Generating Hierarchical Summaries for Web Searches*

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

ABSTRACT

Hierarchies provide a means of organizing, summarizing and accessing information. We describe a method for automatically generating hierarchies from small collections of text, and then apply this technique to summarizing the documents retrieved by a search engine. We show that these hierarchies provide better access to the documents than a simple ranked list and that the terms in the hierarchy are better summaries of the documents than the top TF.IDF weighted terms. In addition, we discuss the formal framework of the technique and how the technique has been used with news databases and TREC collections.

General Terms

Language Models, Web IR, Summarization

Keywords

topic hierarchies, web search results, evaluating hierarchies

1. INTRODUCTION

A topic hierarchy is a description of a body of text which both summaries the text and provides a method of navigating through it. The popularity of hierarchies as a method of organization indicates that this type of summary is relatively easy to understand. For example, the Yahoo hierarchies[21] and MeSH headings[11] have been used for many years. Hierarchies provide two kinds of information. One is a method of navigating to particular sub-parts of a collection that contain information of interest to a user. The second is a summarization of the collection. As a hierarchical summary, topic terms appear as the entries in the hierarchy and should describe the documents that are found under them. Thus a hierarchy is both a sparse summary and a tool for navigating a collection.

Although hierarchies are useful tools, they are very time consuming and expensive for people to build. An alternative to creating hierarchies by hand is to develop techniques that create them automatically. In this paper we present a formal framework for automatically building hierarchies for small collections of text. We envision building hierarchies for personal collections of electronic documents, e-mails, documents retrieved for a query, and document summaries generated by search engines. Here, we focus on the latter types of hierarchies, used in conjunction with retrieval.

The hierarchies have been designed with two main attributes in mind. First, they are intended to be predictive summaries. This means that a user browsing the hierarchy should be able to predict the type of documents that will be found at any point in the hierarchy. Second, they are intended to provide a method of navigation, which in the context of retrieval is used in conjunction with a ranked list. This enables users to have access to documents that otherwise would have remained hidden because of their location in the ranking.

In order to construct the hierarchies, we build statistical models of language to identify topic terms in a document set. These statistical models can be built recursively to identify subtopics of a main topic, thus creating a hierarchy that summarizes the collection. Documents are attached to the hierarchy if they include the topic terms, thereby creating a means of navigating the collection.

When searching for web documents, this type of organization can give the user an alternative to the organization provided by a ranked list. Specifically, a hierarchy can be built from the summaries automatically generated by a search engine to describe web documents for the top 200, 500, or even 1000 documents retrieved for a specific query. Unlike a ranked list where a user is unlikely to find relevant documents not ranked within the top 50, a hierarchy can quickly group the 153^{rd}, 217^{th}, and 568^{th} document under a topic, for example. If the topic seems relevant, the user will find these documents that otherwise would have remained undiscovered.

The hierarchical summary has an advantage when compared to clustering algorithms that are used to organize a retrieved set (e.g. Scatter/Gather[6]). In the case of the hierarchy, a document belongs to a topic if the document includes the topic term. A user might still disagree about the inclusion of a document under a particular topic, but she will be able to understand why the document appears there. In addition, there will never be topics that appear in a particular part of the hierarchy that seem completely unrelated to parent topics. The ease with which people can understand the topic hierarchy will help them trust the results that the hierarchy gives them. On the other hand, clustering algorithms tend to group documents that appear to be similar based on several features. Terms that are chosen to describe the cluster never completely cover all aspects of a cluster. For example, in clusters about the Gulf War, oil sales and stock markets, and East and West Germany, there are also documents about Pakistan, Trinidad, South Africa, and Liberia[6], which do not seem to have much to do with the stated topics.

In the following section, we present examples of hierarchies created from Google's automatically generated snippets of documents. In Section 3 we describe the formal framework used to generate topic hierarchies. In Section 4 we discuss the implementation of the hierarchies and parameters used to improve them. In Section 5 we evaluate the hierarchies generated from web search, as well as the hierarchies generated using other text sources. In Section 6 we

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 0-89791-88-6/97/05.

discuss related work, and finally we conclude with future work in Section 7.

2. EXAMPLES

Search engines have become very good at ranking relevant webpages for certain types of queries. Unlike 5 to 10 years ago, when a search for a company web page would often rank personal home pages above a company's own page, today the company page is likely to be the top ranked page. However, there are other types of queries where web search engines do not perform nearly as well. This occurs frequently when the query is a search for general information on a topic. Such topical searches are where hierarchies can help a user the most. The hierarchy provides an alternative to browsing a ranked list, and can be a more effective organization of the documents that were retrieved, thus allowing users to find the relevant information.

Topic hierarchies are very good at assisting a user in several different ways. First, they make it easy to view portions of a ranked list that would not otherwise be seen, and thereby find relevant documents that are not highly ranked. Second, a hierarchy more effectively indicates when the retrieval does not go well. A user may look at several pages of results from a ranked list before finally concluding that there is nothing relevant, while a hierarchy allows the user to determine this almost immediately. Third, the hierarchy can show alternative contexts where a term is used, and bring alternative uses of a term to the user's attention without requiring her to reformulate her query.

To demonstrate these claims, we have selected a few hierarchies from TREC topics[20] built using our system. These hierarchies were generated by first sending a short query (the TREC "title" field) to the Google search engine[8]. Then we used the titles and snippets that Google returned as the text to generate the hierarchies.

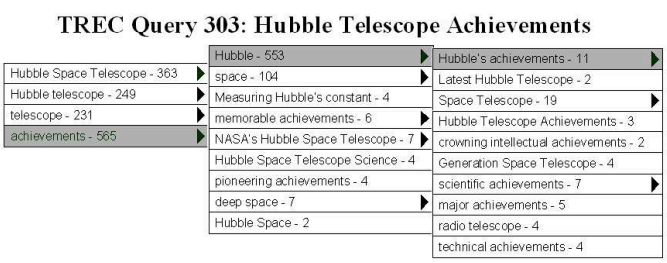
2.1 Hubble Telescope Achievements

Figure 1 shows the hierarchy generated from snippets retrieved for the query: "Hubble Telescope Achievements". The top part of the figure is a portion of the hierarchy that the user would see when interacting with our system. The ranked list that follows is the portion of the ranked list that would be seen by navigating to "achievements→Hubble→Hubble's achievements" and selecting the topic "Hubble's achievement". The order of the documents is the same order that would be seen in the ranked list and the rank numbers correspond to the ranks assigned by the search engine.

The topic "achievements→Hubble→Hubble's achievements" includes eleven documents, although only the first six appear in the figure. Of the eleven documents, three of them are about Hubble telescope achievements including the ones shown in Figure 1 that rank at 100 and 140. In fact, the HubbleSite, which was ranked at 140, is a site devoted to the telescope and includes a fairly in-depth discussion of its achievements. Because of the site's low rank, it is unlikely a user would have found it without using a tool such as the hierarchy.

2.2 Abuses of E-mail

The hierarchy created for the query "Abuses of E-mail" appears in Figure 2, which demonstrates the value of the hierarchy in the case where the quality of retrieval is not good. Of the six high level topics, only two show any promise as avenues to finding documents about E-mail abuses. The fact that only 274 of the 811 documents retrieved contain the word "E-mail" also supports this claim. The two promising avenues are expanded in Figure 2. The topic "unsolicited commercial e-mail" found under "abuses" summarizes the predominate type of e-mail abuse that websites discuss, also known as spamming. It would have been very difficult for a user to draw these conclusions by looking at the top of the ranked list.



achievements→Hubble→Hubble's achievements

Rank 19: *Books with Pictures From Space (Science U)*
... of Our Cosmos, by Simon Goodwin A gallery of the most signifi cant photographs taken by the **Hubble** telescope explains what **Hubble's achievements** can ...

Rank 34: *Amazon. COM: buying info: **Hubble's** Universe: A Portrait of Our ...*
... Ingram A gallery of the most signifi cant photographs of space as taken by the **Hubble** telescope explains what **Hubble's achievements** can tell us about the ...

Rank 63: *ESA Portal - Press Releases - HST's 10th anniversary, ESA and ...*
... A public conference will take place in the afternoon to celebrate **Hubble's achievements** midway through its ... Notes for editors. The **Hubble** Space Telescope ...

Rank 100: *FirstScience.com - The **Hubble** Decade*
... astronauts' fi rst view of the Earth from the Moon - and the **Hubble** Space Telescope's ... View from the top. On the scientifi c front, **Hubble's achievements** ...

Rank 111: *The Hindu : Discoverer of expanding universe*
... **Hubble's achievements** were recognised during his lifetime by the many honours conferred upon him In 1948 he was elected an ... 'The **Hubble** Space Telescope ...

Rank 140: *HubbleSite — Science*
... farther and sharper than any optical/ultraviolet/infrared telescope ... very specifi c goal (like the Cosmic Background Explorer), **Hubble's achievements** ...

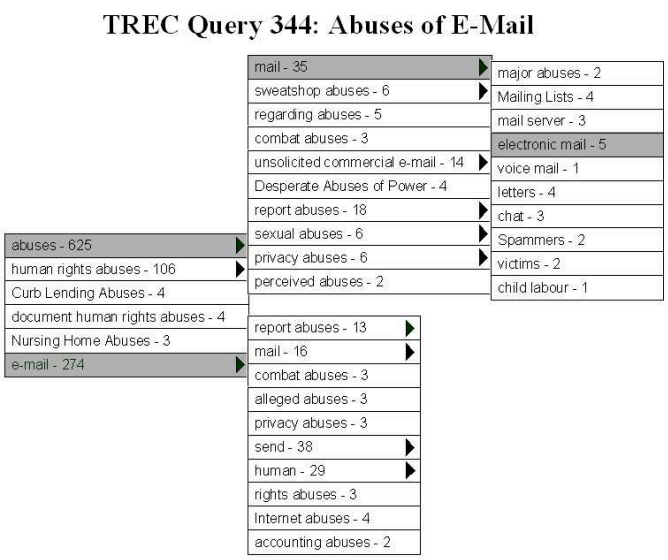
Figure 1: The figure shows a portion of the hierarchy created for the TREC Topic 303: Hubble Telescope Achievements, created from snippets of documents retrieved by Google. The portion of the ranked list displayed corresponds to documents that contain the terms "achievements", "Hubble", and "Hubble's achievements". The snippets indicated that all of the documents have to do with the Hubble Telescope, but one of the best sites for finding out about the Hubble Telescope's achievements is the *HubbleSite — Science*, ranked 140^{th} by Google.

2.3 Airport Security

Figure 3 shows the hierarchy created for the query "Airport Security" using snippets. Most of the 853 documents retrieved for this query have to do with the problem of securing people while they travel on airplanes, which is the predominant meaning of airport security. However, a user searching for information on Apple's AirPort security using the query "Airport Security" would most likely not be persistent enough to find the documents pertaining to this topic with an ordinary ranked list. Because the hierarchy accounts for these minority topics, this small group of documents is easy to locate under the topic "Apple Boost AirPort Security". Most likely the user would still need to reformulate the query to better fulfill her information need, but the first search would be made more fruitful by using the hierarchy.

3. FRAMEWORK FOR HIERARCHIES

The main challenge to creating topic hierarchies is selecting the correct terms that will accurately describe the document set to the user. We propose that the best topical summarization terms are those which are both about the topic and predictive of other terms. We model the set of topical summary terms T as the maximization of the joint probability of topicality and predictiveness, given by



abuses→mail→electronic mail

Rank 14: *Hearing Witness: Jerry Cerasale: Spamming: The E-Mail You Want To ...*
... signifi cant developments that are beginning to effectively combat **abuses** of **electronic mail**. These developments include the termination of service by e-mail ...

Rank 85: *RedIRIS - **Abuse** of **electronic mail** services.*
... for both e-mail users and administrators, with the purpose of increasing their awareness of some ways in which these services are being **abused**. These **abuses** ...

Rank 431: *NTPG : Appendix E : Acceptable Use Policies*
... District will make every effort to protect students and teachers from any misuses or **abuses** as a ... **Electronic mail** (e-mail) is not guaranteed to be private. ...

Rank 529: *http://www.eff.org/CAF/statements/ubc.ca.appendix-g-h-i*
... department name] by phone ([campus local]) or by **electronic mail** ([e-mail ... of these facilities and associated penalties, and the procedures for reporting **abuses** ...

Rank 553: *Thank You!*
... Winner will be sent their prize-winning notifi cation via **electronic mail** (e-mail ... If disqualifi ed for any of the above **abuses**, Sponsor and RealTime Media, Inc. ...

Figure 2: The figure shows a portion of the hierarchy created for the TREC Topic 344: Abuses of E-mail. The portion of the ranked list displayed corresponds to documents that contain the terms "abuses", "mail", and "electronic mail". This hierarchy demonstrates how easy it is to tell when retrieval is poor. With topics about human rights, finance, and nursing home abuses, it is not surprising that relevant documents are difficult to find.

the Term Selection Formula:

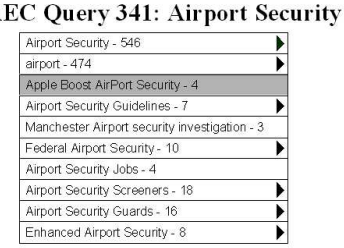
$$\arg\max_{T} \mathcal{P}(A_{\mathcal{T}}, B)$$

where $A_{\mathcal{T}}$ refers to topicality with respect to topic $\mathcal{T}$ and B refers to predictiveness.

For topicality, consider the set of terms that make up the vocabulary of a document. A portion of this set of terms is topical with respect to a given topic $\mathcal{T}$. Topical terms are the content terms used to express the information about a given topic. So given the set of vocabulary terms V, $A_{\mathcal{T}}$ is the function $A_{\mathcal{T}} : V \rightarrow \{0, 1\}$ where:

$$A_{\mathcal{T}}(w) = \begin{cases} 1 & \text{if the word } w \text{ is in } T \text{ (the set of topical terms)} \\ 0 & \text{otherwise} \end{cases}$$

The second quality is predictiveness. Predictive terms are those whose occurrence is a precondition for many other terms. Previous work on topic hierarchies[9, 10, 18] has shown that this is an important aspect of topic terms, because these are the terms that are frequently used to discuss the topic at different levels of generality.



Apple Boost AirPort Security

Rank 250:***Apple Boost AirPort Security**, Features*
... ISP November 13, 2001 **Apple Boost AirPort Security**, Features By Jim Wagner Software programmers at Apple (NASDAQ:AAPL) released the latest iteration of its ...

Rank 489:***Apple Boost AirPort Security**, Features*
... Wireless November 13, 2001 **Apple Boost AirPort Security**, Features By Jim Wagner Software programmers at Apple (NASDAQ:AAPL) released the latest iteration of ...

Rank 575: ***Apple Boost AirPort Security**, Features*
... **Apple Boost AirPort Security**, Features November 13, 2001 Software programmers at Apple (NASDAQ:AAPL) released the latest iteration of its wireless networking ...

Rank 596:***Apple Boost AirPort Security**, Features* ... siliconvalley.internet.com November 13, 2001 **Apple Boost AirPort Security**, Features By Jim Wagner Software programmers at Apple (NASDAQ:AAPL) released the ...

Figure 3: The figure shows a portion of the hierarchy created for the TREC Topic 341: Airport Security. The portion of the ranked list displayed corresponds to documents that contain the term "Apple Boost AirPort Security". This hierarchy demonstrates how a particular aspect of airport security (i.e. the apple networking type) is easy to find using the hierarchy.

So similarly to $A_{\mathcal{T}}$, $B : V \rightarrow 0, 1$ where:

$$B(w) = \begin{cases} 1 & \text{if the word } w \text{ is in } P \text{ (the set of predictive terms)} \\ 0 & \text{otherwise} \end{cases}$$

By combining these two properties, one finds a set of words that will maximize a user's understanding of the information contained in the documents. Topical terms are the content-bearing terms with respect to a particular topic. These are not necessarily the most frequent terms, since they may only be mentioned once in a document; however, the information conveyed to the reader by content-bearing terms is crucial to the reader's understanding of the document. Predictive terms are those which occur with a distinct set of vocabulary and without which it would be highly unlikely that other terms would occur. Included in this set of terms are most stopwords, which due to their frequent occurrence in a document make most terms dependent on them. Also in the set of predictive terms are general terms related to a topic. For example, in a retrieved set about Endangered Mammals, "endangered species" is likely to be a predictive term since it occurs frequently in the document set.

Since there are some terms that are predictive and not topical (i.e. stopwords), and other terms that are topical but not predictive (single occurrences of content-bearing terms), we assume the two probabilities are independent, and so $\mathcal{P}(A_{\mathcal{T}}, B) = \mathcal{P}(A_{\mathcal{T}})\mathcal{P}(B)$. We use statistical language models[12] to estimate $\mathcal{P}(A_{\mathcal{T}})$, topicality, and $\mathcal{P}(B)$, predictiveness. In order to estimate topicality, a unigram language model is computed, and then the Kullback-Leibler divergence contribution of each term is calculated. In order to estimate predictiveness, a language model is necessary to show how terms relate to each other. In the following subsections we describe our estimation techniques.

3.1 Estimating Topicality

When creating the hierarchies, it is necessary to determine the likelihood that a particular term is in fact a topic term. This means that the probability of a term being a topic term needs to be estimated. It has been shown in other work[4] that terms ranked highly by their contribution to the Kullback-Leibler divergence score are

likely to be about the topic of the document, while terms that are not ranked highly are less likely to be about the topic. This is why we choose to estimate topicality using KL divergence[3].

In this context, KL divergence is a measure of relative entropy between the language model of the text used to create the hierarchy, H, and the language model of general English, GE, given by

$$\text{KL contribution}(w) = \mathcal{P}_H(w) \log_2 \frac{\mathcal{P}_H(w)}{\mathcal{P}_{GE}(w)}.$$

A particular term w has a contribution of zero when $\mathcal{P}_H(w) = \mathcal{P}_{GE}(w)$. It has a positive contribution when $\mathcal{P}_H(w) > \mathcal{P}_{GE}(w)$, and a negative contribution when $\mathcal{P}_H(w) < \mathcal{P}_{GE}(w)$. This means that the most topical terms will be the terms where $\mathcal{P}_H(w) \gg \mathcal{P}_{GE}(w)$.

Now, $\mathcal{P}_H(w)$ and $\mathcal{P}_{GE}(w)$ must be estimated. The most straightforward approach is to estimate the unigram language model for the hierarchy where

$$\mathcal{P}_H(w) = \frac{\#\text{occurrences}(w)}{\#\text{terms in hierarchy text}}.$$

The language model of general English can be estimated in a similar way by using the frequency of terms in a suitably large collection.

When we use this methodology for estimating topicality for building hierarchies from documents in TREC volumes 4 and 5, we find some highly ranked terms that are clearly not topic terms, such as the term "src". When investigating the cause of this problem, we find that "src" is a fairly common term in the Congressional Records, a sub-collection of TREC volumes 4 and 5. However, relative to the other sub-collections, the Congressional Record is very small. In many other collections, the term "src" never occurs. In the end, $\mathcal{P}_{GE}$("src") will be very small. This means that if a hierarchy happens to have a number of documents from the Congressional Records, $\mathcal{P}_H(\text{"src"}) \gg \mathcal{P}_{GE}(\text{"src"})$ even if "src" only occurs an average number of times for the Congressional Record.

There are two ways of addressing this problem. One method is to bias $\mathcal{P}_H(w)$ towards the query. This method can only be used when the hierarchy is being constructed for a group of documents retrieved for a query. Given a query Q, we estimate the probability of w using the formulation in Cronen-Townsend and Croft[4]:

$$\mathcal{P}_{H_Q}(w) = \mathcal{P}(w|Q) = \sum_{D \in H} \mathcal{P}(w|D)\mathcal{P}(D|Q),$$

where H refers to the set of documents used to create the hierarchy and

$$\mathcal{P}(D|Q) = \prod_{q \in Q} \mathcal{P}(q|D).$$

This method demotes terms such as "src" because they are not terms related to the query. A query bias provides a better estimate of topicality; however, it does not provide a universal approach – there are instances where a hierarchy is useful, but a query is not present.

The second method relies on the knowledge of a document's sub-collection or document type. This means that given a set of documents H, $H = \{H_1, H_2, ...H_n\}$, where there are n different sub-collections from which the documents originated. And given H_i, there exists a language model SC_i that models the type of language used by a particular sub-collection. This means that the KL divergence contribution of a term can be calculated for each sub-collection separately, and then combined to get a single KL divergence contribution in the following manner:

$$\text{KL contribution}(w) = \sum_{H_i \in H} \mathcal{P}(H_i)\mathcal{P}_{H_i}(w) \log_2 \frac{\mathcal{P}_{H_i}(w)}{\mathcal{P}_{SC_i}(w)},$$

where

$$\begin{aligned} \mathcal{P}(H_i) &= \frac{|H_i|}{|H|}, \\ \mathcal{P}_{Hi}(w) &= \frac{\#\text{occurrences}_{H_i}(w)}{\#\text{terms in } H_i}, \text{ and} \\ \mathcal{P}_{SCi}(w) &= \frac{\#\text{occurrences}_{SC_i}(w)}{\#\text{terms in } SC_i}. \end{aligned}$$

This method also gives a better estimation of a term's topicality, but does not bias the hierarchy with respect to the query.

The background model is usually estimated as a probability distribution of words in "general English" calculated from any suitably large collection of text. In this application, however, it is necessary to model the same type of text for which the hierarchy is generated. For web hierarchies, we used a background model estimated from half a million Google snippets collected using randomly selected words in a dictionary to have as true an estimation of the type of language used on the web as possible.

3.2 Estimating Predictiveness

Since the hierarchical summary enables users to predict the kind of documents she is likely to see under a given topic, the terms chosen to be part of the hierarchy should be predictive of the terms that are not seen by the user. The algorithms for estimating predictiveness appear in Lawrie and Croft[10].

First, a co-occurrence language model is constructed. A given entry in the model contains $\mathcal{P}(t|v)$ where t is a possible topic term and v is another term in the vocabulary and is estimated using the following formulation:

$$\mathcal{P}(t|v) = \frac{\text{windows}_x(t \cap v)}{\#\text{occurrences}(v)},$$

where $\text{windows}_x(t \cap v)$ refers to the number of windows of size x where t and v co-occur. The size of windows is one of the parameters set when generating a hierarchy. The effects of different settings of the parameter are studied in Lawrie and Croft[10]. The window size should capture the range that a particular terms occurrence is dependent on the occurrence of another term.

After computing the language model, the probability of predictiveness can be estimated in the following manner:

$$\mathcal{P}(B) = \frac{1}{|V_t|} \sum_{v \in V_t} \mathcal{P}(t|v),$$

where V_t are the possible topics for a given level of the hierarchy. This means that the terms that have a high conditional probability with many terms in the set V_t are the most predictive terms.

By restricting the language model and the set V_t for different levels on the hierarchy, we are able to identify subtopics of topics, and thus generate the hierarchy recursively. Initially, the language model is created using all the text that the hierarchy is summarizing. The top level terms are selected based on their topicality and predictiveness. However, when finding subtopics of a topic, the language model is recalculated over the text that the parent topic term originally predicted, which is determined by the window size. This biases the model to the type of text that occurs near the topic term. Again, the most topical and predictive terms are chosen as subtopics, but because of the bias of the language model and the

vocabulary set, the terms are only topics in the context of the parent topic.

4. HIERARCHY IMPLEMENTATION AND PARAMETERS

In this section we discuss the implementation of the hierarchies. This includes both a discussion of how the system works in Subsection 4.1 and of the different parameter setting that influence the construction in Subsection 4.2.

4.1 Implementation

The hierarchies are implemented in several segments. First the document set must be collected. Second the set of vocabulary is determined. Then the topicality of each term is computed. Finally, the predictiveness is calculated and the best terms selected in a recursive fashion.

In the context of web hierarchies, documents are collected by sending a query to a search engine. The system will collect the number of documents specified by the user, unless the search engine returns fewer than the specified amount. The titles and snippets are collected in a file. We then build a database of the documents so that term statistics can be accessed quickly. The document-file, database, and the parameter settings are used to build the hierarchy.

The first step in building a hierarchy is to parse the documents. Initially, a list of single words and phrases are compiled. We use a probabilistic technique to find the phrases[7] and accept any phrase that occurs in at least one percent of the documents or a minimum of two documents, whichever is greater. Single words consist of non-stopwords that are at least three characters in length and are not numbers. Single words must also occur in at least one percent of the documents or a minimum of two documents. We refer to the combination of single words and phrases as the vocabulary.

Once the vocabulary is determined, the topicality is computed. Since a hierarchy is constructed for a single topic, the probability that a term is a topic term is computed once and used throughout the entire process of building the hierarchy. The building procedure then enters the recursive phase. It first computes the co-occurrence language model. Then the terms are selected using the term selection formula in Section 3, implemented using a greedy approximation to the Dominating Set Problem[10]. This method helps ensure that all topics are covered in the hierarchy rather than just the predominant ones.

Once the topics are selected, the documents pertaining to each topic are found. This information is then written to a file, which can be displayed to the user or used by other processes for evaluation. We are currently completing an online demo that will allow users to issue a query to Google, then view the ranked list side-by-side with the corresponding hierarchy.

4.2 Parameters

The parameter settings have a great influence on the time needed to create the hierarchies. When evaluating the system, the time requirements are dependent on both the size of the documents and the number of terms under consideration as topic terms. Although the size of the documents cannot be changed, the number of topic terms can be manipulated. This can be accomplished in several ways. First, one can manipulate the size of the vocabulary set *a priori* through the types of terms considered and by using KL divergence contribution. There are also ways of restricting the terms considered at lower levels of the hierarchy. Finally, terms that occur frequently in the collection are likely to show up at many levels in the hierarchy. There are principled ways of dealing with this repetitiveness. An in-depth discussion of each point follows.

When one considers a document, it is made up of single words and multiword phrases. In order to create a hierarchy faster, the vocabulary can be restricted to only single words or only phrases. Phrases can be more informative than single words; for instance, "Hubble Space Telescope" is much more informative than "telescope" because it is a more specific term. Creating hierarchies containing all phrases is a very good way to limit the vocabulary, but invariably topics are missed because they do not occur in many phrases.

Another method of limiting the vocabulary is to impose a cut-off using the KL divergence contribution of a term. A negative KL divergence contribution means that a term is less likely to appear in the text making up the hierarchy than in general English. One can eliminate these terms since the likelihood that they are actually topic terms is very small. The increase in efficiency is usually worth any possible errors caused by eliminating topics, and in fact, usually helps improve the hierarchy.

By definition, a hierarchy should have general terms at the high levels and become more specific at lower levels. This restriction is not enforced by combining the qualities of topicality and predictiveness. However, the frequency of a term is a good indication of generality or specificity[18]. This means that lower level terms should be less frequent than their parents. Aside from enforcing a quality that should be present in the hierarchy, this restriction also limits the number of terms that will be considered at a given level of the hierarchy, improving the efficiency of the algorithm.

Finally, one of the observed problems is the repetitiveness of terms in the hierarchy. Consider Figure 4 about "Abuses of E-mail", where the hierarchy is created without controlling repetitiveness. The topic "send" occurs in three of the four levels displayed in the figure. This type of repetitiveness can be eliminated by excluding occurrences of "send" at levels below the first occurrence of the topic. In effect, the subtopic is promoted to a higher level. In Figure 4 the subtopics of "send" are explored when "send" is a subtopic of "abuses". There is no need to re-explore the subtopics when "send" is a subtopic of "e-mail". This does not eliminate all repetitiveness, however, since in Figure 2 the topics "combat abuses", "privacy abuses", and "report abuses" appear as subtopics of "e-mail" and "abuses". Since these are two different contexts where the terms occur, both are interesting and valid.

5. EVALUATION

We use three different metrics to evaluate the hierarchies. The first evaluation measures how good a summary the hierarchy is. The second measures the percentage of documents a user can find. The third measures how quickly all relevant documents can be found.

Since the hierarchy is intended to be viewed as a summary, it is important to determine how well it summarizes the text in the documents. This can be done using automated techniques because the hierarchy is a predictive summary, which means the terms that occur in the hierarchies should predict the text. If one were to remove the structure of the hierarchy, a bag-of-words is all that would remain. By treating the documents as a bag-of-words, we can compare the distribution of terms found in the hierarchy to the distribution of all terms. To do this we calculate EMIM, which measures the extent to which the distributions of the two sets deviate from stochastic independence as described in Lawrie and Croft[10]. The greater the dependence between the two distributions, the better the hierarchy is a summary of the text. In Section 5.1 we use this evaluation to show that the terms in the hierarchical summary are better summary terms than an equal number of the top TF.IDF terms. We

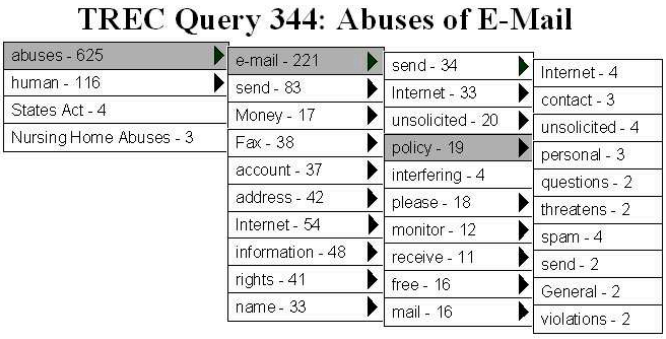


Figure 4: The figure shows a portion of the hierarchy created for the TREC Topic 344: Abuses of E-mail, using a uniform topic model. This hierarchy includes a number of non-topic terms including "name", "address", and "please". Although it performs well in our automatic evaluations, it does not reveal as much information as the hierarchy in Figure 2 created using an unbiased topic model.

also use this test in Section 5.3, when testing hypotheses about the different parameters used to create the hierarchy.

Another important attribute of the hierarchy is the ability to find documents within it, which we refer to as the reachability of the hierarchy. Because of the statistical nature of the hierarchies, there is no guarantee that there will exist a path to all documents used to create the hierarchy. The hierarchy stops adding terms to a level when all the vocabulary is predicted by the terms already chosen. It is possible that this "stopping criterion" may be reached despite the fact that no topic terms are present in a particular document. Another possible reason that a document cannot be found is that it does not occur in any small groups. This condition occurs because of the way we formulate the evaluation. We believe that it is unlikely for a user to explore a group that has become too large. This evaluation imposes different cut-offs as the maximum size group a user would explore, and then calculates the percent of documents that are reachable. In Section 5.2, we test how the reachability of the hierarchy compares to that of using a ranked list. We also use this evaluation when testing our hypotheses about different parameter settings.

Finally, when relevant documents are known, we can test how quickly all the relevant documents can be found. This evaluation is developed in Lawrie and Croft[9]. The evaluation assumes that a user can find the most efficient way to read all relevant documents using the hierarchy while reading as few non-relevant documents as possible. This evaluation favors a hierarchy that has clusters with a very high concentration of relevant documents. The evaluation is only used for testing our hypotheses of parameter settings because we do not have relevance judgments for all collections of documents.

For our test bed, we created several different hierarchies for the TREC Topics 301 to 350. We used the topics as queries to create three different document sets for each topic. The first set consists of 500 documents retrieved from TREC volumes 4 and 5. We have relevance judgments for these documents which enables us to perform the third evaluation. The second set consists of 200 documents retrieved from an all-news database. The third set consists of up to 1000 snippets retrieved using the Google Search Engine. The number of documents in this set varies from 38 to 882 depending on the number of documents the search engine found for a particular query. We have no relevance judgments for the second and third sets of documents. When creating the hierarchies we used a window size of 400 for the first and second sets of documents. This size means that 200 terms on either side of a particular term were used in the estimation of predictiveness. The whole document was used as the window size for the web hierarchies because these documents are so short, consisting of about 25 words.

5.1 Summary Evaluation

Since the summary evaluation evaluates the terms chosen to be part of the hierarchy without considering the structure of the hierarchy, we can compare these terms to any group of terms. We chose to compare the term selection to the top TF.IDF terms. TF.IDF is a popular technique for weighting and selecting terms[16], and has been used as a method of applying labels to clusters[6]. In order to make the test fair, we compare the unique topics in the hierarchy to the same number in the top TF.IDF terms. For example, if there are 286 unique topics in the hierarchy then the top 286 TF.IDF terms would be used as the summary. After calculating the EMIM value of each group of terms, we used ANOVA to compare the values of the two groups. Our hierarchies are sub-divided into groups based on the maximum number of topics in a level – 5, 10, 15 and 20. We used ANOVA comparisons between the hierarchies and the top TF.IDF terms for each hierarchy size. For all three document sets and all four sizes, we found that the topics selected for the hierarchy were significantly better at summarizing the documents than the top TF.IDF terms. We tested for significance using Tukey's Honest Significant Difference at $p = 0.05$. This means that our algorithm is doing a better job of finding summarizing terms than TF.IDF.

5.2 Comparing to a Ranked List

One of the benefits that using a hierarchy offers a user is the ability to find documents that they might not have seen using a ranked list. To illustrate this point, we compared the effects of different policies with hierarchies and with ranked lists. For instance, a user might decide that she will only examine topics in the hierarchy that contain 10 or fewer documents, or that she will only look at the top 100 documents in a ranking. With either of these policies, some documents will remain undiscovered. In this evaluation, we compared the percentage of documents that can be discovered with different policies for both the hierarchy and the ranked list.

The performance of a hierarchy is dependent on the maximum size of the levels in the hierarchy. Figure 5 shows the results of this experiment. A rank followed by the number x is a policy in which the user examines the top x documents in the ranked list. An entry labeled "Hier. Topics = y" indicates a policy where the user looked at topics whose document groups were less than or equal to y. As expected, the hierarchy with the most number of topics in a level allows access to the most number of documents. Also, the policy that looks at the largest size document groups accesses the most number of distinct documents. A policy of looking at document groups no larger than 20 allows access to 52.6% of the documents with a standard deviation of 11.6% over all queries for hierarchies with a level of size 20. Figure 5 shows that is in equivalent to examining the top 400 in the ranked list. With levels of size 15, using the same policy decreases the average to 44.4% of the documents with a standard deviation of 12.0%. This is in equivalent to examining the top 300 in the ranked list . With levels of size 10, the average falls to 33.3% with a standard deviation of 11.0%. This is in equivalent to examining the top 250 in the ranked list. Finally, with levels of size 5, the average number of documents found is 18.8% with a standard deviation of 9.8%. Even with only 5 topics per level of the hierarchy a

user can reach about 150 documents, given that an average number of documents are returned by the search engine. This number of documents is still more than the typical user is likely to see using a ranked list. Because the groups are described in the hierarchy, the documents found are also more likely to be about the topic of interest.

5.3 Evaluating Hierarchy Variations

In order to evaluate the parameters, we tested six different hypotheses related to the settings used to create the hierarchies. These tests evaluated the usefulness of using different topic models, using different groups of terms, using KL divergence contributions to eliminate terms, enforcing the notion of general to specific through the predictive qualtiy, eliminating redundancy, and using sub-collections.

For these tests, we used the same sizes of hierarchies mentioned above, namely those with a maximum number of topics set to 5, 10, 15, and 20. We did this to determine if there was a dependence on size for any of the parameters. We limited our tests to 20 topics because larger hierarchies become too large for a user to completely digest.

We used ANOVA analysis to compare the different versions in the hierarchies. Tukey's Honest Significant Difference at p=0.05 was used to determine where significant differences occur.

The tests confirmed our hypotheses for hierarchies created from TREC collection documents and news documents. First, a hierarchy built using a query model or an unbiased model is a better summary and allows better reachability than a uniform topic model. When an unbiased model is used, making use of sub-collections significantly improves the hierarchy in terms of the summary, reachability, and relevance evaluations[1]. Second, including phrases in the hierarchy helps the performance. In fact, creating a hierarchy out of all phrases creates a better summary, but is the worst in terms of reachability. Using both single words and phrases works well for both types of evaluations. Third, excluding all terms whose contribution to the KL divergence is less than zero significantly improves the summary qualities of the hierarchy in most cases. Unfortunately, this sometimes hinders the reachability of the hierarchy. Fourth, requiring topics predict their subtopics leads to a better summary, and in most instances better reachability. Finally, reducing the redundancy in the hierarchy by disallowing topics to be subtopics of other topics significantly improves most hierarchies both in terms of reachability and being an effective summary.

Interestingly, the same settings do not carry over into web hierarchies. According to our measures, the best web hierarchies are created using a uniform topic model with single words and phrases. An example of such a hierarchy appears in Figure 4. In comparison to the hierarchy in Figure 2, the hierarchy in Figure 4 gives a lot less information about the topics it covered. The abuses related to finance are summarized with terms such as "Money" and "account" rather than "Curb Lending Abuses" and "accounting abuses". In addition, there is a subtopic devoted to "please". This discrepancy between what looks good to a human evaluator and our automated evaluations points to the incompleteness of our tests. Most likely the results of the evaluation are due to the fact that the web hierarchies are summarizing snippets, which are fragments of sentences found in a webpage. By using a query or unbiased topic model, the generated hierarchy is able to summarize the actual documents, but our evaluation tests how well the snippets are summarized, thus creating a discrepancy in the performance. This does not deal with that fact that more documents can be reached using a hierarchy generated using the uniform topic model. This has to do with disjointed text, and that there are only a few topics that appear in many snippets. This is why the hierarchy in Figure 2 has so many tiny groups of documents. In contrast, the hierarchy in Figure 4 has much larger groups of documents at the second and third levels of the hierarchy. This is the one major drawback to generating hierarchies of snippets, though it does not negate the usefulness of such a hierarchy.

[1]This was the only hypothesis where the relevance evaluation showed a significant difference for more than one size hierarchy

6. RELATED WORK

Generating hierarchies is not a new goal for information retrieval, and there have been past attempts using automatic techniques. One example is Crouch[5], who automatically generates thesauri; however, the generated thesauri are not suitable for human use. Another example is Scatter/Gather[6] in which clustering is used to create document hierarchies. However, because of the nature of clustering, fully explaining the contents of each level in the hierarchy is difficult. More recently, new types of hierarchies have been introduced that rely on the terms used by a set of documents to expose some structure of the document collection. One such technique is lexical modification[1, 2, 14] and another is subsumption[18]. The lexical modification techniques have no way of prioritizing which topics are more important than others. This means that the hierarchies tend to be extremely large and unruly. The subsumption hierarchies only include high precision relationships. A term is accepted as a subtopic only if $\mathcal{P}(topic|subtopic) \geq 0.8$. This works well for high quality documents such as news articles, but not for web documents[17]. Our hierarchies have been compared to both types in previous work[10]. These comparisons show that ours perform as well as or better at summarization and finding relevant documents than these alternatives.

From the perspective of multi-document summarization, most automatic techniques extract sentences or portions of sentences from documents[13, 15, 19], and then string them together to form a summary. These techniques are only suitable for a small number of documents and are incapable of handling the larger document sets that hierarchies can.

7. CONCLUSIONS AND FUTURE WORK

This paper describes our design and implementation of hierarchies for web documents. In our preliminary evaluation, we show examples of the usefulness of the hierarchies. We also use a series of evaluation measures developed in previous work to explore different qualities in the hierarchy. We show that the terms selected to be part of the hierarchy are better summary terms than the top TF.IDF terms, and that the hierarchy provides users with more access to the documents retrieved than using a ranked list alone.

When evaluating our choices for parameter settings, the hierarchies created from TREC documents and news articles reinforced our hypotheses. Most notably, we found that using an unbiased or query model is much better than using a uniform topic model. Although the evaluations did not confirm our hypotheses for web hierarchies, we are planning a user study which we hope will demonstrate that the web hierarchies enable users to more quickly and successfully fulfill their information need. Designing and implementing the user study will be a major focus of our future work.

Although there is room to improve the hierarchies, we have reached a point where the hierarchies would be a useful enhancement to the ranked list. The evaluations of the hierarchies created from full-text agree with human inspection of what characteristics are useful, and the same type of hierarchies generated from web summaries reveal useful information to a user.

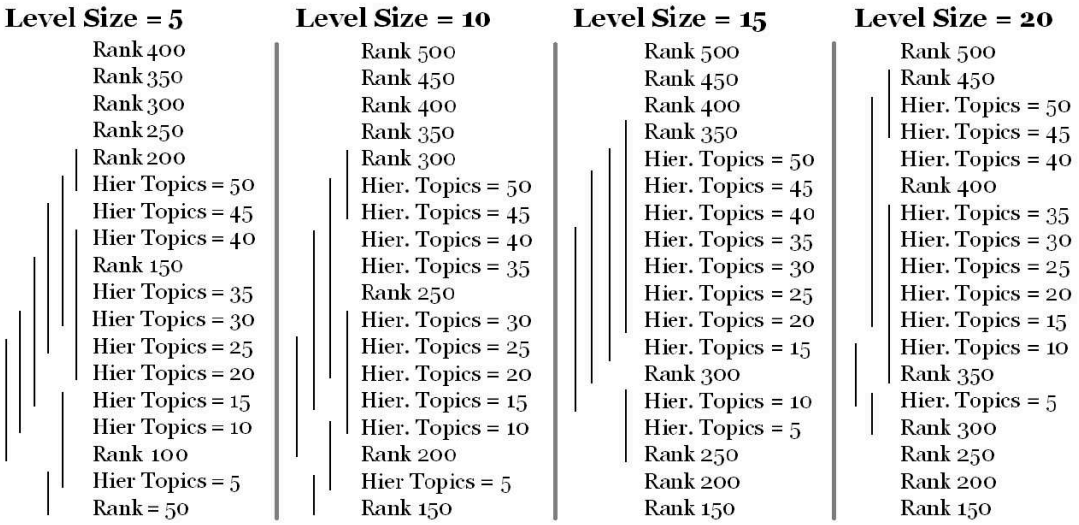


Figure 5: The figure shows the comparisons among different polices when using the hierarchy and a ranked list to search. The lines labeled "Rank x" indicate the maximum rank searched in the ranked list. The lines labeled "Hier. Topics = y" indicate policies using the hierarchy where a user explores any grouping of y or fewer documents. The four charts list the differences for different size hierarchies. The vertical lines indicate where Tukey's Honest Significant Difference test found no differences in the ANOVA analysis. Given how different policies of exploring the hierarchy yield little significant differences, one will not miss a significantly greater portion of the documents by looking only at relatively small groups.

8. ACKNOWLEDGMENTS

We would like to thank Victor Lavrenko, Stephen Cronen-Townsend, and James Allan for their insightful discussions pertaining to this work. We would like to thank Google for providing JAVA apis to access their web collection. We would also like to thank Fangfang Feng for his phrase-extraction software.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

9. REFERENCES

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

[2] 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.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, New York, 1991.

[4] S. Cronen-Townsend and W. Croft. Quantifying query ambiquity. In *Proceedings of HLT*, pages 94–98, 2002.

[5] C. Crouch. A cluster-based approach to thesaurus construction. In *Proceedings on the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320, 1988.

[6] 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.

[7] F. Feng and W. Croft. Probabalistic techniques for phrase extraction. *Information Process Management*, 37(2):199–220, March 2001.

[8] GOOGLE. Google. www.google.com.

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

[10] D. Lawrie and W. Croft. Finding topic words for hierarchical summarization. In *Proceedings on the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, 2001.

[11] 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.

[12] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[13] 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.

[14] 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.

[15] 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.

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

[17] M. Sanderson. Personal communication, September 2002.

[18] 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.

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

[20] 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.

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