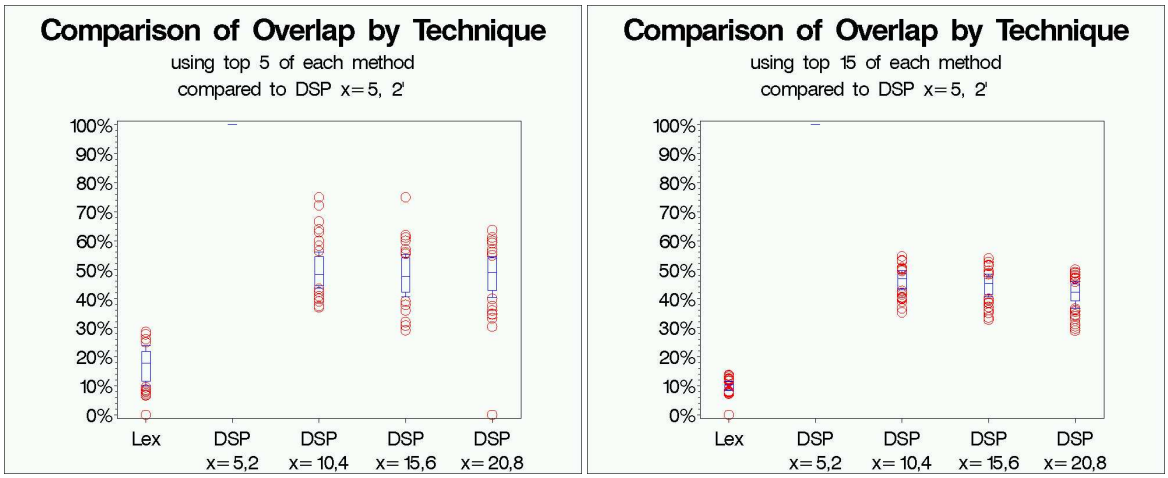

Figure 3.11. Illustrates the similarity of the top 5 and top 15 topics between DSP x=(5,2) and Lexical, DSP x=(5,2), DSP x=(10,4), DSP x=(15,6) and DSP x=(20,8). DSP x=(5,2) is always exactly the same since it is a comparison with itself. The difference between Lexical and DSP x=(5,2) is about the same as the difference between Subsumption and DSP x=(5,2). This is a considerably different from the comparison of the first level where the median similarity was almost 50%.

3.4 Summary

Thus far we have developed two formal models for selecting topic terms and done a preliminary evaluation using a scoring metric and investigating overlap. The scoring metric presented in this proposal shows that there is no significant difference between the Minimum Average Code and the other techniques at the first level of the hierarchy. It also shows that there are no significant differences for hierarchies of two levels deep. We believe that the shortcomings observed do not point to failings in the models but instead to failings in the evaluation metric. Although scoring the hierarchy makes sense for fully developed hierarchies, it only reveals whether a technique is particularly bad when applied to truncated sections of the hierarchy.

The reason that Entropy Model merits further analysis despite its mediocre performance is that the terms it selects are not as dependent on the particular language model when comparing it to the Dominating Set. Also, before discounting it, an analysis of the contents of the hierarchy needs to be done because it is choosing different terms than other methods. However, this model does not make any guarantees about the coverage of the topics. It will

have to be determined if there exists a lack of coverage. Finally, limiting topics to nouns and noun phrases will be investigated since verbs and adjectives violate the assumption of the model that people can use associations to figure out what other vocabulary terms are present in the document set.

The Dominating Model needs further analysis for the same reasons that the Entropy Model does. A true test will be for a hierarchy that is four or five levels deep to be compared to the full subsumption hierarchy. If significant differences are found, there will be much more substantial reasons to believe that Dominating Model is a good technique. However, it may be that the hierarchies are indistinguishable using the scoring metric and other evaluations will reveal the optimal technique.

The remainder of the research will focus on developing more robust evaluations and verifying that the models used to select topic terms are correct.

CHAPTER 4

INTENDED RESEARCH

In order for this research to be successful we need to define an optimal hierarchy. This will enable us to find comparison measures that truly distinguish between a useful hierarchy and one that is not. Part of this definition will encompass characteristics of the hierarchy such as its depth, the size of document clusters at the leaves, and the change in the size of document clusters as one explores deeper in the hierarchy. Another part of the definition will need to address the content of the hierarchies. An optimal hierarchy should outline the topics of the document set. A third part of the definition must address the issue of usability by determining whether people can use the hierarchies to learn the topics of the document set and find interesting groups of documents.

From the initial results, we can select characteristics that should be present in a definition of an optimal hierarchy. One characteristic is that each level's cluster sizes should be roughly the same; however, the top levels may have a greater variance than lower levels. The hierarchy will most likely be between three and five levels deep for a set of five hundred documents. The depth will be governed by the goal size for cluster leaves. Currently, we believe leaf-clusters should have somewhere between 3 and 10 documents.

The second and third issues will be developed with future experimentation. Testing the ability of a hierarchy to cover different topics can be done through simulations. We will combine documents that are relevant to a diverse group of queries so that we can predict the topics the hierarchy should discover by testing for the presence of query terms in the hierarchy. We will also need to find out if the terms in the hierarchy actually summarize the document set. An experiment to determine this could involve users highlighting terms from

the document that they would include in a summary. User terms would then be compared to the terms in the hierarchy.

To address the issue of usability, we will develop a demo environment where users can interact with the hierarchy. We will then test their ability to do retrieval-oriented tasks. Since all of our current document sets are retrieved for a query, some of the tasks will be oriented toward finding non-relevant documents, while others will focus on relevant documents. Tasks may also be defined that make use of a known document. We may also define tasks that can be accomplished without use of a hierarchy as a control.

4.1 Detailed Plan

- Optimize and debug code for generating the language models and finding topic terms. Ideally this optimization would allow the creation of a hierarchy with a depth of 5 in less than 2 minutes. Such a hierarchy with 20 terms at each level requires roughly 3 million language models, so this might be a bit ambitious. (1 month)

- Continue to build deeper hierarchies and test them with existing techniques. The characteristics of the hierarchy will be analyzed to find the parameters for optimal hierarchies. These parameters may be dependent on the size of the document set, and we will need to explore this possibility. (1 month)

- Explore building hierarchies with nouns and noun phrases. Verbs and adjectives tend to make disappointing topics, since one has little notion of what types of terms will be associated with it. We will also experiment with hierarchies made entirely from phrases, since phrases tend to convey more information than single words. (3 weeks)

- Complete a failure analysis. Determine what types of conditions yield poor hierarchies. (1 month)

- Investigate how different size initial document sets effect the hierarchy ,and specifically what the effect of clustering is on the hierarchy. Find out if the models compen-

sate for the lack of a clustering algorithm, which we found to be quite helpful when creating lexical and subsumption hierarchies. If the model does not compensate, determine if a clustering step should be added. (2 weeks)

- Develop an evaluation that looks at how well topics are covered. This may be done by creating document sets made up of relevant documents from a number of different queries and looking at how well the hierarchy groups the documents. The topics should be judged to determine if the terms describe the documents. The easiest way to do that is to look for topic words that are part of the query. (2 weeks)

- Continually develop and test the idea of what a globally optimal hierarchy is.

- Develop an interface that can be used in a demo environment for user studies. (2 months)

- Investigate how one would use the system in a real setting based on other work. (2 weeks)

- Do a user study that incorporates the demo and measures how well users can perform retrieval oriented tasks using a ranked list verse using the hierarchy. (3 weeks)

Preparation and writing of the dissertation itself will take approximately four months. The entire project is expected to talk 12 months to complete. Including other degree obligations, this should allow a thesis defense to be scheduled in August of 2002.

CHAPTER 5

CONCLUSION

This proposal outlines the development of an approach to automatically generate hierarchical summaries. There are several steps that contribute to the creation of the summary: building the language model, selecting topic terms, and testing the hierarchy. The contributions of this thesis will include:

- a definition of a globally optimal hierarchy in terms of both physical characteristics and the information users gain from the hierarchy,
- the first formal framework for selecting topic terms, and
- the development of evaluation measures for hierarchical summarization based on large text collections.

The foundation for this work has already been laid. We have defined two new formal approaches to selecting topic terms and evaluation metrics that help us decide which hierarchies are good. In the coming months we will develop these ideas further.

BIBLIOGRAPHY

[1] P. Anick. *Automatic construction of faceted terminological feedback for context-based information retrieval*. PhD thesis, Brandeis University, 1999.

[2] A. Berger and V. Mittal. Ocelot: 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] W. Chuang and J. Yang. Extracting sentence segments for text summarization: a machine learning approach. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 152–159, 2000.

[5] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 318–329, Copenhagen Denmark, 1992.

[6] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 121–128, Berkeley, California, 8 1999.

[7] 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.

[8] M. Hearst and C. Karadi. Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 246–255, Philadelphia, PA, 1997.

[9] T. Hofmann. The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *Proceedings of International Joint Conference on Artificial Intelligence 1999*, pages 682–687, 1999.

[10] F. Korn and B. Shneiderman. Navigating terminology hierarchies to access a digital library of medical images. Technical Report HCIL-TR-94-03, University of Maryland, 1995.

[11] J. Kupiec, J. Pederson, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, 1995.

[12] J. Lamping and R. Rao. The hyperbook brower: A focus + context technique for visualizing large hierarchies. In S. Card, J. MacKinlay, and B. Shneiderman, editors, *Reading in Information Visualization: Using Vision to Think*, pages 382–407. Morgan Kaufman Publishers, Inc, 1999.

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

[14] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, page ?, 2001.

[15] N. Light. Main page. www.northernlight.com, 1997.

[16] 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.

[17] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

[18] I. Mani and E. Bloedorn. Multi-documemt summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 622–628, 1997.

[19] K. McKeown, J. Klavans, V. Hatzivzssiloglou, R. Barzilay, and E. Eskin. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 74–82, 1995.

[20] 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.

[21] V. Mittal, M. Kantrowitz, J. Goldstein, and J. Carbonell. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 467–473, 1999.

[22] S. Pollitt. Interactive information retrieval based on faceted classification using views. In *Proceedings of the 6th International Study Conference on Classification*, 1997.

[23] 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.

[24] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[25] P. Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24(5):577–587, 1988.

[26] 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.

[27] Yahoo! Main page. www.yahoo.com, 1995.

[28] Y. Yand, T. Pierce, and J. Carbonell. A study on retrospective and on-line event dection. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, 1998.