MemoPlex Browser: Using a Semantic Network for Browsing and Searching a Document Collection

Yoel Lanir °

Department of Computer Science

University of British Colombia

ABSTRACT

The two dominant ways of finding information in a document collection are searching and browsing. These two paradigms are complementary: searching is able to provide specific information quickly, while browsing is useful in cases where appropriate search keywords are nonexistent, or unavailable to the user. In this work, I introduce MemoPlex Browser, an interactive tool for searching and browsing a document collection. MemoPlex Browser uses the semantic similarities between the documents as links to smoothly and efficiently browse from one document to another. In addition, it uses clustering to give a better overview of the collection's content, and to maintain user context while browsing.

Keywords: Information Visualization, Searching, Browsing.

1 INTRODUCTION

Retrieving information in an efficient way has been a major task of information management systems. Software companies like Google offer retrieving solutions which are based on strong indexing algorithms. These solutions use an interface in which the user formulates the search query and gets a list of possible results. If the users do not find what they are looking for they need to refine their query, resubmit it, and look over the results again.

While direct search is fast when the target is known, it fails when the user does not have an exact idea of the target. The user may not be familiar with the vocabulary describing a specific topic, or may not be able to formulate a query answering his or her specific request. If the user selects a search query which is too general or vague, the search results can be too large to find the target. Conversely, if the search query is too narrow, the results may exclude the target. Browsing begins where such failures occur.

Browsing can be a useful addition to conventional search methods. The user could enter a search query, look at one of the result nodes, and then browse from one node to another according to some sort of association rule between the nodes. Following the links between the nodes according to links more strongly related to the target, the user can then find the target node similar to a gradient descent search. Browsing can also be useful for searching for a target not yet defined in the searchers mind, or to get a general idea of the contents available in the information space.

Traditionally, documents on computer systems are stored in hierarchal or relational structures. Information is accessed by navigation, or by searching certain properties. Yet cognitive science has provided much evidence that humans tend to store and retrieve information in an associative way, usually semantic [7]. Therefore, a different way to store information more similar to the way human memory works is to store it in an association or a semantic network. The main idea is to build an associative or semantic similarity network from the corpus of data. Each node in the information space will be connected to other nodes according to some predefined classifier function. This approach is based on the idea that information will be better accessed if it is stored and presented in a way which mimics the way we store and retrieve information.

The main goal of this project is to assist the user in navigating and investigating a large data collection using a browsing tool which navigates efficiently through this semantic network.

2 RELATED WORK

There are several approaches for visualizing a document collection. One approach is based on showing the thematic content of a document collection as a whole. These visualizations include SPIRE's ThemeView visualization[12], and ThemeRiver[4] which shows the themes of the temporal changes in a document collection using a river metaphor.

Another more common approach is based upon extracting features from the documents. These visualizations attempt to show the relationship between documents in a global view. The relationships often use proximity to convey similarity of documents. Examples of this approach are the SPIRE galaxy visualization [12], and several Kohonen SOM-based browsing tools [1].

° yoel@cs.ubc.ca

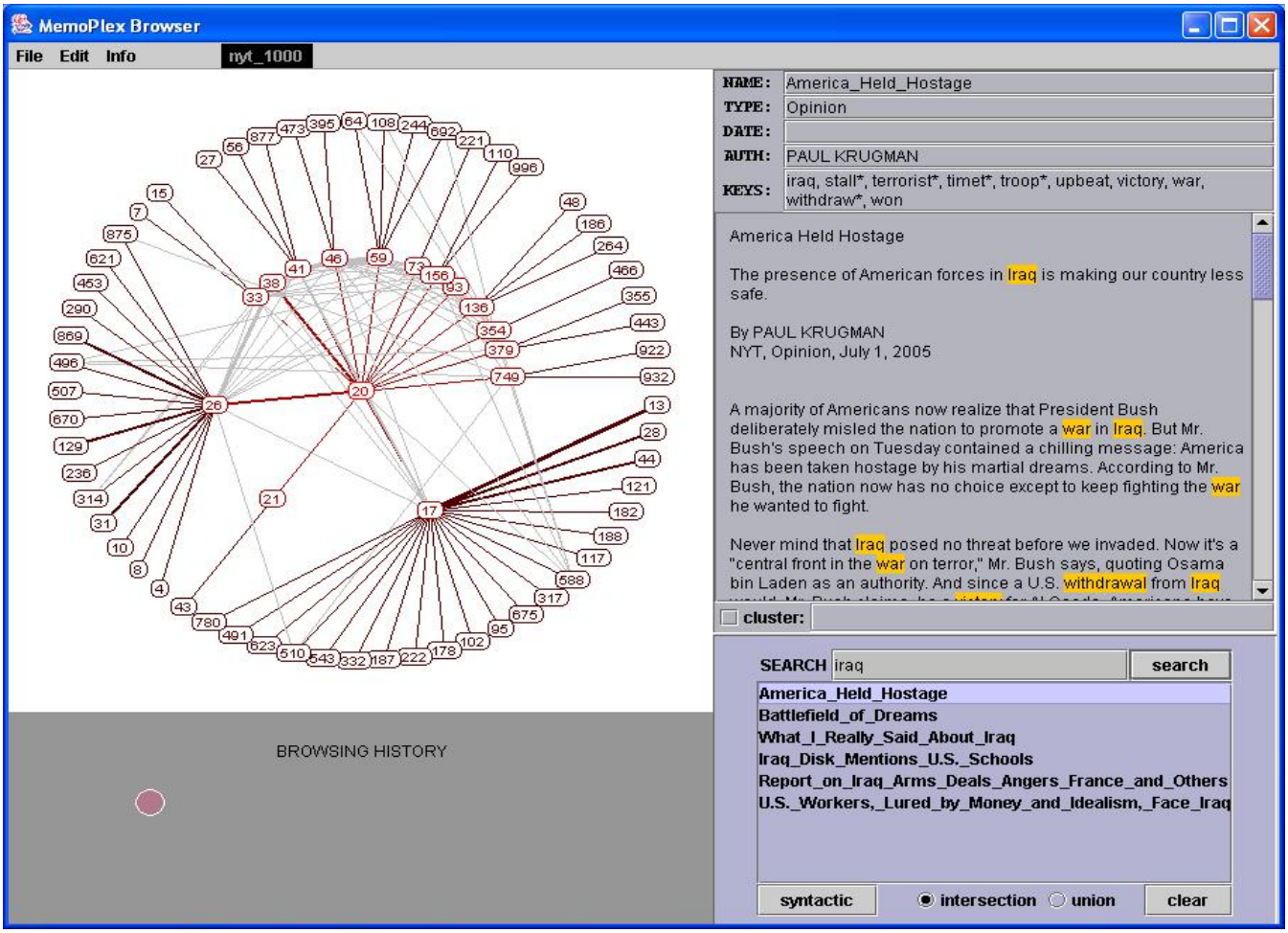

Figure 1 – An overall view of MemoPlex Browser

When showing a global view, some systems use different kind of automatic category extraction to show the categories in a corpus, and then may or may not descent to the document level. The infoCrystal tool [11] uses glyphs and icons to show document groups, and allows different queries on the information space. MindMap[10] is a system for browsing and exploring topics and concepts within a document collection, MindMap combines a global view of all documents, with a category view, down to the document level.

The common part of all these approaches is that they try to give us a global view of the document relationship and from there may or may not descent to a document level. A different approach, is to give a small view of the of the document space, allowing the user to browse through the documents. Most query based text retrieval systems use this approach. Our system is an attempt to visualize a small view of the document collection in such a way that will assist browsing through the whole collection.

3 MEMOPLEX BROWSER

MemoPlex Browser facilitates searching and browsing in a document collection based on the semantic similarities between the documents. It provides smooth and efficient browsing of documents in the corpus, while maintaining a general overview of the context of each document compared to the other documents in the corpus using clustering of documents to several groups. Figure 1 shows an overall view of the system. In the upper left corner is the Document Browsing Component(3.2), while in the bottom left corner is the browsing history window(3.4). The search window is in the bottom right corner, while the center node's data which includes the name of the article, the date, the author's name and the text is in the upper right corner.

3.1 Semantic Similarity Network

When a new corpus is added, it is first parsed by the system. A semantic similarity network is then built based on the similarities between the documents in the corpus.

When building the similarity network, each document represents a single node in the network. In addition, one must have a

classifier which gives us a measure to compare two documents. Numerous similarity measures for comparing documents have been proposed; all treat the document as a set of words, usually with frequency information, and compare the word overlap between documents [8]. *Tfidf* [9] is a simple classification algorithm which extracts human readable identifying attributes (keywords) for each document in a corpus.

We use this attribute vector as the classifier for the document domain, which assists us in building the semantic network. Nodes sharing many attributes will be regarded as similar and have strong semantic ties, while nodes sharing few attributes will have weak semantic ties. Also, the frequency of each keyword in a specific document is taken into account. Thus, the similarity score of two documents, if lies above a certain threshold, would be translated to an edge in the network with the similarity value as the weight.

3.2 Document Browsing Component

The principle component of MemoPlex Browser is the document browsing window. This window uses a radial tree view to show a graph with the document of interest in the center of the view, connected by two levels of its semantic tree neighbors as can be seen in figure 2. The user can select any node in the visible graph, which then becomes the focus node. The graph then changes using animation according to the new focus node's semantic neighbors.

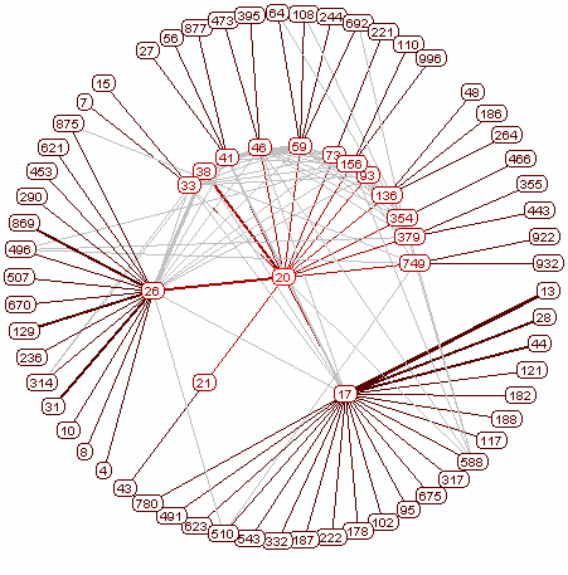

Figure 2 – radial view

3.2.1 Layout

Since we are interested in viewing the details (text, date, author) of one document at a time, in a given time we need to put the focus on one node. Furthermore, the graph showing the whole network including all the similarity connection between the different documents can be very large with many edges and presenting it as a whole to the user can be very confusing. In addition, browsing decisions are likely to be based on local neighborhood of the current node. Therefore, a radial view which puts the focus node in the center is a natural selection.

The component uses a radial view similar to Yee et al.[6] which puts a single node of interest in the center. The layout performs a breadth-first search to determine the first two levels of the tree, forming a parent-child relation starting from the focus node. These two levels are arranged in two concentric rings around the center. The children of the center node will be placed on the inner ring, while their children will be placed on the second level ring. Each node will be drawn in the same size, since each node represents a single document.

After laying out all nodes, edges between father and child nodes are drawn, and all other edges between two nodes in the display are drawn in a lighter color. Edge widths are drawn according to similarity strength between two nodes in such a way that two nodes with a strong similarity link will have a thick edge between them, while two nodes with a weak similarity link will have a thin edge connecting them. This will give the user an intuitive feeling of the similarity strength between two documents.

3.2.2 Highlighting and Animation

When hovering over a node in the graph, the user can see the node's name in a tool tip. The user can also see all the node's connected edges in the visible graph as can be seen if figure 3. He or she can then click on the node to choose it to become the new center node. The new graph layout is then determined according to the new center node as been explained above, and the view animates from the previous layout to the new layout according to the animation technique presented by Yee et al.[6]. This animation helps to reduce disorientation, and performs a smooth transition from one view to the other. The transition linearly interpolates the polar coordinates of the nodes resulting in a smooth animation from one state to another.

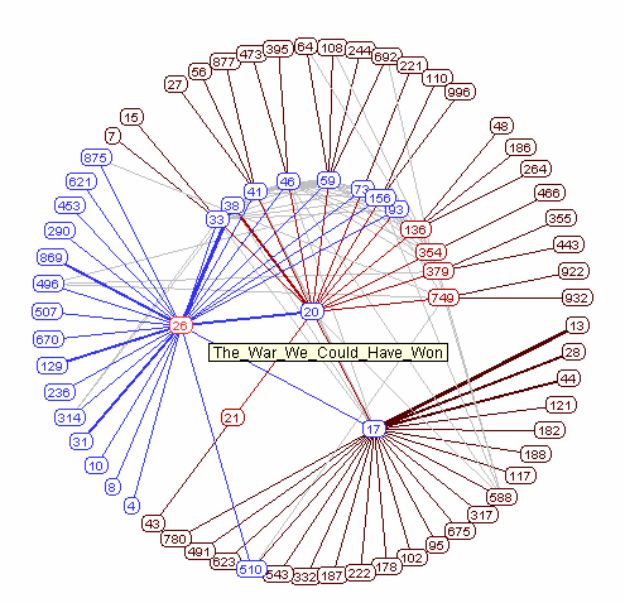

Figure 3 – radial view with mouse hovering over node number 26. the tooltip shows the node's title

3.2.3 Scalability

The component can show up to around 200 nodes without cluttering the view. The Labels of the nodes, which are the documents titles, can be read as tool tips when the user hovers over each node. Since the second level nodes are all on the same ring, above 200 nodes causes the nodes to obscure each other, causing some nodes not to be visible. To prevent this situation, a threshold value was chosen for the similarity links, in which the network will only create edges which pass this threshold value. This threshold value can be changed by the user, but not lower than a certain minimum.

One of the ways to enhance the scalability of the component is to place the second level nodes in more than one ring. The artifact of such a layout is that in contrast to the implicit notion in the radial display that distance from the center conveys network distance from the focus node, the distance from the center node varies between two nodes of the second level. It was more important in our view to keep this notion, than to add to the scalability of the graph.

3.3 Clustering

Document clustering is an important tool to enhance information retrieval capabilities. Similar documents can be clustered together into groups, improving retrieval by broadening a search request from a single document to a group of documents. This is based on the cluster hypothesis of information retrieval that closely associated documents tend to be relevant to the same request [8]. Clustering is also used in information retrieval as a method to enhance near-neighbor search.

Cutting et al. [2] used clustering in their scatter/gather tool to browse through a large database collection. They showed how it is possible to use clustering as in a table-of-contents metaphor for navigating in a collection of documents. They used clustering to describe groups of similar documents using keywords present in these documents. Their users could then select a group or more, which would be scattered to new groups for further examination. For their document clustering algorithm, they presented a partition based algorithm which can cluster a large number of documents in a short period. We will use a similar technique to cluster our documents.

3.3.1 Application of Clustering

We use clustering in the Document Browsing Component window to show the user a more global view of the area of the focus document. An example of this can be seen in figure 4.

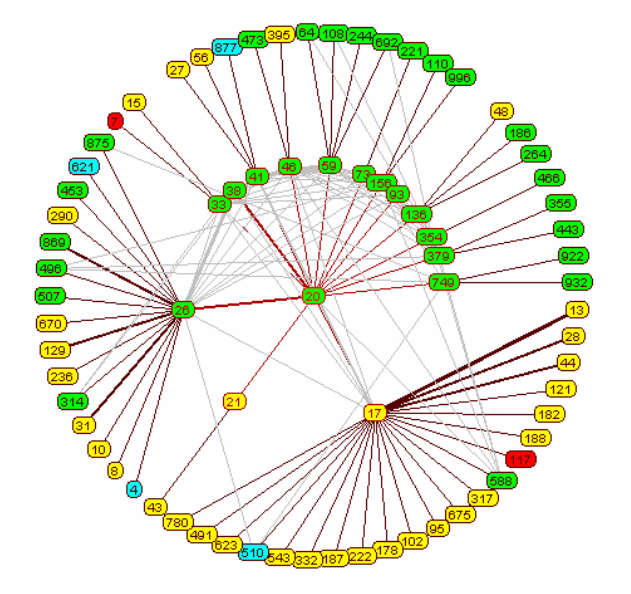

Figure 4 – An example of the clustering view. Each color shows a different cluster group.

Each cluster is mapped to a specific color, and each document is colored according to the cluster it belongs to. As we can see in figure 4, the center node belongs to the green cluster, and as can be expected most of its children, but we can see that two children and their descendants belong to the yellow cluster, and we understand that they are on the edge of the two groups. We can also see in the figure that there are two red nodes. These nodes, belonging to a different cluster, can be seen as peep holes which we can use to enter a different area of the graph.

In order to give the clusters a meaning for the user, the keyword frequency in each cluster is summed, and the 15 most frequent keywords of the focus node's cluster are presented to the user. These keywords can be used as categories of each cluster. It is usually simple to infer the category or categories of each cluster from its list of keywords. For example, in figure 4, the keywords presented to the user for the red cluster are:
student, school*, scienc*, university*, theory, univers*, tailgat*, technology, teach*, singapor*, textbook*, scientif*, teacher*, theater*, evolut**

3.3.2 Clustering Algorithm

In order to establish a clustering algorithm, one must establish a pair-wise comparison between each two documents. Then a method to partition the documents can be established. This comparison is already ready using our similarity nodes. Thus, two documents linked together will get a similarity score according to their strength. Two documents which are not linked will have a similarity score of zero.

Two different types of clustering algorithms are used in document collections. The first group is hierarchical clustering which can be defined recursively as one or more documents are hierarchically clustered. The final clustering defines a tree on the documents with the root node as the whole corpus.

The second group is partitioned clustering algorithms. They aim at decomposing the corpus into a set of clusters rather than a hierarchical nesting. Generally, these algorithms choose in some way a number of seeds – each seed representing the center of a group in the final partition. Each document in the collection is then assigned to the closest seed. The algorithm is then iterated to refine the division, and in each iteration the seed is recalculated, and all the documents are reassigned to the new seeds. The algorithm is continually run until there are no more movement of documents from one group to another.

Our algorithm is a partitioned algorithm similar to the one presented in [2]. The first part of our algorithm is to find the initial center of the partitions. In our algorithm K documents are randomly chosen as seeds. We then add to each document any document which has a link to that document and passes a certain threshold. If a certain document has links to two different seeds, both stronger than the threshold, then it will be assigned to the one which it has a stronger link to.

At the end of the first part, we have K groups, and each group has a number of documents. We then recalculate the center seeds of each group. Once K centers have been calculated, each document in the corpus is assigned to one of the groups according to the nearest center to it. We then have an initial partition of the whole corpus.

The last part of the algorithm is the refinement. Given an initial clustering, we now wish to refine this clustering into a better one. This is done by iterating the process of recalculating the center seeds for each group, and reassigning each document in the corpus to a center. This iteration can be continued until there are no more document changing groups, or until the movements are less then a certain threshold.

When comparing documents to the center of a group when assigning each document to a group, the comparison is done according to the similarity function used to classify the documents (see 3.1). The problem is that we need to find a way to represent a center for a group of documents. The center has to be similar to a single document in such a way that it will be possible to compare a document to it. In addition, in contrast to most documents which have a similarity score of zero, it has to yield a non-zero result when compared to most documents in the corpus. To calculate a center from a given group of documents, we first extracted all keywords from the documents, taking into account the frequency of each keyword in the group, and the weight each keyword has in each of the documents. Then, the most frequent and heavily weighted words in the group were assembled into a vector which represents the center of the group. The length of this vector was tested, so that all documents will be assigned to groups.

3.4 Browsing History

The browsing history of the user is important for the user for his or her browsing experience. Users might start their browsing in a specific node. Then, they may want to explore a specific area of the graph stemming from that node. After exhausting this path, the user might want to return to the initial node to browse through another area stemming from it.

In order to give the user the possibility to return to previously encountered nodes, we present to the user the browsing history window, in which it is easily possible to browse through the focused nodes from the beginning of a session. The browsing history window is presented in figure 5. The window shows the browsed nodes in a linear way, each node represented by a circle. Hovering over a node shows the full text of the nodes title. The reason the browsing history was presented as circles and not as a text list, is to keep consistency with the Document Browsing Component in which each document is represented as a circle.

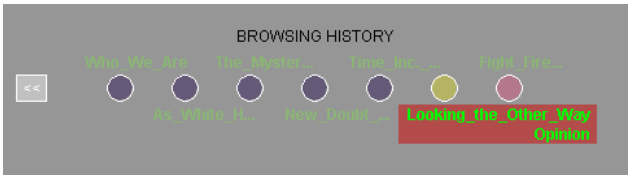

Figure 5 – Browsing History window, with the second node from the right hovered over.

3.5 Data

The data that was used when building and testing MemoPlex browser was composed of around 2000 text documents extracted from the New York Times website. The articles contexts were of different domains and included politics, sport, science and more. Each article included a title, author, date published and text. Although each article included type, we did not use this to cluster or as categories in order to handle the wider case where the type of document is not provided.

3.6 Implementation

MemoPlex Browser was written in Java with the Eclipse IDE and the Swing library. It used the Prefuse [6] library for the visualization and interactivity of the Document Browsing Component. The server component which builds and gives access to the similarity network was written by Mike Huggett from the Computer Science department in the University of British Colombia.

4 Scenario of Use

I will present here three possible scenarios where MemoPlex Browser can be of use.

4.1 Browsing to complement a search task

A user wishes to find opinion articles which talk about America's public opinion criticism of President Bush after the Iraqi war. Not knowing exactly how to formulate this query, the user types "Iraq" + "Bush" in the search window. As a result the user gets a long list of documents containing the two keywords. The User chooses the first document – "What I really said about Iraq" which is a writer opinion which justifies Bush's actions in the Iraqi war. The user looks at the documents that are one distance away from this document. He sees a document entitled: "America held Hostage". Understanding that this is probably closer to what he is looking for, he chooses this document to become the center node. This is indeed a document criticizing the Iraqi war like he thought, but it doesn't talk about public opinion. Looking at this document's close neighbors, the user finds an article called "Waking up to the war". He sees that this

article is tied up to the two articles he has browsed on, so he clicks on this article and is delighted to see that this article talks exactly about what he was looking for. Looking at this article's close neighbors, the user finds another document which talks about the same thing.

This scenario describes how MemoPlex Browser can aid in a search task. The process described here is similar to a gradient descent, where the user enters the general area, and then refines his search until finding the target he was looking for. Although it is also possible to reach the target using refinement of the search query, it may be easier to use this process. An additional advantage of using browsing for search purposes is the knowledge of the corpus's structure and content acquired during the browsing.

4.2 Browsing to get an idea of the content of the corpus

A user wants to know what kind of documents in the corpus talk about science, and how exactly is the science topic covered in the corpus. To get a general idea, the user types in "science" in the search window. The user then enters the browsing window choosing a document talking about a new science park in Florida. The user then enables the cluster view, and sees that blue color covers science, universities, students and other related keywords. She then browses through the blue documents, noting the structure of the links, the contents of the documents she browsed through, which documents are related to which, and which documents are used as hubs for other blue documents. After this process, the user has a fairly good idea of the coverage and content of the science topic in this specific corpus.

4.3 Browsing to get knowledge of a specific topic

A user wants to know more about different products that Google released. The user types in "google" in the search window and gets as a result all the documents in which Google appear. These documents include opinion articles, press releases, and technology papers. He then chooses one of the technology papers which talks about a specific product. He then browses through the articles' direct neighbors and finds some other "product" articles .

5 EVALUATION

Although we have not developed MemoPlex Browser to stand alone as an information retrieval system, since it puts the emphasis on browsing and lacks the appropriate search engine, we decided nevertheless to get some user feedback to see the system's strength and weaknesses. We therefore conducted an informal user study.

Three participants were given a brief explanation of how to use the system, and were presented to the system with the same starting point document. Participants were asked to browse the documents as they wish, and were left alone for about 15 minutes. Following the experiment session, the users were asked about their opinions of MemoPlex Browser, and were asked to elaborate on the strengths and weaknesses of the system in their opinion.

All three participants said that they liked the browsing experience. In all fairness, it is important to mention that two of the participants were friends of the writer and the other participant was his wife, so although encouraged to say what is on their mind it is possible that the answer to this question was affected by their will to please the writer. Nevertheless, the impression was that they did find interest in using the system for browsing. The strengths and weaknesses of the system as gathered from the interviews as well as the writers opinions are described next.

5.1 Strengths

1. *Provides easy to use, efficient browsing capabilities*

   This was the main goal of the system, which I believe it has achieved. The system as commented by the participants is easy to use, and provides a simple and efficient way to browse quickly through a document collection. Users can quickly browse through many documents, jump from one link to another, to get a good browsing experience.

2. *Combining searching and browsing into one interface*

   Although the search engine does not work well, the users liked the possibility of combining search and browse options in one window. The highlighted keywords in the documents gave the user further ideas on what keywords to search to get interesting results.

3. *Animation*

   The smooth transition between two states helps alleviate the cognitive load, and helps the user keep track of where he or she came from.

4. *Shows additional overview information using clustering*

   The clustering gives the user additional information which can be used to get a better understanding of the place of a specific document in to the corpus, and of the contents of the whole corpus of documents.

5.2 Weaknesses

1. *The document's title does not always tell us what is in the article*

   The titles of the documents were given as titles of articles in a website. They do not always describe the content of the article. When the user browses, he or she uses the tooltip to explore what other documents are. When the title does not describe the article, the user has to choose the article in order to see what it contains, thus loosing the context of the previous document.

2. *Unclear links between documents*

   Another comment by a participant was that there is no way to see what links one document to another. The participant commented that he wants to know what keywords links one document to another, and why the documents are related. The user commented that this information could help him know if he would be interested following that link

3. *Clustering labels are not intuitive*

   The clustering gets its meaning from the clustering keywords. The most frequent keywords in each cluster are presented and are presented as representatives of that cluster. They are supposed to be like categories, and to define the cluster's content. The problem is, that since they are automatically generated, they are sometimes not intuitive, and sometimes not too representative of the cluster's content.

4. *Scalability*

   As mentioned before, the browsing component is only scalable for around 200 nodes, above which nodes start to obscure each other.

5. *No Brushing between a node in the browsing history window, and the document browsing component.*

5.3 Lesson Learned

I have learned many things during the course of this project. Some of these are:

- Clustering is not trivial. I spent a lot of time trying to find a clustering algorithm which will effectively cluster the documents into separate groups. The main problem was, that most partitions put most of the documents in one group. After many trials, I found a solution, which managed to partition the corpus in a good way, but I still feel that a better solution can be found as mentioned later in the future work.

- Creating a layout is a very difficult task. I initially intended to build a layout which maps the distance between two documents to the semantic distance of these two documents. After struggling with this task for a while, I decided to show the semantic distance using the width of the edges instead. Now, this decision seems a better decision to me, but I learned on the way, how difficult it is to write a layout for a complex graph.

- Scalability is almost always an issue. I tried to enhance the scalability of the graph in a couple of ways during the course of this work; one of them was to use Excentric Labeling [7] to allow obscurity of nodes in a way the labels will still be readable. This solution did not work, since Excentric labeling was written in *AWT* which did not combine well with *SWING*.

6 Future Work

A number of improvements can be made for MemoPlex Browser which will make it a more complete and stronger tool for information retrieval.

The most important improvement is to add a strong search engine. The focus of this project was on browsing, and therefore, the search part was not looked as part of the project. The existing search window uses a simple inverted index on the extracted keywords, and is therefore very weak and limited to the existing keywords. A full retrieving solution should incorporate the browsing solution suggested here, with a strong search engine, which will allow the user to use the advantages of both searching and browsing.

A problem which was mentioned in the weaknesses part, is the lack of knowledge of what links one document to another. Showing the keywords on each edge could be a valuable tool for the user. This should be done in a careful manner, maybe using some sort of tooltip mechanism, not to clutter the view with text. Given the keywords, the user then may wish to filter the graph according to some of the keywords.

The clustering algorithm presented here can be further improved. I feel that the initial seed choice can be further investigated. One idea that could be explored is to use the fractionation algorithm [2] for finding the initial centers of the clusters. The metrics to measure the center of each cluster could also be further investigated.

Finally, the browsing solution suggested in this paper has a weakness, that in order to enter the document network, one must use a search query to get to an initial node. Similar to Scatter/Gather [2], a category level based on clustering can be built. To some extent, our clustering solution already does this categorization, but it does not present the user a table-of-contents like view of the corpus, and the ability to start from a category, and gradually scatter the category to different sub-categories narrowing down to the document level.

7 Conclusion

I have presented here MemoPlex Browser, a tool for browsing a document collection using the semantic similarities of the documents. The tool uses visualization techniques on top of an associative semantic network to support a human-intuitive, easy to use browsing experience. In addition, MemoPlex Browser uses clustering to effectively show the different categories in the corpus, and the position of each document in relation to these categories.

I believe that associative browsing can be effectively used in complementary with a direct search engine to give the user a complete tool for searching and browsing a document collection.

8 Acknowledgments

I would like to thank Mike Huggett for his part in implementing the similarity network server, parts of the client, and for all the inspiring ideas and conversations.

References

[1] Chen, H., Schuffers, C., and Orwig, R. (1996). Internet categorization and search: A machine learning approach. Journal of visual Communications and image representation, 7(1), 88-102

[2] Cutting, D.R., Pedersen, J., Karger, D. R. And Tukey, J.W. (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 Retrieva*l (Copenhagen, Denmark, June 21–24). ACM, New York, pp. 330 –337.

[3] J.D. Fekete and C. Plaisant. Excentric labeling: Dynamic neighborhood labeling for data visualization. In CHI '99, pages 512-519, Pittsburgh, PA, USA, May 1999. ACM Press.

[4] S. Havre, E Hetzler, P. Whitney, and L. Nowell. (2002) ThemeRiver: Visualizing thematic in large document collections. *IEEE transactions on visual and computer graphics* 8(1), 9-20

[5] Heer, J., S.K. Card and J.A. Landay (2005): Prefuse: A Toolkit for Interactive Information Visualization. In Proceedings of the Sigchi Conference on Human Factors in Computing. New York: ACM, pp. 421–430

[6] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst, Animated Exploration of Graphs with Radial Layout. Proc InfoVis 2001.

[7] Meyer, D. E. and Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90(2):227–234.

[8] C.J. van Rijsbergen. (1979) Information Retrieval. Butterworths, London, second edition.

[9] Salton, G. and McGill, M. (1983). An Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY.

[10] Spangler, S., Kreulen, J. and Lessler, J. MindMap: Using Multiple Taxonomies to Understand Large Document Collections. Proceedings of HICSS 2002.

[11] Spoerri. A . InfoCrystal: A Visual Tool for Information Retrieval and Management. *Proceeding VIS 93*, 11-20, 1993

[12] J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A Schur, and V. Crow, "Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Document, readings in Information Visualization: Using Vision to think, S.K Card, J.D. Mackinlay, and B. Shneiderman, eds., pp. 442-45, San Francisco: Morgan Kaufmann, 1999.