Chapter 1

BUILDING, TESTING, AND APPLYING CONCEPT HIERARCHIES

Mark Sanderson
Department of Information Studies
University of Sheffield, Western Bank
Sheffield, S10 2TN, UK
+44 114 22 22648
m.sanderson@sheffield.ac.uk

Dawn Lawrie
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
+1 413 545 0728
lawrie@cs.umass.edu

Abstract A means of automatically deriving a hierarchical organization of concepts from a set of documents without use of training data or standard clustering techniques is presented. Using a process that extracts salient words and phrases from the documents, these terms are organized hierarchically using a type of co-occurrence known as subsumption. The resulting structure is displayed as a series of hierarchical menus. When generated from a set of retrieved documents, a user browsing the menus gains an overview of their content in a manner distinct from existing techniques. The methods used to build the structure are simple and appear to be effective. The formation and presentation of the hierarchy is described along with a study of some of its properties, including a preliminary experiment, which indicates that users may find the hierarchy a more efficient means of locating relevant documents than the classic method of scanning a ranked document list.

1. INTRODUCTION

Manually constructed subject hierarchies, such as the Dewey Decimal system, the U.S. Patent and Trademark Office categories, or the Yahoo directory of web sites[1], are successful pre-coordinated ways of organizing documents. By clustering together documents covering similar topics, the hierarchies allow users to locate documents on specific subjects and to gain an idea of the overall topic structure of the underlying collection. Devising a means of automatically deriving a subject classification from a collection of documents and assigning those documents to the classification is undoubtedly one goal of information retrieval (IR) research[2].

The classic automated method of achieving this aim is based on polythetic clustering (Sparck Jones, 1970) where a set of document clusters are derived from a collection, each cluster being defined by a set of words and phrases, referred to here as terms. A document's membership of a cluster is based on its possession of a sufficient fraction of the terms that define the cluster. Hierarchies of clusters can be constructed by re-clustering each initial cluster to produce a second level of more specific clusters and repeating this process recursively to produce more specific clusters until only individual documents remain. This technique has been used to organize document collections (Cutting et al., 1992), sets of retrieved documents (Hearst and Pedersen, 1996) and groupings of web sites (Chen et al., 1998). Clustering has also been used to arranged query expansion terms (Veling and van der Weerd, 1999).

Although successful at grouping documents containing common terms, automatically labeling a cluster is still an active and important research issue. Two common techniques used to label polythetic clusters are

- showing a list of its most representative terms and
- displaying a number of key passages extracted from the cluster's most representative documents.

Neither method is ideal. To illustrate, the following is a term-based cluster label taken from Hearst and Pedersen, 1996:

- battery california technology mile state recharge impact official cost hour government.

Although one can deduce the topic of the cluster, it is not as concise or as clear a description as the manually generated version given by the authors of the paper: “alternative energy cars”. As well as being verbose, the labels can be overly specific. For example, Cutting et al show in their paper sample clusters (produced by their system) labeled with both passages and term lists. Three

[1] www.yahoo.com

[2] It is of course possible to automatically train a classifier on an existing manually created hierarchy and use it to assign documents to the classification (Larkey, 1999, McCallum et al., 1998). However, the hierarchy may be deficient in the range of topics it covers relative to the documents being classified. Therefore, there will be times when it is desirable to automatically derive a classification directly from a collection.

of the illustrated clusters were labeled as follows, one was about the Gulf War (mentions of the U.S, Iraq, Kuwait, Saudi Arabia), one on oil sales and stock markets, and the other on East and West Germany. Cutting et al combined the documents in these clusters and re-clustered them to reveal that documents about Pakistan, Trinidad, South Africa, and Liberia were in the three original clusters as well. Based on their labels, it is not immediately clear which of the three clusters these documents would have resided in. Essentially, the labels of a polythetic cluster reveal the cluster's central focussed theme. As illustrated, it is quite possible for a cluster to hold documents on topics different from that theme. It follows that if the labels are hard to comprehend or in some way misleading, a user's understanding of the formation and content of a cluster will be impaired. This suggests that an alternative means of grouping documents should be sought.

Polythetic clustering is not the only form of clustering, as Sparck Jones, 1970, points out. There are also monothetic clusters. Like polythetic, these are defined by a set of terms, but a document's membership of such a cluster is based on its possession of all those terms, not just some fraction as occurs with polythetic. This alternative form of clustering has not proved popular in IR, as monothetic clusters composed of many terms are likely to contain only a few documents. However, monothetic clusters composed of a single word or phrase may produce useful groupings. Clearly, such groupings are different from the polythetic clusters illustrated above; however, this form of cluster does address the two issues of labeling and focus[3]. Labeling is simple: the label is the defining term of the cluster. The focus of the cluster content should be clear as documents are only members if they contain the cluster's defining term. Therefore, all members of the cluster will, at the very least, mention the topic specified by the term. This could still be confusing if the term is ambiguous, however, this issue will be dealt with later.

Given the transparent nature of their composition, it is expected that users will find these clusters easier to understand. Indeed most users should be familiar with them already, as a single term monothetic cluster is akin to the set of documents retrieved if that single term were a query. Given the propensity of users to generate short queries (Jansen et al., 1998), this form of document grouping is a common experience for many users. A hierarchical organization

[3] The distinction between monothetic and polythetic clusters reflects the distinction between the classic view of human categorization and the more recent prototype theories as described in the opening chapter of Lakoff, 1987. Like classic categories, the members of a monothetic cluster are considered equally good members of the cluster because they all share the same attributes. As with prototype theory, some members of a polythetic cluster are regarded as better representatives of the cluster than others due to the different range of attributes members can have. Lakoff argues that prototype theory better models the way humans categorize than the classical approach. One might view this as an argument in favor of polythetic clusters; however, the issue presented here is the understandability of clusters, which is a separate notion from the modeling of human categorization.

of single term monothetic clusters will, in form at least, be similar to existing manually created subject hierarchies, which are a familiar means of organization for most users.

Given these anticipated advantages of using single term monothetic clusters, henceforth, called concepts, the means of automatically building a hierarchical organization of these concepts was undertaken. It is this work that is described here. It starts with a review of possible approaches to building a hierarchy, initially examining the utility of a thesaurus, and concentrating on term clustering methods. The means chosen to build the concept hierarchy is then presented, followed by a set of examples illustrating the structure and the technique used to display it. Next, a preliminary user experiment designed to test the properties of the structure is outlined, its results are described. Another method of evaluation is included which measures the ability to find relevant documents within a hierarchy. Finally, conclusions are drawn and future work is detailed.

2. BUILDING A CONCEPT HIERARCHY

In the introduction, it was established that the goal of this work was to automatically produce, from a collection of documents, a concept hierarchy similar to manually created hierarchies such as the Yahoo categories. This was broken down into five basic principles:

- terms for the hierarchy had to best reflect the topics covered within the documents;
- their organization was such that a parent term referred to a related but more general concept than its children, in other words, the parent's concept subsumed the child's;
- the notion of a parent being more general than its children held transitively for all descendants of the parent;
- a child could have more than one parent, therefore, the structure was a directed acyclic graph (DAG) although it is referred to as a hierarchy here;
- and finally, ambiguous terms were expected to have separate entries in the hierarchy one for each sense appearing in the documents.

It might be expected that the relatedness between a parent and child might also hold transitively for all the descendants of the parent; however, as pointed out by Woods, 1997 , some types of relationships between a general concept and its related, more specific descendants are intransitive. Using an example from Woods, a "ship's captain" is a "profession" and "Captain Ahab" is a "ship's captain", but the relationship between "Captain Ahab" and the concept "profession" is less clear. In practice, many parts of a created concept hierarchy may show transitivity in relatedness. With these principles in mind, the building of a hierarchy was addressed, starting with the determination of what sets of documents the hierarchies were to be initially built from followed by finding a means of relating terms to each other.

2.1 BUILD IT FROM WHAT?

The final design principle outlined above forced certain choices to be made about the nature of documents being processed.

As the terms of the hierarchy were to be extracted from documents, it was necessary to know the senses in which they were being used. Though a great deal of work has been expended on performing automatic word sense disambiguation (Yarowsky, 1995, Ng and Lee, 1996), the low accuracy and general lack of availability of such systems effectively precluded the possibility of disambiguating all the words of an arbitrary collection of documents. However, ambiguity could be ignored by choosing to only derive concept hierarchies from sets of documents where ambiguous terms were used in only one sense. For the purposes of this preliminary work, this was achieved by using top ranked documents retrieved in response to a query. Because they all have a similarity to the query, the documents would have a commonality between them, meaning that many of the terms within them would be used in the same sense. (A more general solution that avoids the need for queries and retrieved documents is described in Section 5.3.)

Working with retrieved documents also meant that the set of documents to be processed was relatively small. This had practical benefits as speed and complexity issues would not be a significant problem when developing the software to build the hierarchies. The building of summaries and overviews of a retrieved set of documents is an active area of research (Tombros and Sanderson, 1998) and the creation of a concept hierarchy promised to be a novel approach in this area.

With the issues of which documents to process resolved, the building of the hierarchy could now be tackled.

2.2 RELATING TERMS

From the outset, it was anticipated that a successful concept hierarchy building process would consist of a collection of techniques, which may vary in complexity, coverage, and accuracy. As a starting point, however, it was decided that a relatively simple approach was required that would act as a base on top of which other more sophisticated techniques could be added later.

The planned concept hierarchy was in some ways like the WordNet thesaurus (Miller, 1995): a largely hierarchical organization of terms, organized through a set of relations (synonym, antonym, hyponym-hypernym (is-a/is-a-type-of), and meronym-holonym (has-part/is-part-of)). Therefore, the thesaurus was investigated as a means of relating terms. The WordNet-based term similarity measure, from Resnik, 1995, was used to estimate the relatedness of terms. A small informal experiment was conducted to examine the effectiveness of this method working with terms extracted from fifty sets of retrieved documents and using version 1.6 of WordNet. The main problem encountered was the small number of terms pairs actually found to be related in WordNet. Many pairs that appeared to be have a strong semantic relationship were unrelated in the thesaurus. For example, the terms "volcanic eruption" and "earthquake", both forms of natural disaster, have no connection in WordNet, the former being regarded as an event and the later as a phenomenon. The finding of this small investigation was that the term relationships in WordNet were rarely of any use for the concept hierarchy planned here. What was required was a means of finding broader term relationships that were customized to a particular domain. An obvious area to be examined was term clustering.

Methods for relating terms into graph structures based on document co-occurrence (or co-variance) have been used for many years (Doyle, 1961). The application for most of this work is in query expansion, either automatic (Qiu and Frei, 1993) or manual (Thompson and Croft, 1989, Fowler et al., 1992, Bourdoncle, 1997). Term similarity is calculated using some form of statistical measure, such as the Expected Mutual Information Measure (EMIM) described by van Rijsbergen, 1979.

To the best of our knowledge, most work in term clustering used relations that were symmetric. Our interest was in producing a concept structure with an ordering from general terms to more specific. Forsyth and Rada, 1986, performed such an ordering using the cohesion statistic to measure the degree of association between terms. The number of documents the terms occurred in determined the generality and specificity of terms. This was referred to as a term's document frequency, DF. The more documents a term occurred in, the more general it was assumed to be (the validity of this simple approach to generality and specificity is discussed in Section 2.2.2. The authors reported building a small multilevel graph like structure of terms. Although no testing of its properties were reported, it appeared to be promising. Therefore, it was decided to start with a version of Forsyth's approach, leaving open the possibility of adopting more sophisticated methods for later.

2.2.1 Method used. Although it was used to create a concept hierarchy, Forsyth's term association method was not originally designed to identify the types of association found in concept hierarchies: where, as was stated at the start of this section, a parent node subsumes the topics of its children. Therefore, it was decided to drop cohesion in favor of a test based on the notion of subsumption. It is defined as follows, for two terms, *x* and *y*, *x* is said to subsume *y* if the following two conditions hold,

$$P(x|y) = 1, P(y|x) < 1.$$

In other words *x* subsumes *y* if the documents which *y* occurs in are a subset of the documents which *x* occurs in. Because *x* subsumes *y* and because it is more frequent, in the hierarchy, *x* is the parent of *y*. Although a good number of term pairs were found that adhered to the two subsumption conditions, it was noticed that many were just failing to be included because a few occurrences of the subsumed term, *y*, did not co-occur with *x*. Subsequently, the first condition was relaxed and subsumption was redefined as

$$P(x|y) \geq 0.8, P(y|x) < P(x|y).$$

The value of 0.8 was chosen through informal analysis of subsumption term pairs. The change to the second condition ensures that the term occurring more frequently is the one that subsumes the less frequent. In the rare case of two terms co-occurring with each other exactly, $P(y|x) = P(x|y) = 1$, the two terms will be merged into one monothetic cluster.

Subsumption satisfied four of the design principles outlined at the start of this section:

- as a form of co-occurrence, subsumption provided a means of associating related terms;
- it did not prevent children from having more than one parent;
- the DF of terms provided an ordering from general to more specific;
- and the ordering from general to specific would hold transitively.

As will be seen later on, the subsumption process was adapted further in the light of experiences in implementing the system. Before moving on with term selection, the validity of using DF for determining the generality or specificity of terms is now addressed.

2.2.2 Is DF good enough. One may wonder how well DF models generality and specificity. There is evidence to indicate that it is sufficient. The DF of a query term is successfully used in IR through the application of Inverse Document Frequency (IDF) weighting. Query terms with a low DF are regarded as being more important than those with a high DF when computing a document ranking. There are a number of interpretations of what IDF is modeling, but in the original paper on this weighting scheme, Sparck Jones, 1972 asserts that IDF interprets the specificity of a query term.

More recently, Caraballo and Charniak, 1999 presented results of an experiment that, amongst other things, examined the specificity of nouns based on their frequency of occurrence in a corpus. Caraballo split the nouns she was examining into two groups dividing them on whether they were more general than basic level *categories*[4] or not. Caraballo found that DF worked well determining the specificity and generality of nouns at or below the basic level. But for those above, their DF was a much less effective indicator.

From these two works, it was expected that DF would provide a reasonable ordering of terms from general to more specific, although for terms one might wish to appear at the top of a concept hierarchy, it may prove less successful.

The final issue to be tackled before building the hierarchies was how to select terms from the set of documents from which the hierarchy was to be built.

2.3 TERM SELECTION

Given that the concept hierarchies were to be derived from a set of documents retrieved in response to a query, there were two clear sources of terms: the documents and the query.

The query was expected to be a good source of terms as it was to be processed and expanded using a proven automatic expansion technique called Local Context Analysis (LCA), which works in the following manner. An initial set of documents is retrieved in response to a query in its original form. The best passages of the top ranked documents are examined to find words and phrases that commonly co-occur with each other across many of the passages. The best of these terms are then added to the query and another retrieval takes place. Xu and Croft, 1996, presented experimental results showing retrieval based on the expanded query producing a higher level of effectiveness than that measured from the first retrieval. From these results, it was anticipated that the expansion phrases were well chosen and would be representative of the topics covered in the retrieved documents. Therefore, all words and phrases generated by LCA were used when constructing the hierarchies.

For other words and phrases extracted from the retrieved documents themselves, term selection was a two stage process, first, identification of the words

[4] Lakoff provides a detailed description of basic level categories in the second chapter of his book (Lakoff, 1987). Only a very short and incomplete explanation is provided here. Within a hierarchical categorization of things, basic level categories are to be found in the middle levels of the category. These are the categories most likely to be encountered and mastered when first learning a particular categorization scheme. For example, when categorizing animals, for most people, the basic level categories are the names of animals such as “dog”, “cat”, “cow”, “snake”, etc. Terms below the basic level are specializations, such as "German Shepherd", "Siamese", “Aberdeen Angus”, and “Cobra”. Those above the basic level are more general, possibly esoteric, groupings: clustering “dog”, “cat”, “cow” under the term “mammals”, “snake” under “reptiles”, and “mammals” and “reptiles” under “animate beings”, for example.

and phrases to be extracted, and second, determining which of the extracted terms should be selected for inclusion in the concept hierarchy.

2.3.1 Identifying words and phrases. Given that the documents being processed resulted from retrieval, it was decided to extract terms from the best passages of the documents. It was hoped that this would produce terms that reflected the content of the documents with a bias towards the information need expressed in the query.

Identification of words from the best passages was a simple process of extracting alphanumeric character sequences delineated by common word separators such as spaces, punctuation marks, etc. The extracted words were then stemmed using Krovetz's KSTEM system (Krovetz, 1993). Phrases were extracted using a 'in-house' phrase identification process created within the CIIR group at the University of Massachusetts. The process works best when extracting phrases from a number of documents at the same time. It operates as follows.

Text is first segmented using a number of phrase separators such as: stop words, irregular verbs, numbers, dates, punctuation, title words (e.g. Mr. Dr. Mrs.), company designators (e.g. Ltd., Co., Corp.), auxiliary verbs or phrases, and format changes (e.g. table fields, font changes).

Then the candidate phrases extracted from the text are stored in a lookup table along with their frequency of occurrence in the documents being processed.

Next, the words of the candidate phrases are tagged with all their possible Part Of Speech (POS) tags using grammatical information taken from WordNet. Using a set of syntactic rules, the candidate phrases are checked to see if they are syntactically correct. Those that are not are removed from the lookup table.

Finally, the frequency of occurrence of the remaining phrases is checked. Those occurring more often than a specified threshold are returned as valid phrases. The remaining phrases are searched to find any that have a sub-string (of significant length) in common. For any found, the longer phrase is removed and its frequency of occurrence added to the shorter phrase's occurrence value. If this phrase now occurs more often than the threshold, it is returned by the system as a valid phrase. As a final form of normalization, all valid phrases returned are, like individual words, stemmed using Krovetz's KSTEM stemmer.

With all words and phrases extracted from the best passages of the documents, the process of selecting a subset for the concept hierarchy now took place.

2.3.2 Selecting "good" terms. Term selection used the classic approach of comparing a term's frequency of occurrence in the set of retrieved documents with its occurrence in the collection as a whole. Terms that are 'unusually frequent' in the retrieved set compared to their use in the collection are selected.

The formula used to calculate this value was simply,

$$x_r/x_c$$

where x_c is the frequency of occurrence of x in the retrieved set, x_c is its occurrence in the collection. The extracted words and phrases were each assigned their frequency comparison value and were ranked by this score. The top N terms were selected for inclusion in the concept hierarchy.

With the terms selected, the process to create a concept hierarchy could now take place.

2.4 HOW TO BUILD A HIERARCHY

The process to build a concept hierarchy consisted of a number of phases, which are now described.

First, occurrence information on the extracted words and phrases was gathered. For each term, a list of all the documents that a term occurred in was gathered along with the location of that term within each document. This information was passed onto the subsumption module.

Here, each term's occurrence data was compared to each other term's data to find subsumption relationships. This was an O(n^2) process. All term pairs found to have a subsumption relationship were passed onto a transitivity module.

This final process removed extraneous subsumption relationships. For example if it found that a subsumed b and a subsumed c, but also found that b subsumed c, then the a, c pairing was removed because there was a pathway from a to c via b. The output of this module was the data needed to display a concept hierarchy.

It was decided to test this method out on the 500 top ranked documents retrieved in response to a selection of queries taken from the TREC test collection (Voorhees and Harman, 1998). Retrieval was performed using the INQUERY search engine. After words and phrases were extracted from the documents (on average 12,000 terms from the 500 documents) and their document position information was recorded, the subsumption process took a relatively short time[5] and produced 4,500 subsumption term pairs.

The concept hierarchies that were generated were examined by one of the authors and as a result, an ad hoc modification was made to the subsumption process. It was determined that if x subsumed y and y occurred infrequently, this subsumption relationship was less likely to be of interest. Consequently, terms occurring only once or twice in the document collection were not considered for subsumption.

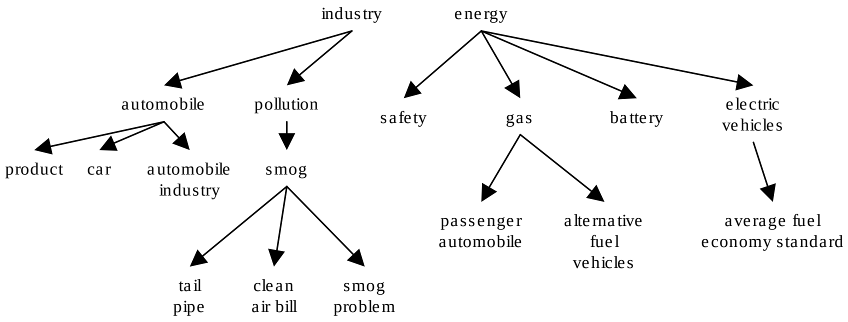
[5] On a 266MHz Pentium II computer with 96Mb of RAM running Linux v5.2, the developmental software used to perform the subsumption process took on average 15 seconds per query.

With the hierarchy creation process determined, an example structure is now displayed and contrasted with other document clustering methods.

2.5 CREATING A HIERARCHY AND CONTRASTING IT WITH OTHER METHODS

Figure 1.1 shows a fragment ($\sim$ 10%) of the concept hierarchy resulting from the 500 documents retrieved in response to TREC topic 230: "Is the automobile industry making an honest effort to develop and produce an electric-powered automobile?". As can be seen, much of the concept organization is promising, especially under "pollution". Other term pairs - "average fuel economy standard" and "electric vehicles" or "safety" and "energy" - seem less sensible. Nevertheless, the hierarchy appears to display the desired property of general terms at the top leading to more specific terms below.

Figure 1.1 Fragment of concept hierarchy from TREC topic 230.



According to Hearst and Pedersen (Hearst and Pedersen, 1996), topic 230 is reminiscent of the topic used to illustrate their system's (Scatter/Gather) creation of polythetic clusters. In their paper, Hearst and Pedersen show documents retrieved in response to the query being assigned to one of five clusters, whose topics are (descriptions taken from paper)

1. "...safety and accidents, auto maker recalls, and a few very short articles";

2. "alternative energy cars, including battery [cars]";

3. "sales, economic indicators, and international trade, particularly issues surrounding imports by the U.S.";

4. "also related to trade, focuses on exports from other countries"; and

5. the final cluster is said to act as a “junk” cluster holding those document difficult to classify.

As can be seen, there is little similarity between the polythetic clusters, and the hierarchy displayed in Figure 1.1. This should not be surprising, however, as polythetic document clustering works quite differently from the monothetic clustering used here. Document clustering is based on finding document-wide similarities to form clusters. In Scatter/Gather, a document is assigned to only one cluster (Sparck Jones, 1970) classifies this as an *exclusive clustering*), consequently, the cluster acts as a summary for that whole document. In contrast, a document can belong to many clusters in a concept hierarchy (which Sparck Jones classifies as *overlapping clusters*); consequently, each cluster represents one of potentially many themes running through a document.

As has already been stated, the organization of terms used in the concept hierarchies is akin to term clustering techniques. To show this similarity, one such system, Refine from AltaVista[6] (Bourdoncle, 1997), is illustrated. Publications about this system are somewhat limited (it appears to be based on a combination of term co-occurrence and term co-variance), but as it is publicly available, it is easy to create a term cluster also reminiscent of topic 230. Figure 1.2 shows the output of Refine after entering the query “auto car vehicle electric” (use of the full TREC topic produced poor output). Each node represents a word grouping, which is expanded via a pop-up menu. Remembering that Refine is working from a different document collection (i.e. the web as opposed to TREC), there is more similarity between its output and the presentation in Figure 1.1 than the output of Scatter/Gather. However, the main difference between Refine and the concept hierarchy is in the organization of terms: the layout of the Refine groups has no apparent significance or ordering.

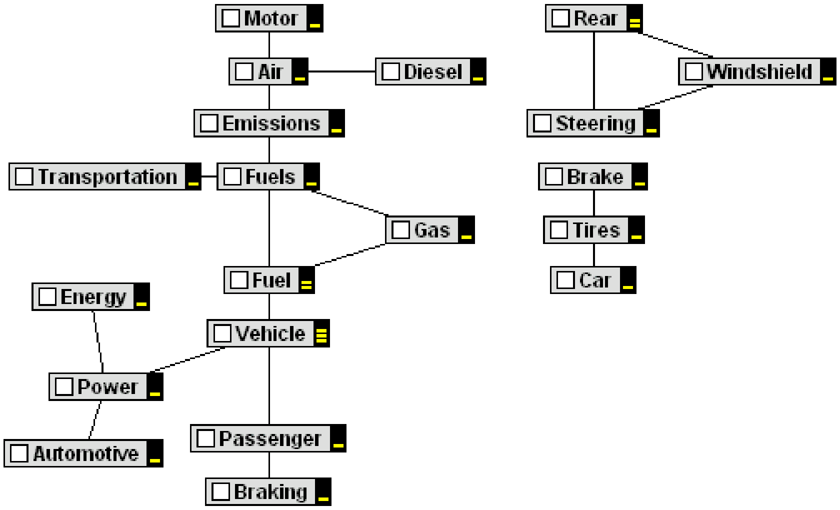
3. PRESENTING A CONCEPT HIERARCHY

As seen in Figure 1.2, it is possible to lay out a small graph structure on screen; however, the concept hierarchies being generated were much bigger: the fragment in Figure 1.1 showed only one tenth of a typical hierarchy. Laying it all out on screen was judged to be potentially complex, time-consuming and maybe even impossible given the size of the structure. Therefore, an alternative means of displaying the structure was examined.

An informal assessment of a couple of possible layout schemes was conducted. The first was a hierarchical arrangement of bullet points. The second involved creating a series of web pages one for each monothetic cluster and for each subsumption relationship between a cluster and other related clusters a hy-

[6] www.altavista.com

Figure 1.2 Clustered term structure from Refine.



pertext link was added to the cluster's web page. Neither presentation worked well, but from this study several priorities were determined

- It was preferable for the structure to fit onto a single screen to avoid the use of scrolling or change of context;
- users should be familiar with the interface components used to present the structure;
- users should be able to move around the structure easily and quickly; and
- when at a particular 'level' in the hierarchy, users should be able to easily determine the possible paths that led to that level.

The means of presentation found to hold to almost all of these priorities was a hierarchical menu.

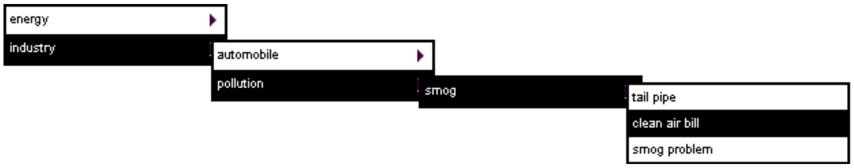
- Because menus only show the current menu plus the path of menus used to get there, the chances of getting the structure to fit in a single screen was higher.
- Hierarchical menus have been a standard feature of operating systems for many years.

- Due to their familiarity, users can generally move around menus with relative ease.
- Hierarchical menus are used to display a strict hierarchy, where a child has a single parent. A child in the concept hierarchies, however, can have multiple parents, and observing that a child has multiple parents can be important information to a user. Unfortunately, there was no immediately obvious solution to this problem. As menus were judged to be a good means of presentation, the problem was ignored and any child having multiple parents was duplicated and placed under those parents. No link was displayed between the copies of the child.

The hierarchical menu system chosen was one written capable of being displayed within a web browser[7]. Most menu systems are designed to allow a user to get to a known item in a sub-menu as fast as possible without making a mistake. This is generally achieved using delays related to mouse movement, which temporarily prevent the closing of the currently open sub-menu. Such a provision was not helpful for the task required here as the user was to be encouraged to browse around the entire structure as fast as possible. The menu system obtained did not have such delays and so was well suited to the browsing task.

To illustrate the look of the menu system, the sample structure in Figure 1.1 is shown in its menu form in Figure 1.3.

Figure 1.3 Menu version of structure displayed in Figure 1.1.



3.1 LIMITATIONS OF THE MENUS

Certain limitations in the workings of the chosen menu system along with restrictions of screen size meant that additional constraints had to be imposed onto the concept hierarchy formation process.

The first was caused by the large size of the hierarchy structures. If a term in the hierarchy had a great many parent terms, in this menu system, the child term

[7] www.dhtmlab.com

was duplicated and appeared under each of its parent terms. If the child was itself a parent to a great many other terms, the size of the menus became very large and the menu display code failed. Consequently an appearance limit was placed on all terms in the hierarchy: any appearing more than a certain number of times (typically 25) were removed completely from the structure. While this action appears somewhat draconian, it was necessary to enable the menu system to function properly. It is worth noting that a better implementation of a hierarchical menu system would in all likelihood avoid this problem.

With so much information being displayed, screen space was inevitably an important issue. A limit on the vertical size of a menu was consequently imposed. On a large display, the limit was set to 30 terms per menu. A menu larger than this limit was simply truncated loosing its extra terms. In order to ensure that less important terms were those that were lost, the terms within a menu were sorted based on their DF as it was found that terms with a high DF appeared to be more important. This ordering can be seen in the examples illustrated in the next section: 3.2.

A final problem was so-called 'singleton menus': those containing only one term, such as the “smog” menu in Figure 1.3. A large number of these were found to exist in the created concept hierarchies. As they use up a lot of horizontal screen space, the menu creation procedure was adapted to merge the term of a singleton menu into its parent term and remove the offending menu.

With these final adaptations in place, an example of the menu display is now presented.

3.2 EXAMPLE HIERARCHY FRAGMENTS

Figure 1.4, Figure 1.5, and Figure 1.6 shows three parts of a concept hierarchy, this time generated from TREC topic 302: “Poliomyelitis and Post-Polio: Is the disease of Poliomyelitis (polio) under control in the world?”. The number next to each term is the *DF* of that term, which, therefore, is the number of documents assigned to that particular monothetic cluster. It is worth noting that at the top level of the hierarchy (the left most menu in the Figures) the DFs of all the terms on that level add up to more than the 500 documents the hierarchy was built from. This is an indication that there are documents appearing in more than one place in the hierarchy. It should also be noted that it is possible for documents to be missing entirely from the hierarchy, due to them not containing any of the terms that were subsumed.

As has been seen, from the three figures as well as the structure in Figure 1.1, there is a trend of general terms leading to the more specific. One can see that “Salk” (inventor of a polio vaccine) appears both in the “polio” and the “disease->vaccine” sections of the hierarchy; both sensible locations for this term. The structure while initially satisfying could be improved: in Figure 1.4 for example,

Figure 1.4 A fragment of concept hierarchy from topic 302

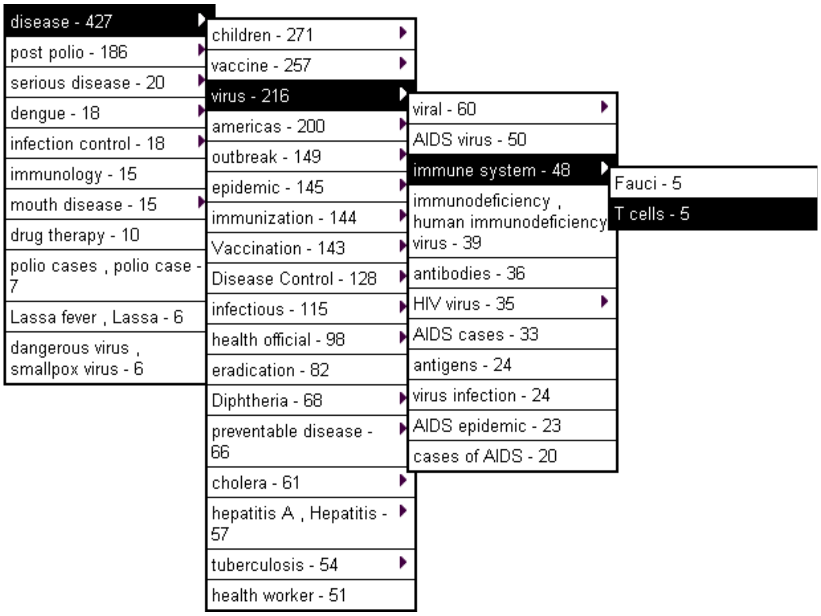


Figure 1.5 Second fragment of concept hierarchy from TREC topic 302

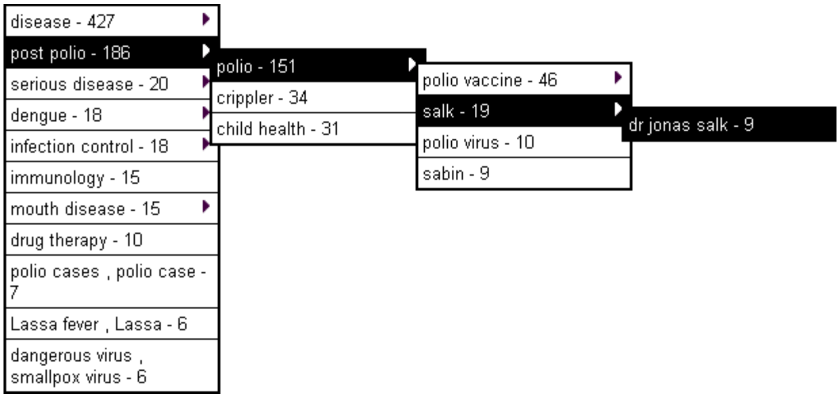
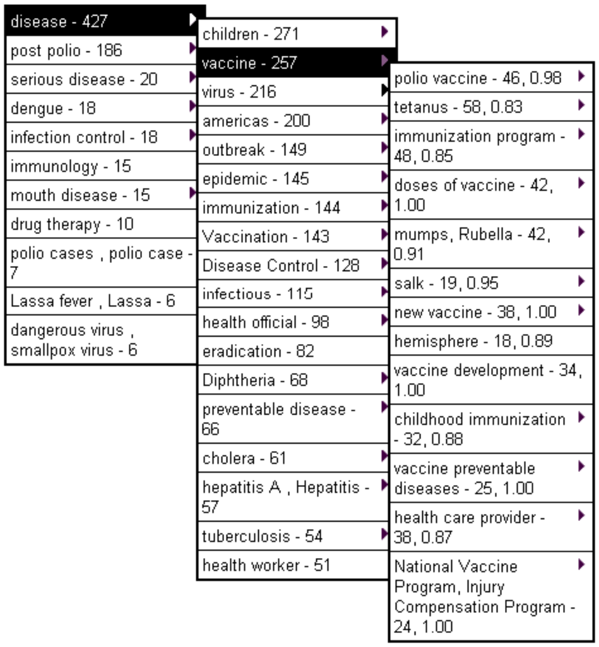


Figure 1.6 Third fragment of concept hierarchy from TREC topic 302



“Fauci”, the surname of an AIDS researcher, might have been better categorized under “AIDS” instead of “virus”. Nevertheless, as a first step towards building a concept hierarchy, the structure appeared to be promising.

4. EVALUATING THE STRUCTURES

Evaluating the concept hierarchies presented a challenge; their intended purpose was to provide users with an overview of the topical structure of the documents retrieved in response to a query. Measuring how well something provides an overview was not going to be counted by some objectively derived value. In a paper on user evaluation of Scatter/Gather, Pirolli et al (Pirolli et al., 1996) reported using a method aimed at testing how well users understood the topical structure of documents after seeing Scatter/Gather clusters. Unfortunately, the test involved asking users to draw a concept hierarchy, something that would inevitably be influenced after seeing the structures generated here.

Before taking on a large-scale user study of the hierarchy’s over-viewing capabilities, it was felt that some of the basic properties of the structure should be examined first. Therefore, an experiment was created that addressed the second and third design principles outlined at the start of Section 2.: testing the relatedness of a child to its parent; and examining the type of relationship between the two. The details of the experiment are described in an earlier paper (Sanderson and Croft, 1999). The results of the experiment found that approximately 50% of the subsumption relationships within the concept hierarchies examined were found to be of interest and that the parent term was judged to be more general that its child. This figure compared favorably to concept hierarchies created with a random formation process.

Another use of the hierarchies is as an aid to finding relevant documents. Rather than examining a ranked list of documents from a retrieval system, a hierarchy can be used. By using knowledge of the query topic, a person can follow paths in the menus that lead to relevant documents. There are at least two aspects to the problem of finding relevant documents within the hierarchy. One is how long it takes to traverse the hierarchy once it is known where relevant documents are located. If it is found that traversal (in this situation of having perfect knowledge about relevant documents) is better at locating relevant documents than scanning down a ranked list, then the other aspect of the problem can be studied. This is how easily humans locate the menu pathways that lead to a relevant document.

4.1 THE TRAVERSAL ALGORITHM

Our algorithm estimates the time it takes to find all relevant documents by calculating the total number of menus that must be traversed and the number of documents that must be read. The algorithm aims to find an optimum route through the hierarchy travelling to nodes that hold the greatest concentration of relevant documents. Since we begin with the knowledge of where documents are located, our algorithm iterates through all the relevant ones and assigns a path length to each. Any relevant documents not found in the hierarchy (which is possible) are assigned a path length of negative one. The total path length for a hierarchy is the summation of all non-zero (relevant) document paths. The algorithm follows.

Given that documents often belong to more than one menu, it is necessary to choose which of these will be used when calculating the path. To do this, we break the menus into two groups. The first group consists of leaf menus. These types of menus are favored because they tend to have a smaller number of documents associated with them. Smaller document groups are also likely to be more homogenous. From among these leaf menus, we favor the menu with the most relevant documents because we are computing an optimal path. If there are no leaf menus, then all menus containing the document are considered. In this case, we favor menus that contain a small number of documents, since it is unlikely that a human would read more documents than necessary.

[t]

Figure 1.7 Document path length algorithm

```
Document path length algorithm {
   for each relevant document $d$ {
      if ($d$ seen before?)
         $\{d.path\_length = 0\}$
      else {
         find all leaves with $d$
         if ($num\_leaves > 0$) {
            $lm$ = menu with max # rel docs
            $d.path\_length = lm.new\_menus + lm.total\_newdocs$
         }
         else {
            find all menus with $d$
            if ($num\_menus > 0$) {
               $m$ = menu with min # total docs
               $d.path\_length = m.new\_menus + m.total_{n}ew\_docs$
            }
            else
               $\{d.path\_length = -1\}$                    /*no path*/
         }}}}
```

The path to a relevant document is composed of the previously unexplored menus that are traversed to reach it and the unread documents associated with the final menu. As the documents belonging to a particular menu item are not sorted in any way, it is assumed that users will have to read all new documents in the menu in order to find the relevant one(s).

Although this algorithm leads to a succinct analysis of the concept hierarchy, it is worth noting that it contains certain simplifying assumptions. First, all documents are regarded as equal despite the expected variability in document length. Similarly, all menus are treated equally despite the variability in their length. Finally, when computing the path length, documents and menus are treated the same, i.e. the time and effort to read a document is regarded as being the same as that to read a menu.

4.2 EXPERIMENTS

Our experiment makes use of TREC topics 301-350 and associated relevance judgements. We have retrieved 500 documents using INQUERY for each of the 50 queries. We treat a set of 500 documents for a given query as a document set. Concept hierarchies are generated for each document set.

Hierarchies are assigned a path length score using the algorithm described above. A lower score denotes a superior hierarchy. We compare our hierarchies to those formed through a random subsumption process. These hierarchies were formed in the same manner as the concept hierarchies (as described in Section 2.4 except that when all terms were compared to all other terms, random selection was used to form parent-child pairs instead of subsumption. Note the ordering of terms based on frequency of occurrence was still present in this structure.

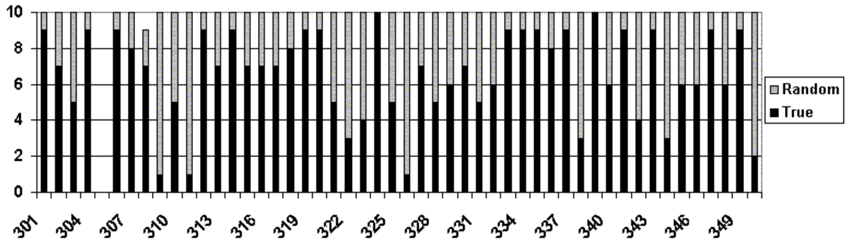
Once all the menus were scored, they were compared on a basis of the average path to a document. This was used instead of doing a straight comparison of the total path length because it was possible that some relevant documents were unreachable. The total path length for a particular hierarchy could end up being shorter simply by leaving out relevant documents. By using the average path length, we neither rewarded nor penalized a hierarchy for excluding relevant documents. It was found empirically that randomly generated hierarchies were more likely to leave relevant documents out of the hierarchy than the true hierarchies. The true concept hierarchies contained no path to a relevant document 1.9% of the time. The random menus contained no path to a relevant document 19.4% of the time. These percentages are based on the number of relevant documents excluded compared to the total number of relevant documents.

4.3 RESULTS

4.3.1 Evaluation Relative to Random Hierarchies. Ten randomly generated hierarchies were created for each query. The average relevant document path length was then averaged among the randomly generated hierarchies before comparing the path lengths so that an average baseline hierarchy could be compared to the true hierarchy. In 41 of the 50 queries, the true hierarchy had a smaller average document path than the baseline hierarchy. When the true hierarchy had a smaller path, it was on average 5.03 units shorter for each relevant document. In 8 of the 50 document sets, the baseline hierarchy had smaller document paths. However, these paths were only 2.15 units shorter on average. The paths were equal in one case where INQUERY retrieved no relevant documents within the document set. Figure 1.8 compares each randomly generated hierarchy to the true hierarchy. The black part of the column represents the number of times that the true hierarchy had a shorter path length than the random ones. The gray part of the column represents the number of times that a random hierarchy had a shorter path than the true hierarchy. In cases where the column has a height less than ten, there were random hierarchies of exactly the same path length as the true hierarchy. Query 305 had no relevant documents so all hierarchies are equivalent, which is why there is no column.

We performed ANOVA (ANalysis Of VAriance) on the data. To linearize the data for the ANOVA, we performed a log transform on the average path length, indicating that the average path length differed between the models by a multiplicative factor. We discovered that the path lengths from the random hierarchies were 33% longer (7.11 vs. 5.35) than the path lengths from the concept hierarchies (p < 0.0002). (Path lengths are geometric averages.) See Appendix 7., Table 1.A.1 for ANOVA table.

Figure 1.8 Shows the comparison of the true hierarchy to ten random ones.



4.3.2 Evaluation Relative to Ranked List. Since a ranked list is a widely used method of displaying retrieved documents, we also compare our hierarchies to the ranked order that INQUERY generates for the retrieved document set. In order to deal with the difference in the number of relevant documents, we used the same average document path length as was used in comparing the hierarchies. This means that the lowest ranked document is treated as the total path length of ranked list. The path length is divided by the total number of relevant documents since all relevant documents within the set will be ranked. When the scores from the hierarchy are compared to INQUERY's ranked list, the hierarchy required the user to read fewer documents in 47 of the 50 topics. On average, 224.2 fewer documents and menus were read. In the two cases where INQUERY required fewer documents to be read, the difference in the number of documents read was on average 12.5. Again the topic where INQUERY returned no relevant documents had the same scores.

We performed ANOVA on the average path length data comparing the random hierarchies, the concept hierarchies, and the INQUERY ranked list. The log transform was the best model again. The randomized path length was 33% longer (7.11 vs. 5.35) than the concept hierarchy path length (p < 0.001 by Honest Significant Difference (HSD) paired comparison). The INQUERY ranked list generated a path length 156% longer than the concept hierarchy path length (13.67 vs. 5.35), and 92% longer than the randomized path length (13.67 vs. 7.11) ($p < 1.0E\text{-}12$). (Again, path lengths are geometric averages.) See Appendix 7., Table 1.A.2 2 for ANOVA table.

5. FUTURE WORK

As was stated in Section 2.2, the work described so far is only a starting point in the automatic building of a concept hierarchy. A number of potential improvements to the formation process are now presented, followed by a brief discussion of alternative means of presenting the hierarchies, concluded with ideas for their wider use.

5.1 IMPROVING TERM IDENTIFICATION

Currently a simple phrase extraction process performs identification of concepts within documents; however, there are a number of other utilities created within the field of Information Extraction (IE) which may improve identification accuracy. A Named Entity Recognizer (NER) is a basic tool used to perform initial text processing in an IE system (Wakao et al., 1996). It locates and types common text forms such as proper nouns, dates/times, money expressions, postal addresses, etc. For proper noun recognition, name lists for people, places, and companies may be used. It is anticipated that use of such a mark-up tool will better inform the term selection process by avoiding text types that are unlikely to be good terms, such as email addresses or phone numbers. In addition new conceptual groupings will be possible based on the NER types such as the names of people or companies related to a particular term.

One other IE tool that will be examined is co-reference resolution. This tool finds different references to the same concept in text. The range of co-references that such a system can tackle is large, but for the purposes of this project only Proper name co-references will be resolved (Wakao et al., 1996). For example, determining if, in a document, the name “Dr. Jonas Salk” and the name “Salk” refers to the same person. Successful use of this tool would group multiple references and thus remove duplicates from the concept hierarchy.

5.2 WIDENING THE RANGE OF CONCEPT RELATIONSHIPS

Although subsumption identifies relatively accurately a large number of valid concept relationships, it is believed that a range of other existing methods can be employed to increase this number and will provide validation of existing relationships.

The subsumption-based work used so far was found to be successful in providing a set of concepts organized into a hierarchy leading from the most general concepts to the most specific. No attempt was made to locate synonymous relationships. There is a body of work on using forms of statistical co-occurrence to locate such relationships. One such technique is co-variance. Two concepts are said to co-vary when the contexts in which they occur are similar. Grefenstette, 1994, has had success in locating synonym relationships using co-variance. This technique will be applied to the concept hierarchy formation process to group sets of synonymous concepts. Another source of information on synonymous relationships is a thesaurus.

Despite the relatively poor utility provided by WordNet in the small investigation outlined in Section 2.2, it was felt that a sufficient number of concepts were successfully related to warrant a re-examination of WordNet. In the concept hierarchy illustrated above, for example, "polio" and "poliomyelitis" are located in different parts of the hierarchy despite being synonyms of each other; use of WordNet would concatenate these two terms into a single concept. In addition, WordNet may also provide some evidence on the generality and specificity of concepts to further improve the hierarchy formation process particularly for terms above basic level categories, where, as Caraballo and Charniak, 1999, has found, DF is a poor indicator of generality or specificity.

In addition to use of an existing thesaurus to locate hyponym/hypernyms and synonyms, a number of corpus based techniques have been developed to locate such relationships. Hearst, 1998, found that certain key phrases could be an indicator of such a subsumption-like relation. Three of the phrases she found were

- "such as", e.g. "...popular forms of entertainment such as movies...";
- "and other", e.g. "...Julia Roberts, Robert De Niro and other actors...";
- "especially", e.g. "...most horror films, especially Psycho and The Exorcist.".

Sentences that contained these phrases were parsed to identify the noun phrases being related. Hearst discovered around ten such phrases that were accurate identifiers of the "type-of" relation. However, manual intervention was required for their discovery and the scope of the noun phrase pairs identified was limited. Hearst suggested using the key phrases to help thesaurus lexicographers search for new relations. Use of this technique could be applied to the formation of the concept hierarchies and an investigation of its utility will be conducted.

In a similar vein to Hearst's work, a series of key phrases could also be identified to locate terms that are synonyms within the context of a subsuming term. Working from the examples shown above, "Julia Roberts" and "Robert De Niro" are both actors, "Psycho" and "The Exorcist" are both horror movies.

Observing that these terms are components of a list should be a relatively simple task.

Two pieces of work on phrase analysis are also promising avenues of research. Grefenstette, 1997, has described a method of phrase classification, where, through the use of simple syntactic analysis, he was able to place noun and verb phrases into one of nine classes. He illustrated his ideas by examining all possible phrases containing the word "research". For example depending on whether "research" was the head or the modifier of a noun phrase, Grefenstette was able to differentiate types of research (e.g. market research, recent research, scientific research, etc) from research things (e.g. research project, research program, research center, etc). No tested application of this classification scheme was reported.

Woods, 1997, also used phrase analysis in addition to a large knowledge base to organize terms into a concept hierarchy. By locating the head and modifier of noun and verb phrases, Woods was able to make choices on how to classify phrases. For example in the phrase "car washing", Woods' system would identify "car" as the modifier and "washing" as the head of the phrase. This would inform the system to classify the phrase "car washing" under "washing" and not "car". The success of the technique relied on a large morphological knowledge base of information to help identify phrase components. Woods used the concept hierarchy to automatically expand non-matching terms of a query.

5.3 CREATING HIERARCHIES WITHOUT QUERIES

In Section 2.1 it was noted that concept hierarchies rely on words being used in the same sense. It is thought that a homogeneous document set provides an environment where word sense disambiguation is not an issue. Using the top ranked documents of a query is one way to achieve the desired environment. An alternative method is to create polythetic clusters of the document set. Concept hierarchies can be then be created in cases where there is no query. In fact the hierarchy becomes a description of the polythetic cluster. The hierarchy does not suffer from the traditional problems of labeling polythetic clusters which may leave out the topics of sub-clusters since only the most frequent words are used in the description. A concept hierarchy seeks to create a complete description of the document set, and thus creates a complete description of the cluster.

Lawrie and Croft, 1999, studied the effectiveness of using clustering as a preprocessing of the document set before creating the hierarchy. It was found that this can expose more relations in a document set than using a single hierarchy for the entire set. However, some relations may be left out because a group of documents that formed a subpart of the initial single hierarchy are clustered into different groups and no longer have a sufficient number of occurrences independently for inclusion in the hierarchy. In the task of finding relevant documents, as described in Section 4.1, creating hierarchies of clusters provides a faster method to finding relevant documents.

5.4 VISUALIZATION OF HIERARCHIES

Currently, presentation of the concept structure is achieved using hierarchical menus. Although simple to manipulate and interpret, this form of visualization looses some of the information held within the structures, the reason (as described in Section 3.) being that the hierarchical menus visualize a strict hierarchy, one parent to each child. The actual data, however, has children possessing multiple parents, which can be important. For example, in a hierarchy built from documents on international conflicts, the child term “war” had two parents “India” and “Kashmir”. Seeing this shared link helped users' understanding of the concept organization. Currently, the hierarchical menu system handles this situation by placing a copy of such a child under each of its parents, the hope being that a user will notice the child term under each parent and mentally make the link between them.

Alternative visualizations will also be explored. Much work has been conducted on tools to visualize directed acyclic graph (DAG) DAGs and some are freely available such as the daVinci system (Fröhlich and Werner, 1994). One of these tools will be selected and applied. It remains to be seen how well these tools will display as large a structure as that currently being generated (each hierarchy holds several hundred concepts). If the use of these tools fails to be successful, an alternative will be to work within the existing menu framework and produce a system whereby any child can be expanded in some alternate manner to show a list of its parents.

5.5 ALTERNATIVE APPLICATIONS FOR THE HIERARCHIES

The work presented here attempts to provide an automatically constructed meaningful categorization of documents that was similar to manually constructed categories. The intended use of this structure was, like their manual equivalents, to allow users to locate documents of interest within the hierarchy and to provide users with an overview of the topic structure of the document collection being categorized.

The manual topic structures can have additional uses as well, for query expansion and for organizing documents written in foreign languages (Pollitt et al., 1993). Both these alternative uses are now discussed.

5.5.1 Query expansion. It is a well known feature of searching that the vast majority of queries submitted to widely available IR systems are very short, typically one or two words in length (Rose and Stevens, 1996). Query expansion whether it is through automatic or semi-automatic means (Xu and Croft, 1996, Harman, 1992) or via manual intervention (Magennis and van Rijsbergen, 1997) has been shown to increase the number of relevant documents retrieved. What has not been successful is persuading users to employ these techniques.

When presenting automatically extracted expansion terms to users, most systems present these terms in a simple list. The concept hierarchies could be used to sensibly organize these words and phrases to make the range of possible expansion terms easier for users to process. Some related work has already been conducted in this area which indicates this may be a promising line of enquiry.

Anick and Tipirneni, 1999, presented a technique that attempted to select terms that reflected the main topical threads running through a collection of documents. To do this, the method looked for terms that had a high "lexical dispersion": terms that occurred with many other different terms. Anick showed the terms with the highest lexical dispersion in a collection of Financial Times documents were “market”, “group”, and “company”. He used lexical dispersion to select words and phrases from a set of retrieved documents and present these terms to users as candidates for query expansion. Not only were the terms shown but all the phrases that those terms were part of were shown as well through a series of menus. The authors presented an analysis of access logs to a retrieval system using the expansion method. It is unclear in the paper how often the expansion terms were used, but when they were, expansion appeared to be of use.

Taking a less statistical and more NLP based approach, Bruza and Dennis (Bruza and Dennis, 1997) presented their hyperindex system. Working on top of a web search engine, their system parsed the titles of retrieved documents and looked for the query phrase in conjunction with other words linked by certain connectors (“in”, “of”, “with”, “as”, etc). The new phrases were presented to the user in a structured fashion, showing phrases that were either restrictions or expansions on the existing query. All new query phrases were derived through this simple parsing technique. Titles of documents were used because they were mostly expressed in passive form, which was easier to work with when finding new phrases. The paper claimed that the titles parsed fairly well. No users testing of the system was reported in the paper.

Both papers have presented means of structuring query expansion terms, though neither has presented a large user study to examine the utility of their respective techniques. Therefore, although expansion clearly can be presented in this structured form, its utility remains to be determined. If concept hierarchies are to be investigated as a means of query expansion, such a study will have to take place.

5.5.2 Use in a cross language environment. A considerable amount of research has been conducted on the cross language retrieval problem (retrieval based on a query written in (what is referred to here as) a *source language* retrieving documents written in (what is referred to here as) a *target language*). The best results approach the effectiveness of a monolingual system (Ballesteros and Croft, 1998).

The most likely outcome of a user session with a CLIR system is the need to translate some of the retrieved documents back into the source language. Such a process is usually costly and time consuming. Consequently, it is in the interest of the user of such a system to locate, as accurately as possible, the best set of relevant documents. In a monolingual retrieval system, users refining their query through several retrieval iterations would normally achieve this. In order for users of a cross language system to conduct a similar refining process, it is necessary for them to be able to assess, at some level, the relevance of the retrieved documents. Full automatic document translation is not accurate, one approach is to generate a translated concept hierarchy.

Translating a target language concept hierarchy into a source language is not as hard as it might appear at first. As the translation is occurring in the context of a retrieval system, there are certain features that can be taken advantage of. First, there already exists a set of translated terms - those of the query - and these can be exploited. Second, the documents to be retrieved have a degree of similarity to them and this quality will also be beneficial. We start by working with the query terms.

In the work by Ballesteros and Croft, 1998, a successful method of cross language retrieval was described, one aspect of which involved the expansion of users' original queries with other source language terms, which were then translated into the collection language to produce an effective target language query. From this form of retrieval, a reliable mapping exists between the translated terms in the retrieved documents and the expanded query terms. As a starting point, one can build concept hierarchies from these translated terms alone. Because of the existing mapping, further translation is not necessary. Although the resulting hierarchies will be small, they will still be of use to users unable to read the target language documents.

As was found with monolingual concept hierarchies, their quality and richness can be improved by including terms found within the documents in addition to those of the query. The accurate translation of the additional terms will be conducted using Dagan's technique, which is designed to work with minimal translation resources (Dagan et al., 1991). When translating a particular term, the context in which it occurs is used to disambiguate the term. If that term occurs in other retrieved documents, it is reasonable to assume it will be used in the same sense throughout those documents. All the contexts, therefore, can be conjoined to provide more information to make the disambiguation, and therefore, the translation more accurate.

It is believed that the translated concept hierarchies show great promise in conveying the topical structure of retrieved documents, and a series of initial attempts are planned for future work.

6. CONCLUSIONS

Through use of a simple term association technique, a method for building concept hierarchies has been presented. The hierarchies were informally compared to other methods that derive structure from collections of documents. From this comparison, it was shown that a hierarchical organization of monothetic clusters is quite different from polythetic document clustering. Through two small-scale experiments, it has been shown that the generated concept hierarchies provide some level of sensible organization of concepts and provide a reasonable means of access to relevant documents.

7. ACKNOWLEDGEMENTS

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, and also supported in part by United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235.

Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor(s).

The authors wish to thank Russell Swan for his help in the experimental analysis.

Appendix: ANOVA analysis

Table 1.A.1 Compares concept hierarchies to random ones

	DF	*SS*	*MS*	*F*	*P-value*
CONSTANT	1	2019.9	2019.9	8416.35025	0
qf	48	220.8	4.5	19.16696	0
sysf	1	3.6111	3.6111	15.04645	0.00011934
ERROR1	489	117.36	0.24		

Table 1.A.2 Compares concept hierarchies, random hierarchies, and INQUERY

	DF	*SS*	*MS*	*F*	*P-value*
CONSTANT	1	2334.4	2334.4	8865.76987	0
qf	48	244.41	5.0919	19.33863	0
sysf	2	24.358	12.179	46.25429	0
ERROR1	537	141.39	0.2633		

References

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

Ballesteros, L. and Croft, W. (1998). Resolving ambiguity for cross-language retrieval. In Croft, W., Moffat, A., van Rijsbergen, C., Wilkinson, R., and Zobel, J., editors, *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, Melbourne Australia.

Bourdoncle, F. (1997). Livetopics: recherche visuelle d'information sur l'internet (livetopics: visual search for information on the internet). In *Proceedings of RIAO (Proceedings of RIAO (Recherche d'Informations Assistee par Ordinateur - Computer Assisted Information Retrieval)*, pages 651–654.

Bruza, P. and Dennis, S. (1997). Query reformulation on the internet: Empirical data and the hyperindex search engine. In *Proceedings of RIAO (Recherche d'Informations Assistee par Ordinateur - Computer Assisted Information Retrieval)*, pages 488–499.

Caraballo, S. and Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing (EMNLP) and very large corpora (VLC)*, pages 63–70.

Chen, H., Houston, A., Sewell, R., and Schatz, B. (1998). Internet browsing and searching: user evaluations of category map and concept space techniques. *Journal of the American Society for Information Science*, 49(7):582–603.

Cutting, D., Karger, D., Pedersen, J., and Tukey, J. (1992). 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.

Dagan, I., Itai, A., and Schwall, U. (1991). Two languages are more informative than one. In *Proceedings of ACL'91: the 29th Annual Meeting of the Association for Computational Linguistics*, pages 130–137.

Doyle, L. (1961). Semantic road maps for literature searchers. *Journal of the Association of Computing Machinery (ACM)*, 8(4):553–578.

Forsyth, R. and Rada, R. (1986). Adding an edge. In *Machine Learning: applications in expert systems and information retrieval*, Ellis Horwood series in artificial intelligence, pages 198–212. Chichester: Ellis Horwood: Halsted Press, New York.

Fowler, R., Wilson, B., and Fowler, W. (1992). Information navigator: An information system using associative networks for display and retrieval. Technical Report NAG9-551, #92-1, Department of Computer Science, University of Texas, Pan American Edinburg, TX 78539-2999.

Fröhlich, M. and Werner, M. (1994). The graph visualization system davinci - a user interface for applications. Technical Report 5/94, Department of Computer Science, Universität Bremen, Bremen, Germany.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

Grefenstette, G. (1997). Sqlet: Short query linguistic expansion techniques, palliating one-word queries by providing intermediate structure to text. In *Proceedings of RIAO (Proceedings of RIAO (Recherche d'Informations Assistee par Ordinateur - Computer Assisted Information Retrieval)*, pages 500–509.

Harman, D. (1992). Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 1–10, Copenhagen Denmark.

Hearst, M. (1998). Automated discovery of wordnet relations. In Fellbaum, C., editor, *WordNet: an electronic lexical database*. MIT Press.

Hearst, M. and Pedersen, J. (1996). Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, Zurich, Switzerland.

Jansen, B., Spink, A., Bateman, J., and Saracevic, T. (1998). Real life information retrieval: A study of user queries on the web. *SIGIR Forum: A Publication of the Special Interest Group on Information Retrieval*, 32(1):5–17.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 191–202.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things*. University of Chicago Press.

Larkey, L. (1999). A patent search and classification system. In *Proceedings of the 4th ACM conference on Digital libraries*, pages 179–187.

Lawrie, D. and Croft, W. (1999). Discovering and comparing hierarchies. Technical Report IR-183, CIIR, Department of Computer Science, University of Massachusetts, Amherst, MA 01002.

Magennis, M. and van Rijsbergen, C. (1997). The potential and actual effectiveness of interactive query expansion. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–332.

McCallum, A., Rosenfeld, R., Mitchell, T., and Ng, A. (1998). Improving text classification by shrinkage in a hierarchy of classes. In Brasko, I. and Dzeroski, S., editors, *Machine Learning: Proceedings of the 15th International Conferences (ICML '98)*, pages 359–367. Morgan Kaufmann Publishers.

Miller, G. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ng, H. and Lee, H. (1996). Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. In *Proceedings of ACL'96: the 34th Annual Meeting of the Association for Computational Linguistics*, volume 34, pages 40–47.

Pirolli, P., Schank, P., Hearst, M., and Diehl, C. (1996). Scatter/gather browsing communicates the topic structure of a very large text collection. In *Conference proceedings on Human factors in computing systems (ACM CHI '96)*, pages 213–220.

Pollitt, A., Ellis, G., Smith, M., Gregory, M., Li, C., and Zangenberg, H. (1993). A common query interface for multilingual document retrieval from databases of the european community institutions. In *Proceedings of the 17th International Online Information meeting (Online '93) Learned Information*, pages 47–61. Learned Information.

Qiu, Y. and Frei, H. (1993). Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 160–170. ACM Press.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453.

Rose, D. and Stevens, C. (1996). V twin: A lightweight engine for interactive use. In *NIST Special Publication 500-238: The 5th Text REtrieval Conference (TREC-5)*, pages 279–290.

Sanderson, M. and Croft, W. (1999). 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.

Sparck Jones, K. (1970). Some thoughts on classification for retrieval. *Journal of Documentation*, 26(2):89–101.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Thompson, R. and Croft, W. (1989). Support for browsing in an intelligent text retrieval system. *International Journal of Man Machine Studies*, 30:639–668.

Tombros, A. and Sanderson, M. (1998). Advantages of query-biased summaries in ir. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 2–10.

van Rijsbergen, C. (1979). *Information retrieval*. Butterworths, London, second edition.

Veling, A. and van der Weerd, P. (1999). Conceptual grouping in word co-occurrence networks. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 694–699.

Voorhees, E. and Harman, D., editors (1998). *The 7th Text REtrieval Conference (TREC-7)*. Department of Commerce, National Institute of Standards and Technology.

Wakao, T., Gaizauskas, R., and Wilks, Y. (1996). Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 418–423.

Woods, W. (1997). Conceptual indexing: a better way to organize knowledge. Technical Report TR-97-61, Sun Labs, Editor, Technical Reports, 901 San Antonio Road, Palo Alto, California 94303, USA.

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

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 189–196.