Concept Maps: Theory, Methodology, Technology
Proc. of the Second Int. Conference on Concept Mapping
San José, Costa Rica, 2006

RANKING CONCEPT MAP RETRIEVAL IN THE CMAPTOOLS NETWORK

Thomas C. Eskridge, Adrián Granados, Alberto J. Cañas,
Florida Institute for Human and Machine Cognition (IHMC), USA
www.ihmc.us

Abstract. Concept maps are very good at organizing knowledge about a wide variety of subjects. However, they present some difficulties and opportunities when it comes to the retrieval of concept maps based on individual query terms. In this paper, we review the search architecture of CmapTools and some of the key issues involved in the ranking of concept maps search results. We compare several different methods for ranking the results of a query term search on concept maps. We draw conclusions and present some ideas for future work.

1 Introduction

The use of concept maps has expanded rapidly since their introduction in the 1970's by Dr. Joseph Novak (Novak and Gowin 1984). Concept maps are two-dimensional, visual representations of the relationships between concepts representing individual knowledge, collaborative group consensus, and corporate memories (Cañas, Hill et al. 2004; Coffey, Eskridge et al. 2004). Users from elementary school to universities have used concept maps to develop new ideas, organize information, and preserve knowledge in a wide variety of subject domains (Novak 1998).

CmapTools is a client server software environment developed at the Florida Institute for Human & Machine Cognition (IHMC) that facilitates the construction and sharing of concept maps. The software has been downloaded numerous times in countries around the world, and over 100 public CmapServers have been established that enable users to share their own concept maps, collaborate with others, and discover information on a wide range of subjects.

The task of discovering relevant information in concept maps is the focus of this paper. CmapTools has a search facility that integrates the results of search queries that are executed simultaneously on client's local search indices as well as on the indices of all other public servers. The number of results returned from even a simple search can be on the order of several hundred concept maps and other resources. In order to enhance the user experience with CmapTools, the first results returned should be the most relevant resources for the query issued.

The growing CmapTools network has prompted us to investigate the current search result ranking method, to quantify its performance, and to compare it to other methods that have the potential to produce more relevant rankings. This paper presents preliminary results on an experiment designed to compare and quantify CmapTools search ranking method performance. The paper is structured as follows: First, we describe the current client-server search architecture of the CmapTools environment. We then discuss the general characteristics of search in CmapTools. The competing ranking methods and experiment design are described, and the results of the experiment are presented. We discuss the findings from the experiment, and discuss some ways to improve the experiment and results.

1.1 Search Architecture

The search architecture in CmapTools encompasses searching over three different groups of indices and integrating the results at the client in the standard CmapTools search window (see Figure 0). A client side search index is created for the concept maps and resources stored locally on the user's "My Cmaps". This index will contain all of the information about the resource stored locally, and will be accessible only by the local user. Each server creates and maintains a search index that is typically forwarded to an IndexServer (the solid lines in Figure 0). The IndexServer is a special CmapServer whose purpose is to aggregate CmapServer search indices and execute search requests over the entire population of indices at one time. Each time a search is performed, queries are issued to the local index, all available IndexServers, and directly to any CmapServer that is not registered with an IndexServer. These results are grouped and

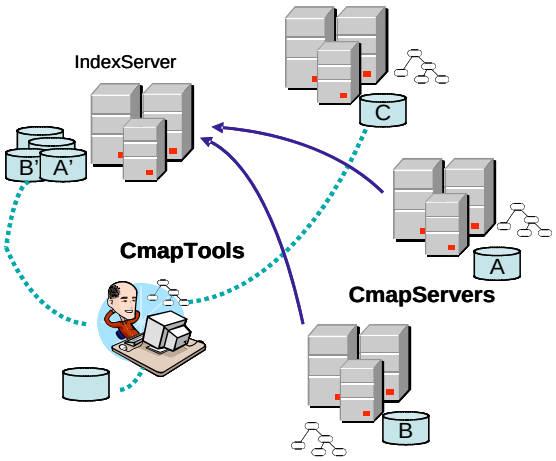

Figure 0. CmapTools Search Architecture

sorted for presentation in the Search dialog box. When a search result is found, it is returned to the client interface (see Figure 1). The user can double-click on the search result to open the resource directly from the server on which it is stored.

1.2 General Search Characteristics

The search index is created at startup time and is updated whenever concept maps or resources are modified or saved. The search index contains information from the concept map, such as the concept and linking-phrase labels, as well as meta-data, such as the description, keyword, title, and author of the map. By default, all indexed information is used to satisfy search queries: A query term that appears in the description or keyword fields of a concept map will cause the map to be retrieved.

The standard search in CmapTools makes a number of default assumptions:

1. **All queries are OR'ed**. Executing a search for the query *Tangerine Apple*, will result in documents that match the word *Tangerine* or documents that match the word *Apple*, and possibly, but not necessarily both. To get documents that are required to match both words, the query terms must be explicitly AND'd together, like this: *Tangerine AND Apple*.
2. **All queries are case insensitive**. All indexed content of the concept map is converted into lowercase text during index creation. As the query is issued, it is also converted to lowercase text.
3. **Search terms with greater than four characters are searched using "wildcards"**. If you search for the term *Apple*, the CmapTools search facility internally searches for **apple**, which means that it will return documents containing the words *Snapple*, and *applesauce*. Wildcards are not used on search terms with less than four characters because doing so can result in the generation of very large (internal) queries that significantly detract from the performance of the search engine.
4. **Search terms enclosed in quotes are treated as a single term**. Searching for "*Tangerine Apple*", will return all documents where the words *tangerine* and *apple* are next to each other and separated by a space. This makes it easy to search for documents containing things like the names of people. If you were to search for *Mark Johnson* you would get all documents containing the words *mark* or *johnson*. However, if you searched for *"Mark Johnson"*, only documents with that particular name will be returned. Also, wildcards are not added to search terms enclosed in quotes.
5. **Boolean ordering can be accomplished using parentheses**. To order the logical combinations in a query, parentheses can be used. For example, the query *animal plant* returns resources that are either plants or animals. But to prevent search from returning manufacturing plants as well as green plants, we can write the query like this: *animal (plant AND green)*. Now the search returns documents that are either

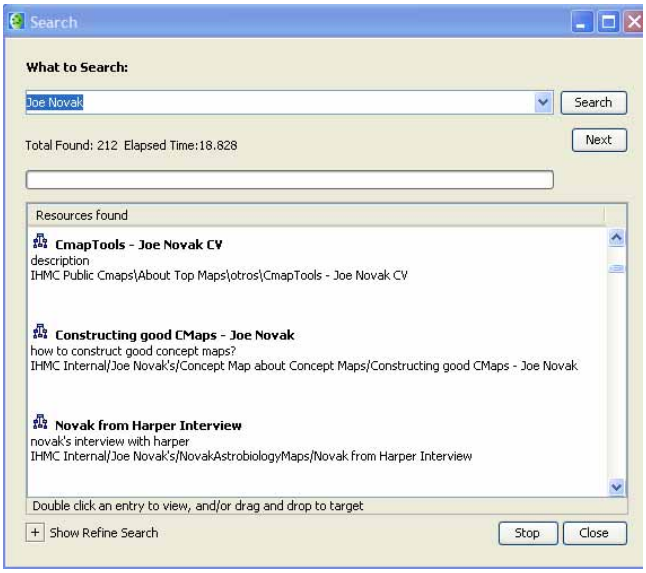

Figure 1. Results are integrated in the Search dialog.

an *animal* or that contains both *plant* and *green*. Note that the documents containing *plant* and *green* may also contain the word *animal.*

6. **Handling Negation.** Negation is handled using the *NOT* query keyword. One important limitation CmapTools handling of negation is that it is not proper to search using only negation. Thus, a query such as *NOT manufacturing* logically means the search should return all resources do not contain the term *manufacturing*, but is illegal in CmapTools. So an additional term must be added to this. Thus, a search for *plant NOT manufacturing* returns documents that mention the term *plant* but that do not mention the term *manufacturing.*
7. **Handling permissions.** The results of a search are filtered by the permission settings on the individual servers. Because of this, users who do not have authorization to open a resource, for example, by clicking on it in the Views window, will not receive that resource as a search result. This has implications for users who protect their maps on a server, but try to access them from another CmapTools client. Unless they have entered a username and password that can open the resources stored in the protected folder, the resources will not be returned as part of the search.

For the purposes of this experiment, only assumptions 1, 2 and 3 are utilized.

2 Ranking Methods

The inverted-tree indexing structure used by CmapTools ensures that a document will be retrieved if it matches any query term. There is no limit on the number of results retrieved from the index present on the CmapTools client. However, we currently limit the number of results retrieved from CmapServers and from the IndexServers. Because of this, it is important that the most relevant maps are conveyed to the user first. The ranking methods order the list of search results with the goal of providing the most relevant search results at the top of the list. We are interested in improving the search result ranking in CmapTools because finding relevant results quickly not only improves productivity, but also enhances the user experience.

2.1 Keyword Match (KM)

This is the most simplistic matching algorithm used in this test, and is also the default ranking algorithm for CmapTools. It simply counts the number of query term matches in the document, and normalizes this result to the number of terms in the query, with bonuses for keyword matches in the title (0.001), and for matching concept maps (0.0001).

$$Score_{km} = \sum_{i=0}^{K} t_i \Big/ K + TITLEBONUS + CMAPBONUS$$

Despite its simplicity, this approach has demonstrated generally satisfactory results. This is in part due to the most common type of query being issued is a “location” query. That is, the user remembers a seeing particular map, but not where the map was. Because it is remembered, it is usually a simple task to search with query terms that will easily discriminate the desired map. By issuing the query with multiple discriminatory terms, and ranking the results according to how many of the terms are matched in the indexed documents, the desired map (or resource) will often appear in the top results.

However, as the adoption of CmapTools grows, we envision more use of the search as a knowledge exploration tool. That is, users will search concept maps to gain new knowledge rather that to locate maps that they know already exist.

2.2 TF-IDF (TI)

The TF-IDF methodology was developed by (Salton and McGill 1989). The TF-IDF vector space model of information retrieval is described in (Baeza-Yates and Ribeiro-Neto 1999) as a clustering model where the intra-cluster similarity term (to be maximized) is defined by the frequency of the term in the document (which indicates how well the term describes the document), and the inter-cluster similarity (to be minimized, or dissimilarity to be maximized) is the inverse document frequency. This factor is based on the

idea that the more a term is shared between documents, the less able it is to distinguish between the two documents.

$$Score_{TI} = \frac{\sum_{i=0}^{K} w_{i,d} \times w_{i,q}}{\sqrt{\sum_{i=0}^{K} w_{i,d}^2} \times \sqrt{\sum_{i=0}^{K} w_{i,q}^2}}$$

The weights for the terms in the index are computed according to formula $w_{i,d} = f_{i,d} \times \log\frac{N}{n_i}$, where $f_{i,d}$ is the normalized frequency of term *i* in document *d*. The weights for the query terms are computed using the formula $w_{i,q} = \left(0.5 + \frac{0.5\, freq_{i,q}}{\max_l freq_{l,q}}\right) \times \log\frac{N}{n_i}$, where $freq_{i,q}$ is the raw frequency of term *i* in the query.

2.3 Partitioned TF-IDF (PTI)

The degree of similarity between the retrieved concept maps and the query is computed from a vector representation of the concept maps similar to the term-frequency vector with inverse-document frequency adjustment (TF-IDF) (Baeza-Yates and Ribeiro-Neto 1999) but that also considers the unique structure of the map. It adjusts weights according to the distance of the concepts to the top concept and the number of outgoing and incoming links to a concept (Leake, Maguitman et al. 2003). Using TF-IDF adjusted by the number of links, a keyword that appears in a concept that has no outgoing or incoming links will have zero weight. From the search perspective, all keywords that appear in the concept map must have a weight greater than zero. Thus, the number of outgoing and incoming links for every concept is increased by one. Then, the weight of keyword *i* in map c_d is computed as:

$$Score_{PTI} = \frac{\sum_{i=0}^{K} w_{i,d}}{\sqrt{\sum_{i=0}^{K} w_{i,d}^2}} \text{ where } w_{i,d} = (n + m + 1) * \sqrt{1/(h+1)}$$

In the formulas above, *n* is the number of incoming links, *m* the number of outgoing links and *h* is the number of steps between the concept that contains the keyword and the top concept of the map. A slight adjustment is made to include the keywords of the title of the concept map in the computation. The number of steps from any concept to the top concept is increased by 1, so that the title is positioned at level 0 (as if it were the top concept), the top concept at level 1, and so on. The scoring function for the PTI ranking method is identical to the TI method, where each query term weight is 1.

2.4 Concept Distance (CD)

For each retrieved concept map, the degree of similarity is computed by analyzing the proximity of the search terms in the concept map. The distance is zero if both terms appear in the label of a concept; otherwise, the distance is the shortest of all possible paths that exist from a concept that contains a search term to the concept that contains the another search term. The system computes the relevance of the concept map *j* as:

$$Score_{CD} = 1 \Big/ n * \sum_{i=0}^{n} 1/(d_i + 1)$$

In the formula above, *n* is the total number of pairs of terms and *d* is the minimum number of steps between the terms of the pair *i* in the concept map *j*. In this algorithm, the title of the concept map is also treated as a top concept, so that keywords from the title are weighted and considered in the similarity computation of the map as well.

2.5 Combined PTI-CD

The combined PTI-CD ranking method computes a simple weighted sum of the PTI and CD scores. The motivation for this is that the PTI and CD methods involve complementary methods of using the concept

map structure to influence the search result ranking. By combining the two methods, we expect the resulting performance to be better than PTI alone. Since CD requires more than one search term to be used, it cannot be used by itself, but must be combined with another method capable of single term queries.

3 Experimental Setup

To test our hypothesis that the new ranking algorithms would improve the rate at which relevant documents are presented in the top 5 results, we conducted an experiment aimed at determining how effectively search ranks the results of CmapTools searches.

The motivation for the experiment is that CmapTools users often remember seeing particular pieces of information while browsing through hyperlinked concept maps, but may not have a convenient way to retrieve that information. The CmapTools search ranking algorithms should aid the user in finding this information if they accurately reflect the relevance of concept maps to the search query. If the map containing the recalled information is ranked high in the result set following the query, the ranking measure is considered accurate. If it is not, then it is considered inaccurate. Our principal question is "What is the performance of the current CmapTools ranking algorithm?" Our secondary question is "Can changes in the ranking method provide better results?"

3.1 Method

3.1.1 Data preparation

We constructed a data set based on the concept maps that were stored on the IHMC Internal CmapServer as of Jun 2006. From this server, we removed all documents that were written in a language other than English, were protected by a username and password, or were not concept maps. We then created a search index for this data set that computed and stored the weights associated with the different ranking methods in the index.

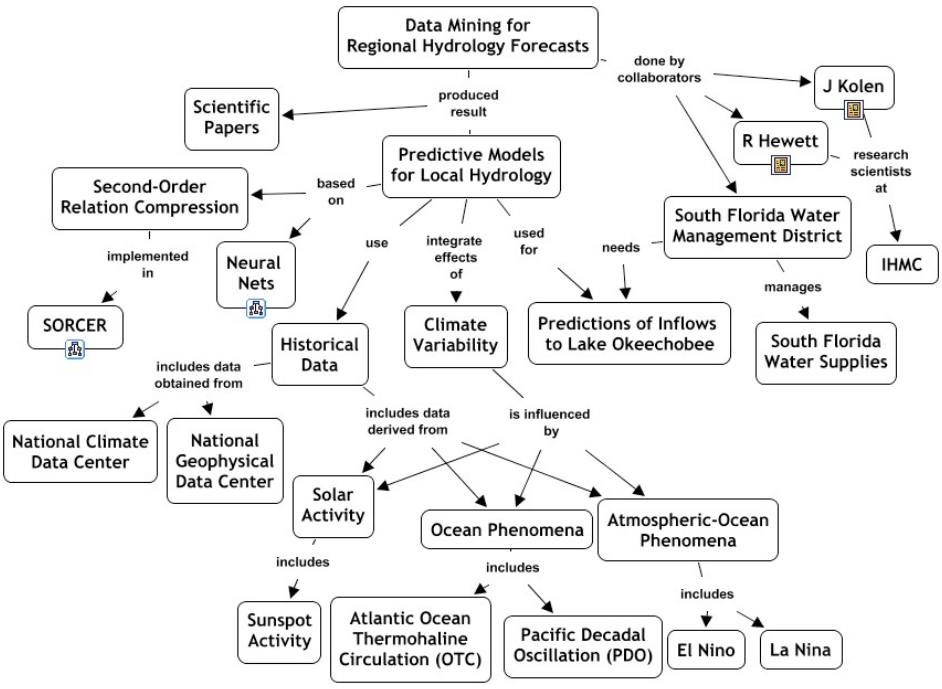

Figure 1. Example concept map typical of those used in the ranking study.

As part of the indexing process, we randomly selected 20 maps for potential use as targets. Of these, we found 10 that were well-constructed concept maps. The maps that were not selected generally had defects such as being entirely composed of unlabeled concepts ("????" in CmapTools), having three or fewer concepts, or being entirely disconnected.

3.1.1.1 Question generation

Eight people volunteered to participate in this experiment. Subjects were asked to examine each concept map in the test set and to write down two questions whose answers can be found in the map. An example map used in this experiment is shown in Figure 3.

We asked subjects for two questions to avoid having only the obvious "What is <root concept>?" types of questions. Many subjects asked this question, but then asked a more relevant question as their second attempt. When subjects did ask the "What is <root concept>?" question, it was removed from the test set, as virtually all algorithms perform very well on this type of query. We also discarded non-responsive and duplicate questions, leaving 98 queries out of a possible 160.

3.1.1.2 Query generation

After the questions were selected, each was converted into a query. This was done by eliminating stop words and connectives. Table 1 shows some examples of typical questions and their corresponding queries.

Subject Question	Search Query
What is Regional Hydrology Forecasts?	regional hydrology forecasts
How can you predict Local Hydrology?	predict local hydrology
What are uses of predictive models?	uses predictive models
What are the current results from data mining for regional hydrology forecasts?	current results data mining regional hydrology forecasts
How can data mining aid the construction of predictive models for local hydrology?	data mining aid construction predictive models local hydrology
How can we build predictive Hydrology Models?	build predictive hydrology models
What does Data Mining for Regional Hydrology Forecasts do?	data mining regional hydrology forecasts

Table 1. Converting questions to queries.

3.2 Evaluation Criteria

Each of the queries was run seven times, once with each ranking method. The results were recorded to a data file, which was subsequently analyzed.

The resulting data file indicated where the concept map that was used to generate the question for the search query appeared in the list of search results. Ideally, the map would appear very high in the search results, within the first five results (which is the number of results visible in the CmapTools Search dialog.)

While it is typical to report ranking experiments in terms of "precision" and "recall" – the number of relevant retrieved document and the number of relevant document retrieved, respectively – this would have required our test subjects to rank the relevance of all results retrieved, or to use a specially constructed data set. Since we wanted to test our algorithms on "live" data, either of these solutions would require a level of effort beyond the resources available for this work. Therefore, we report our results in terms of the position in which the map corresponding to the query appears in the search result list.

4 Results

The data collected from the four ranking methods are shown in Figures 4 and 5. Figure 4 shows how many times the example map was found in the first five positions (<5) of the search result list, the fifth through tenth positions (<10), and so on. The performance of the PTI ranking method was particularly

surprising – being the lowest performing algorithm of the group - as a similar method has proven very effective in clustering concept maps (Leake, Maguitman et al. 2003). **Figure 2** shows the cumulative percentage of example maps found over the same rank location bins as Figure 4. It is interesting to compare the convex shape of the curves for the KW and TI ranking methods with the concave shape of the curves for the PTI (75-25), CD, and PTI methods.

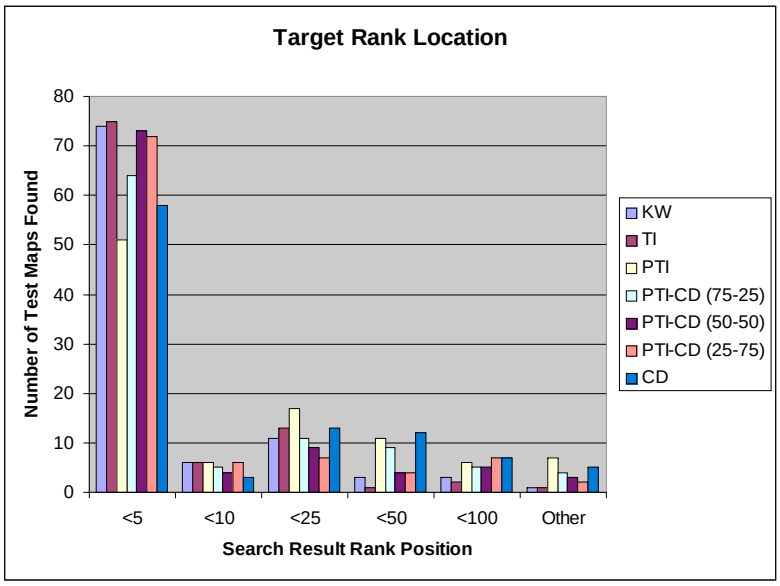

Figure 3. Number of example maps found in each rank position.

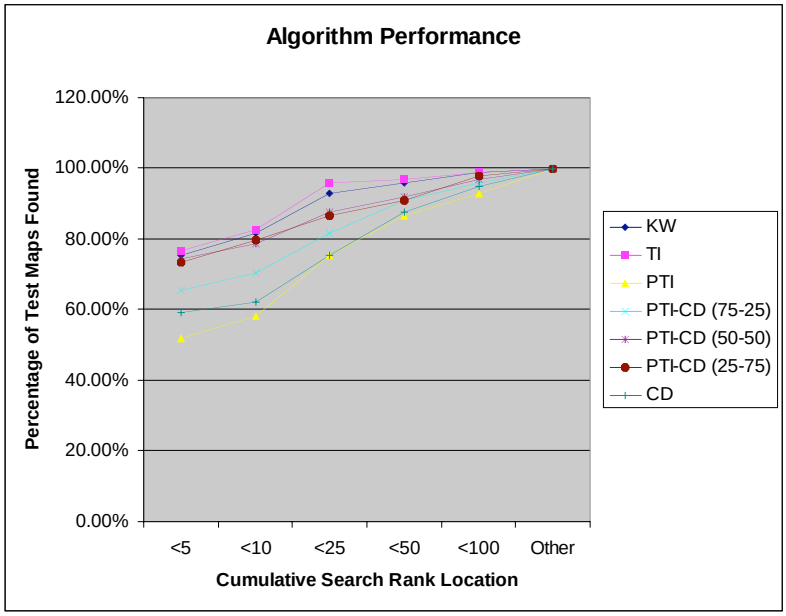

Figure 2. Cumulative percentage of example maps found by rank position.

5 Discussion and Future Work

The results of this experiment countered the intuitions of the authors. We initially expected virtually all of the algorithms to be as good as or better than the KW algorithm. However, the results show that the KW

and TI algorithms perform better than the remainder of the algorithms. We initially expected that the PTI algorithm, especially when combined with the CD algorithm would be the best performer of the lot. Our thinking was that because PTI computed TF-IDF based on location in the concept map, it would allow the search to distinguish between maps that contain concepts as central terms versus those that incidentally mention those concepts. However, individually PTI and CD had the worst performance of the group. Interestingly, the 50-50 and 25-75 combinations of PTI and CD performed significantly better than the two algorithms individually. In future work, we will explore this connection further, and also experiment with the addition of CD data to the KW and TI algorithms. Because of the hierarchical structure of concept maps, the information provided by CD, while not enough to rank results on its own, may provide the extra bit of discrimination necessary to move above 75% of results in the top 5 search rank locations.

Further analysis of the KW algorithm has shown that much of its accuracy comes from the small bonus awarded for finding the keyword in the title of the concept map. Removing this bonus results in the percentage of example maps found in the first five positions falling from 75.51% (74/98) to 58.16% (57/98) making it's performance on par with PTI and CD. This suggests that weighting *where* the keywords are found may be a good way of increasing overall accuracy of any of the algorithms.

This again points to the intuition that the PTI algorithm should be more effective than its performance indicated in this experiment. One reason for this lack of performance may have been that this experiment addressed finding a *particular* map rather than a wide ranging set of maps on a particular subject. On a more broadly evaluated search, the PTI algorithm may do better. Therefore, evaluation of our future experiments will include looking at the top maps found and evaluating the algorithms based on their relevance to the query. For example, even though the map corresponding to the test query was found in location 5, it does not mean that the results in locations 0-4 are erroneous or less relevant. They may be even more relevant than the map used to generate the query. This testing will require considerably more effort that our current experimental methodology, but the benefits of experimentally determining the "optimal" concept map ranking algorithm will extend to the entire CmapTools community.

6 Acknowledgements

We would like to thank Dr. Marco Carvalho and the IHMC CmapTools Research Group for their assistance in the design and implementation of this experiment.

References

Baeza-Yates, R. and B. Ribeiro-Neto (1999). Modern information retrieval. Harlow, England, Addison-Wesley.

Cañas, A. J., G. Hill, et al. (2004). CmapTools: A Knowledge Modeling and Sharing Environment. Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping, Pamplona, Spain, Universidad Pública de Navarra.

Coffey, J. W., T. C. Eskridge, et al. (2004). A Case Study in Knowledge Elicitation for Institutional Memory Preservation Using Concept Maps. Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping, Pamplona, Spain, Universidad Pública de Navarra.

Leake, D. B., A. Maguitman, et al. (2003). Topic Extraction and Extension to Support Concept Mapping. Proceedings of FLAIRS-2003, AAAI Press.

Novak, J. D. (1998). Learning, creating, and using knowledge : concept maps as facilitative tools in schools and corporations. Mahwah, N.J., L. Erlbaum Associates.

Novak, J. D. and D. B. Gowin (1984). Learning how to learn. Cambridge Cambridgeshire ; New York, Cambridge University Press.

Salton, G. and M. J. McGill (1989). Introduction to Modern Information Retrieval. New York, McGraw-Hill Book Co.