Brainiac: a Graph-Based Literature Visualization

Miguel Santos
miguel.d.santos@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2017

Abstract

Nowadays, users face the problem of too much information available. A user trying to research into a new topic will face a collection of context-specific documents, and exploring this collection may require knowledge on specific concepts that is only available with more experienced users. In this work, we address this problem, in the neuroscience context, creating a visualization, in collaboration with Instituto de Biofísica e Engenharia Biomédica (IBEB), that helps users analyzing a collection of documents, indicating documents that may be similar. The developed visualization has potential to help users in this context, by interacting with different views, it allows to combine a document search by similarity and by different topics. We conducted an evaluation, to measure the usability of the developed application, and its utility, to validate the data visualized. The results from the usability test were very good, with no obvious interface problem. Validation of the processed data also show good results, with room for improvement with some errors detected in text processing.
Keywords: Neuroscience, text visualization, information visualization, text processing, document collection

1. Introduction

Nowadays, users face large collections of data as they seek to understand a certain subject better. This ranges from academic research to deciding what product to buy, from a variety of options, and require users to explore a collection of context-specific text data that is often unfamiliar to them. This is one of the consequences of the appearance of the Internet, as it allows considerable amounts of information available to anyone anywhere.

While structured or numerical data is manageable through statistical analysis, text data usually contains noise and its computational analysis is much slower. In order to navigate this rich data, users often use search tools to find relevant information. However, the processing of searching a document database is not adequate in data exploration cases, as the researcher may not have the necessary base knowledge to recognize what to search for, and what keywords should be used. This exploratory search goes beyond the simple retrieval of documents, as investigating a ranked list of search results is insufficient to understand the overall collection and possible relations across multiple documents.

Several visualizations have been designed for this propose, combining both information visualization and text analysis tools. When used individually, either of these approaches yields insufficient results to adequately understand the document collection, as text mining is not considered satisfactory to comprehend the collection, while visualizations such as PaperLens start to have problems as the size of collection grows.

In the neuroscience context, a system that enables discovery could allow researchers from IBEB to freely explore a collection of documents, focusing on their topics and relations, by visualizing a collection as a whole, instead of manually browsing each one.

2. Related work

Information visualization can be seen as an additional tool to help with the process of interpreting documents. This technique can present a new representation of the collection, while highlighting possible new patterns that were not expected. It may be a reasonable approach to the described problem, by showing the user documents of potential interest, which would simplify the process of navigating the collection of documents.

Recent work has proposed different approaches on text visualization, allowing users to go over a text document and understand what it is about. They are mostly designed to visualize different aspects of a document, specifically the document's metadata, the source text directly, computed features like entities or patterns, or even the general

concepts of the document.

The work reviewed is mainly divided into visualizations that focus on analyzing single documents and visualizations for large document collections, or *corpura*. Both these categories are reviewed in the following subsections, as they provide proper contributions to text visualization.

2.1. Single Document Visualization

In order to solve these issues, different visualizations have been developed to represent the content in a simpler way, facilitating the user's process of understanding the subject of the document. Usually, these representations are based on methods that take into account the document's metadata, source text, or some computed features from text data available. Methods used to achieve this representation are usually based on metadata, the source text, or some computed features from the text data available. One example of this type of single document visualization is tag clouds, or word clouds, which utilizes the source of the document to depict the general content of the document.

Word clouds differ from tag clouds in appearance, as they are able to change the orientation and position of words in order to form a shape or figure, usually a cloud. Both these visualizations bring the user a visual representation of the text given, by displaying relevant words. These relevant words, used to depict the content of the text, are obtained through different methods, such as metadata in the case of tag clouds, or word frequency, in the case of word clouds.

Then, there is *Docuburst*[5], it allows to visualize the document content, structured according to a IS-A relation from the *WordNet*[7]. It uses a radial space-filling layout, where the hierarchy represents the hyponymy relation, while the angular width is proportional to the number of leaves in the subtree, as seen in figure 1.

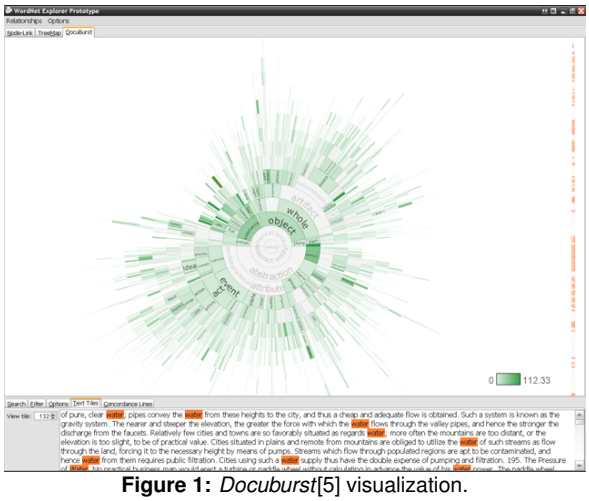

Figure 1: *Docuburst*[5] visualization.

Lastly, there is the *Word Tree*[10], introduced as a visualization focused on exploring repetitive text. It takes form in a tree structure (figure 2), with the words that follow a particular search term, arranging the words spatially.

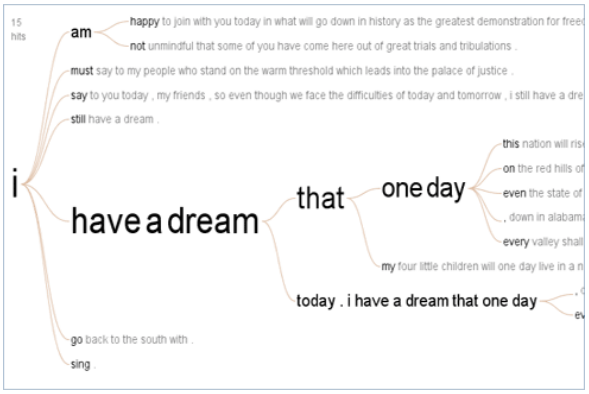

Figure 2: *World tree*[10] visualization.

2.2. Document *corpora* Visualization

When the scope expands from a single document to the complete collection, visualizations tend to be extended to a more exploratory search, while not disregarding search methods.

The simpler features that can be used are derived from the metadata. The *PaperLens*[8] system was devised to visualize trends and connections in conference papers, extracting the authors, topics and citations of these papers.

An additional example is the Bohemian Bookshelf[9], which also resorts to a visual representation based on the usage of metadata. This system is laid out as a digital book collection, designed to tackle accidental discoveries – serendipity. This is accomplished by allowing a "shelf browsing" like experience, which have been shown to inspire serendipitous discoveries. The "shelf like" browsing is attained with the different visualizations acting as a whole, offering multiple access points due to different perspectives from the views, drawing attention with the visually distinct visualizations and providing distinct, yet playful, approaches to information exploration.

The source text of the documents in the collection can be used to visualize the dataset, however, due to the large nature of the collection, this rapidly becomes unfeasible. Computed features such as word similarity or topic similarity are used to compare a likeness metric that is used to compare documents and project the differences onto a visualization, which by itself may lead to trust problems[4]. While the first similarity metric utilizes directly the source text, the second takes into account related terms used, which is convenient when the documents do not use the same exact words.

An example of this is the Dissertation Browser[4], a visual analysis tool developed to investigate collaboration between different academic departments. The adopted approach resided in detecting shared language or terms across publications of various areas, seeing that the authors mention the different vocabulary across distinct areas.

Jigsaw[6] is a separate system that provides different visual representations of computed features, that also takes into consideration the interpretation and trust concerns introduced. It produces a summary of the collection, or a single document, a measure of similarity between documents, clustering, it identifies entities and connections among them, and possible related entities for further investigation. Additionally, it allows for a document sentiment analysis, which provides insight on sentiment, subjectivity, polarity and other attributes. Some of these features can be seen in figures 3 through 6.

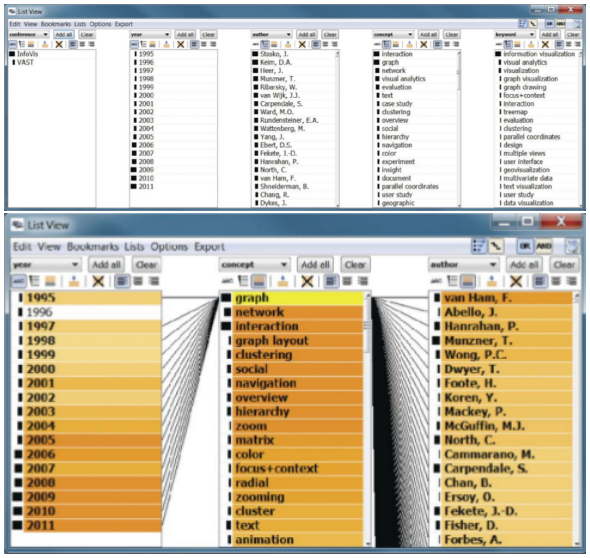

Figure 3: *Jigsaw*'s [6] *List View*. Shows the conference, year, author, concept and keywords associated. In the bottom figure, the concept graph is selected, showing connected years, concepts and authors.

3. Proposed solution

Brainiac is an application focused on visualizing a collection of documents. It is a tool developed in collaboration with IBEB, to help users explore the content of a group of documents, allowing the user to potentially identify documents of interest, arranging documents based on their similarity and topics.

The development of this visualization followed an iterative and incremental development, focusing on user feedback to improve its usability and main features. As such, there were two main testing phases in this process. An informal testing phase, where the focus was gathering feedback from the users, and a formal one, aiming to measure the usability of the final version of the application.

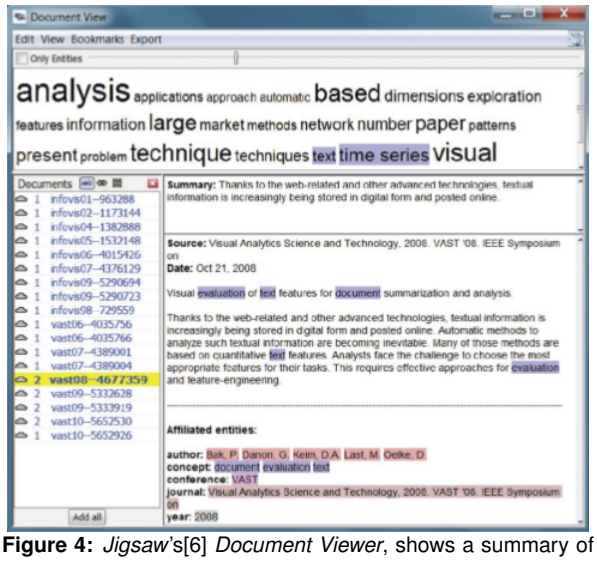

Figure 4: *Jigsaw*'s[6] *Document Viewer*, shows a summary of the loaded documents(left panel) at the top, and summarizes the selected document on the right in the **Summary** panel.

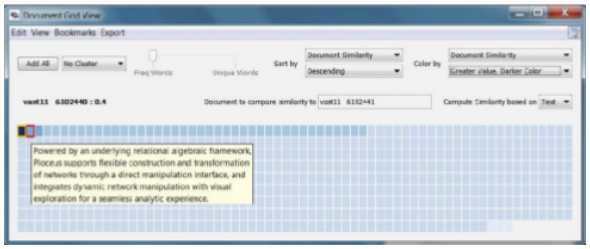

Figure 5: *Jigsaw*'s[6] *Document Grid Viewer*, displays the documents in a grid, ordered by similarity according the selected document.

3.1. Backend document processing

The pipeline of document processing involves the conversion of the collected Portable Document Format (PDF) files to plain text, as these files are stored in binary. This conversion creates some artifacts in the resulting text, due to different encoding in different files. This is solved with a set of rules that filter out unwanted characters.

These plain text files are then converted into a bag-of-words representation. The tokenization process splits the text into its words, and uses a lemmatizer to reduce words to their base form, with the help of the *Wordnet*.

The bag-of-words representation is fit into a **tf-idf** model. Then, the resulting matrix is used to measure similarity between documents across the collection, by calculating the cosine similarity between document's vectors. This yields a similarity matrix, used to identify similarity relations in the database. The resulting tf-idf matrix is fit into a *k*-means model, in order to complement the similarity relations.

Figure 6: *Jigsaw*'s[6] *Document Cluster View*, displays the different clusters of similar documents.

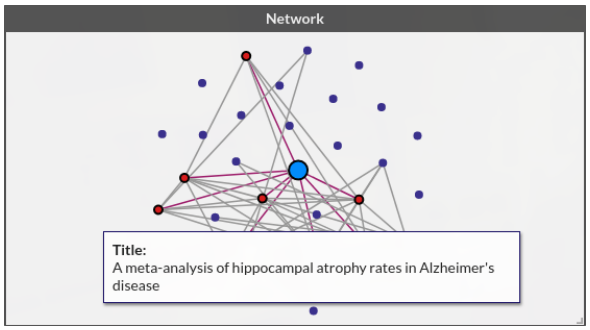

Figure 9: Example of hovering in the *Network* view. Hovering works similarly in the remaining views, except it does not change the node's color in the *Cluster Layout*, in order to preserve the cluster original color.

3.2. Brainiac: a Graph-based Literature Visualization

The main visualization, in figure 7, is divided into three different views: the *Network*, the *Cluster Layout* and the *Timeline*. Then, the sidebar provides with a set interactions that complement the use of these views, together with the topic magnets feature that allows user to search topics in the text.

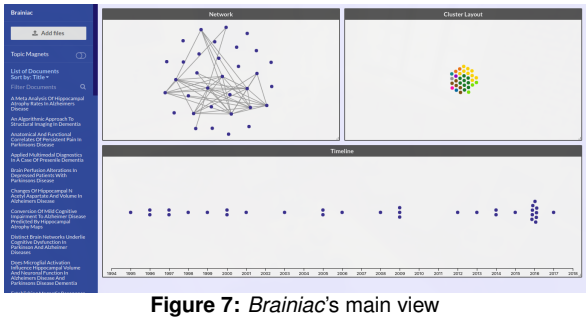

Figure 7: *Brainiac*'s main view

First, the *Network* visualization focuses on showing the user the documents in the collection, as nodes, and their similarity between each other as links between nodes, which can be seen in figure 8, with these being computed as described in the previous subsection. The *Cluster Layout* displays the documents color coded by the cluster they belong to. Finally, the *Timeline* places each document according to their publication year.

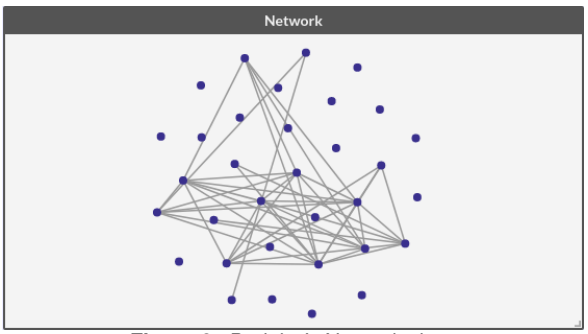

Figure 8: *Brainiac*'s Network view.

By hovering a specific document, all the nodes representing the document in the remaining visualizations become highlighted, as seen in figure 9. It is also possible to focus a document, which works similarly to the hover interaction, but leaves the mouse available for further exploration.

Both the *Network* and the *Cluster Layout* support the ability to zoom, showing a greater amount of detail in the visualization. By zooming out in the *Cluster Layout*, users can collapse nodes into their corresponding clusters. The *Timeline* allows the user to filter documents based on their year of publication, which will gray out nodes that were published outside to the selected period.

In the sidebar, a document list displays all the documents present in the collection. This list allows getting details on each document, and to search for a specific title, using the list filter. The sidebar also contains the Topic Magnets feature. It allows users to create a magnet with a specific word, that when placed in the *Cluster Layout*, attracts the most relevant documents, as seen in figure 10.

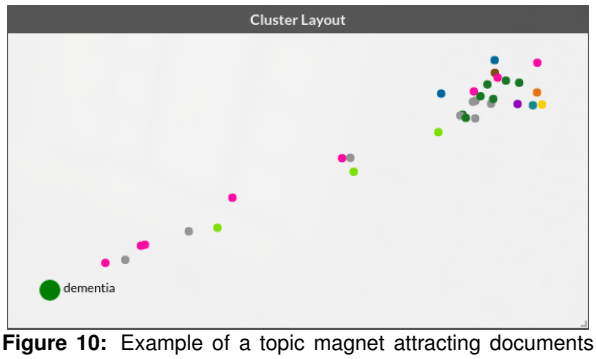

Figure 10: Example of a topic magnet attracting documents based on their relation with the topic.

4. Evaluation

Subjects were recruited through standard procedures including direct contact and through word of mouth. Subjects included anyone interested in participating if they were at least 18 years old. Each participant was asked to sign a consent form.

During this testing phase, 16 tests were per-

formed. In the first test, technical problems prevented the first user from performing the last task. All the tests were conducted between 08h30 and 20h00. None of the subjects that completed the test had professional experience in neuroscience. However, since the test focused on the interface aspect of the visualization, this did not impact the tests.

The tests were performed in a laboratory inside *campus* Alameda, in Instituto Superior Técnico (IST). Each participant was explained the purpose of the study, and what they would be doing. Subjects were asked to fill a consent form, to allow the recording of their actions in the visualization during the test.

After filling the form, users were explained the meaning behind each of the visualizations presented in the application, namely the *Network*, the *Cluster Layout* and the *Timeline*, as well as the interactions between each one. After this brief summary, participants were given **5** minutes to explore the application's interface, experimenting the described functionalities. At the beginning on this phase, a simple *script* was run to start recording the user's actions on screen, for later reference.

Following this exploratory phase, subjects were asked to perform a series of predefined tasks, which the assistant would measure, in terms of time taken to perform the task, and the number of errors showed during the execution. Participants were given a single task at a time, given the next one when the current was completed. The list of predefined tasks are as follows:

1. Identify the year with the most publications;
2. Identify one of the documents that has the most relations in terms of similarity;
3. Identify one of the biggest clusters of documents;
   (a) Give example of two documents belonging to that cluster;
4. Filter documents between 2000 and 2010 and identify two documents belonging to that time span;
5. Identify the year of publication of the document named "Distinct Brain Networks underlie cognitive dysfunction in Parkinson and Alzheimer diseases";
6. Center the network visualization on the document named "Regional volumetric change in Parkinson's disease with cognitive decline";
7. Give two examples of documents belonging to the same cluster as document named "Structural Brain Changes in Parkinson Disease With Dementia";
8. Give two examples of documents that are related to "Temporal lobe atrophy on MRI in Parkinson disease with dementia";
9. Zoom out the cluster view;
   (a) Identify the most recent cluster
   (b) Identify a cluster disperse along the timeline;
10. Create a new Topic Magnet with "Alzheimer";
    (a) Identify two documents related to the topic;
    (b) Identify the closest cluster to the topic;
    (c) Create a new topic magnet with "Parkinson" and place it on the opposite end of the previously created magnet;
    (d) Identify the closest document to the new topic;
11. Upload the given document and update the visualization with the new added document;
    (a) Center the network on the new document, and identify two documents related;
    (b) Identify the cluster the document belong to;

With the completion of this set of tasks, users were then asked to fill the System Usability Scale (SUS) questionnaire. The SUS was utilized to measure the application's usability, and consists of a ten item questionnaire, using a Likert scale to give an overview on how the user felt about the system[3]. Then, testers were given a compensation based in candies and thanked for the time taken.

5. Results & discussion

The fourth task required users to apply a filter in the timeline and identify documents. The box plot for this task shows a more compressed distribution of the time taken, with only two users with an error reported in the completion of this task.

Tasks 5 through 8 required users to search for a specific document and make the same observations as the first group of tasks. As such, the first task of this group, task 5 had a very disperse distribution, with almost all users getting at least one error in this task. Contrary to the first task, the rest displayed a more compact distribution, with less users displaying errors.

Tasks 9, 9a and 9b required the user to combine the *Cluster Layout* and the *Timeline*. The first simply required the user to zoom out on the *Cluster Layout*, and as such, it does not present a sparse distribution, although there were still a few errors. Then, 9a presents a more disperse distribution, with 9b being denser in the box plot.

The next group of tasks, 10a through 10d, including task 10, required users to work with the *Topic Magnets*, in the sidebar. These tasks presented a denser execution time distribution, although there were outliers that did not understand the task at the beginning. Only a few users showed errors in the execution of these tasks, and overall, the distribution of the task time is compact.

Lastly, tasks 11 to 11b also displayed less variation in the spreads, with errors detected only on the first two tasks.

From the SUS questionnaires, the usability was measured with a mean score of **82.5** (see figure 11), across all users, with a standard deviation of **9.287**, indicating that results do not vary too much from the mean. Research indicates Web-based SUS scores to be, on average, **68**[2]. Since the score in this testing phase reached an above average score of **82.5**, with a low standard deviation, it can be concluded that users were satisfied with the usability of the application, apart from the identified errors.

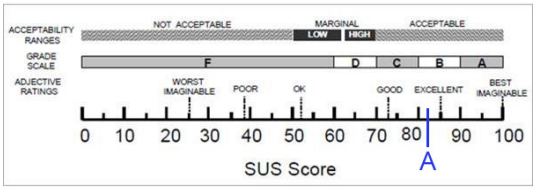

Figure 11: A comparison of the adjective ratings, acceptability scores, and school grading scales, in relation to the average SUS score[1]. The questionnaires place this visualization at **82.5**, marked **A** in the figure.

In general, the results were very good. The execution times presented were low, with only simple errors being made when users did not understand a specific interaction right away. The results did not point to any obvious usability problem, although the analysis pointed to UI elements that required subtle changes to improve the user interface.

The first group of tasks, tasks 1 and 3a were completed without problems, however, task 2 and 3 have a wider spread, that could be attributed to the wording in the requested task.

In the second task, users were asked to identify one of the documents that displayed “the most relations. Some users tried to determine the one with the most, and were unsure what to pick, which caused the larger spread on that task. This occurred in the second task similarly, which lead to some users completing in just a few seconds, while other users tried to compare each cluster’s number of elements.

As mentioned, the fourth task required users to filter the *Timeline*. While this task did not present a significant problem, it caused the large spread on task 5. This happened due to this task requiring the user to filter the timeline. Since the filter is applied not only the visualization, but to the document list on the sidebar, many users did not remember to, at first, remove the filter from the timeline before searching for the required document. Other users also searched the required document by trying to scroll through the list, without using the document filter. This could be attributed to some users not noticing that they could search the document list by typing the name of the document, as the input box may not be obvious on a first look.

The rest of the tasks that required the user to identify a property about a specific document did not have such a large spread, as users had already removed the filter. However, some users noticed there was a bug on the document filter, that did not match any documents if the query started with a Lowercase letter. This flaw was not obvious at first, and lead to some users to search the list by manually scroll the list looking for the needed title.

Tasks 9a and 9b required the user to hover each cluster in the *Cluster Layout* and follow each one’s spread on the timeline. However, some participants did not understand they could exploit the hover interaction to quickly identify the solution. Some tried to manually scan the timeline and identify which cluster that document belonged to, and tried to estimate the answers, which lead to high execution times in those cases.

The group of tasks that involved creating new topic magnets did not present any significant problem. The times measured in tasks 10 and 10c include the time needed for the preprocessing required in the backend, which normally added around 20 seconds to completion time. Due to technical problems that two users faced with the preprocessing, they repeated the task, although there was no significant improvement that could skew the results.

The last group of tasks involved the upload of a new document to the visualization. One of the users also experienced technical problems in this task, and due to time constraints, did not perform any of the tasks in this group. The first task, 11 also included the time required to upload and process the new document, which leads to displayed higher execution times. Most users identified a problem with the interface with the file uploader, as after filling the required details to upload the document, they did not understand where to proceed with the

upload. As such, the positioning of the button was changed to improve clarity in the process of adding new documents to the collection.

The rest of the tasks in this group, tasks 11a and 11b did not present any significant findings, as users simply had to repeat tasks on the new document.

In conclusion, following participants completing the list of tasks given lead to finding some subtle problems with the UI, such as some elements not being as highlighted, like the document filter on the sidebar. There were some other problems that were promptly identified by users. One example was already mentioned, the positioning of the upload button in the file uploader interface, but two users mentioned that there could be some trouble identifying cluster colors on the timeline, when comparing a darker green with the default node color.

6. Conclusions

Nowadays users face the problem of too much information available. A user trying to research into a new topic will face a collection of context-specific documents, and exploring this collection may require knowledge on specific concepts that is only available with more experienced users. With that, different visualizations were reviewed. These tried to help users understanding the content of a single document or making sense of the whole collection of documents, usually helping the user what kind of topics are available in the visualization. Combining the comparison between the reviewed visualizations with the gathered requirements from professor Hugo Ferreira, from IBEB, a list of tasks were derived that helped guide the development of the application. The development followed an iterative model, that relied on the feedback collected from the users to improve the visualization's usability. An informal testing phase took place, in order to gather feedback and detect possible usability problems before the final usability tests. Finally, a formal testing phase took place, which consisted on two phases: usability tests and case studies. The former focused on measuring the usability of the application, while the former aimed to validate the utility of the developed solution.

Future work involves improving the backend text processing, by using bigrams and trigrams. By using these contiguous sequences, text analysis will be able to take context into account, when measuring similarity between documents. This could be sued to solve the wrongly linked nodes in the *Network*, and improve the existing connections. There could also be some further work to improve the method of adding new documents to the visualization. This be the integrating the visualization with a search engine such as *Pubmed* or *Google Scholar*, with a procedure to automatically fetch the full document. On the other hand, another method would be to allow the drag and drop of files into the visualization, with automatic fetching of metadata, from the file or from an online database, removing this concern from the user.

References

[1] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[2] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[3] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[4] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 443–452. ACM, 2012.

[5] C. Collins, S. Carpendale, and G. Penn. Docuburst: Visualizing document content using language structure. In *Computer graphics forum*, volume 28, pages 1039–1046. Wiley Online Library, 2009.

[6] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1646–1663, 2013.

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

[8] M. Spindler and R. Dachselt. Paperlens: advanced magic lens interaction above the tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, page 7. ACM, 2009.

[9] A. Thudt, U. Hinrichs, and S. Carpendale. The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1461–1470. ACM, 2012.

[10] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *IEEE*

transactions on visualization and computer graphics, 14(6):1221–1228, 2008.