TAMPERE UNIVERSITY OF TECHNOLOGY

Apurva Nandan

Exploratory Search Using Interactive Visualization Techniques

Master of Science Thesis

Examiner: Prof. Moncef Gabbouj
Examiner and topic approved by the Faculty of Computing and Electrical Engineering
on 8 April 2015

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
NANDAN, APURVA: Mr.
Master of Science Thesis, 55 pages, 5 Appendix pages
May 2018
Major: Pervasive Systems
Examiner: Prof. Moncef Gabbouj
Keywords: Information Retrieval, Information Visualization

This thesis studies exploratory search of large datasets using machine learning and human computer interaction techniques. It is often that the user wants to search for something but cannot formulate the exact query or the keywords which would help him to reach the most relevant search results. One might also want to address these issues about a particular topic by gathering information after each search. We worked on extending an existing interactive exploratory search system known as SciNet. SciNet allows the users to provide relevant feedback to the system using interactive user interface. The system allows the users to direct their search query using interactive intent modeling. The users obtain relevant results by giving personalized feedback to the system through a radar based layout. Our aim is to make SciNet work for a large news data set and add new features which allow the users to explore and investigate the news articles. We try to visualize the entire collection of news articles stored in the system using neighborhood embedding and display it as an interactive map to the user. The locations of the search results are displayed on the map using markers. The users can explore the articles by clicking on markers. They can also select areas of the map where search results are located, which would enable them to view a list of most relevant unigrams in an area and are able to select relevant unigrams to boost the query. This serves as an additional feedback mechanism. We performed user experiments with twenty users to compare the performances of the original SciNet and the new extended system. The user experiments clearly showed that the extended system performs better than the original one. We also took feedback from the participants of the experiments in a form of a questionnaire, which showed that the extended system improves the overall user experience. We can further improve the performance of the new system by adding more features like tagging different regions of the map with descriptive keywords and using distributed computing based algorithms which would allows us to incorporate more data from different domains.

PREFACE

This thesis focuses on a new and emerging topic in the area of Information Retrieval. It shows how we can use visual aids to augment a traditional search engine. The thesis has been done under the supervision of Prof. Jaakko Peltonen (Faculty of Natural Sciences, University of Tampere and Department of Computer Science, Aalto University) in Aalto University as a part of Helsinki Institute of Information Technology. Jaakko has been pivotal with his guidance throughout the thesis and his expertise on exploratory search was immensely valuable in the research. His vast pool of knowledge has enabled me to approach research based tasks more constructively.

Next, I would like to acknowledge the role of Revolution of Knowledge Work (ReKnow) project for providing all the infrastructure that was needed. I would specially like to thank Han Xiao for providing the MauI keyword extraction results which we ultimately ended up using.

I would also like to express my sincere gratitude to the examiner Prof.Moncef Gabbouj from Tampere University of Technology for guiding me with the whole thesis submission process.

Special thanks to my friends from my office room in Aalto University, who always kept me motivated.

Finally, I would like to thank my mother who has always been my pillar of support. She believed in me in times when nobody else did. Without her unbridled love and care, I wouldn't have been what I am today.

APURVA NANDAN
apurva.nandan89@gmail.com

CONTENTS

TERMS AND DEFINITIONS

IR	Information Retrieval
HCI	Human Computer Interaction
NLP	Natural Language Processing
SciNet	The interactive search system used in this work
InfoVis	Information Visualization
Radar	Intent Radar
Map	Global Visualization Map

1. INTRODUCTION

In this modern era of information technology, the user is continuously seeking new ways to extract information for his purpose. There's a need to develop and find novel approaches that help in getting useful information from a large pool of data. One of the major concepts for extraction of relevant information is known as Information Retrieval (IR). IR has existed even before the world wide web. It was used internally by many companies to extract relevant data according to their desired needs. Such a company typically had a database system and could apply database mining techniques to seek the information in some way, using different extraction tools available to them. Mostly, the people who carried these tasks had to be trained on these tools and gradually became experts in retrieving the data. With the emergence of the internet, there has been a huge influx of data from all around the globe, which has made fetching of data even more challenging. The scenario involving few users extracting the information and presenting it, has also evolved. In the present time, everyone who has access to the internet may have needs to explore the web and retrieve the data according to their needs.

Active internet users rely on using search engines to carry out their work or to gain some information which might be required either for personal usage or in a professional capacity. Thus, it is evident that the users would utilize the search engines for seeking information and would then need to formulate queries for carrying out their search. A normal user is not trained at searching using specific tools or using programming techniques, so he would rely heavily on the interface presented to him and performance of the backend to get the relevant results. This new scenario has changed the way how IR systems are designed now. The IR systems being designed are made interactive so that even a normal user can use it to search for the information easily. In some cases, even if the user understands the interface, it can be difficult to formulate the correct query for searching. Interactive systems help users (both expert and normal) in cases where users need to "learn as they search". All IR systems could be evaluated with respect to the user's performance and satisfaction with the results after each query. The difference with interactive IR systems is that they can gather feedback from the user and try to improve the goodness of results during the search process. The role of Human Computer Interaction in an interactive IR system thus becomes very important from this perspective and is also

a major area of research these days.

We work on an emerging topic called Exploratory Search where the user is not sure about the things he need to search for in the first place or he might not have a clear view of how to approach the search process in order to get relevant results back from the system. In a traditional system, either the former or the latter leaves the user with no option but to spend a lot of time on searching which could be done more efficiently, if the system provided more support. The term exploratory lays emphasis on the fact that we need to investigate and analyze the results in order to reach to the desired information. The user also learns about different topics in the data while exploring it. In exploratory search we need to combine the querying process and browsing strategies to boost the exploration to return the most relevant results back more efficiently.

The exploratory search in itself is a complicated process and we need to think from the user's point of view to design the best possible system which helps him in reaching his goals. It is generally quite hard for the users to formulate queries if he is not sure of the domain and goals of his search. As each iteration of the search takes place, the user tends to gain more knowledge about his goals, and hence his information needs evolve. Adapting to this kind of a situation can be complex, as the user who is starting his search provides a quick initial search query which might not lead to the final end result which he wants. Directing the exploratory search with the help of intelligent systems then comes in which lets the user take control and direct his exploratory search as he gains more information about his specific needs. The system should be incorporated with interfaces designed to provide people versatile opportunities to control the search process. A human analyst and the search system take turns in providing information to one another, for the purpose of improving the retrieved search results. It's quite evident that searching, investigating search results, and learning about the search topic over time is a continuous process which requires significant human effort. To support the exploratory search, IR systems need novel methods from HCI research allowing the human more efficient detailed ways of giving feedback and controlling the search system.

1.1 Research Goals

This thesis describes work on the exploratory search domain where novel interfaces are developed to help the user direct his search. The aim of this thesis is to work on creating an extension of an ongoing research project called SciNet. SciNet is based on the works of Ruotsalo et al.[1][2][3], Kangasrääsiö et al.[4]. It uses a novel concept called interactive intent modeling to perform exploratory search, where the user can provide feedback using a machine learning based visual interface called an Intent Radar. We work on augmenting the existing system of SciNet by adding

a new feature to the system which would let users explore a large corpus of news articles as a whole and allow them to send feedback in a new way.

Our aim is to build an interactive interface by generating a global visualization of the whole dataset. The global visualization is based on a nonlinear visualization that preserves neighborhoods between news articles.

The work had the following goals:

1. It was desired to enhance the ability of the user to browse all the news articles using a global view. The user interface was augmented with a Global Visualization Map, which is divided into a grid 20x20. The user is able to click on each grid and view a list of top keywords/unigrams present in that area. To accomplish this feature, we generated a plot which was added to the user interface. The plot was generated using the following techniques. At first, a neighborhood matrix was created between documents by calculating the approximate nearest neighbors of a particular document. The neighborhood matrix generated using nearest neighbors is then used as an input to project the data points from original space to a lower dimensional space. This is done with the help of scalable neighborhood embedding for t-SNE methods using the method proposed by Z.Yang et al.[41]. The embedding aims to preserve neighborhood relationships between data points from a high dimensional space to a lower dimensional space. The created visualization provides a global view of the whole data, which has local regions or clusters. The clusters signify the set of documents which are grouped together because of their similar content.

2. To help users explore the Map to gain a more comprehensive understanding of the documents, we draw markers over the Map to represent the locations of our search results. The users can click on the markers to see the important information like title, URL, abstract and keywords of a document. This allows them to explore a specific region by seeing the similar articles present there. It also helps in recognizing various news events of a given topic, when the users see the search results in different clusters instead of viewing them as a long list of articles. The user can also choose to send feedback by clicking on the keywords beneath each of the documents.

3. The user interface currently contains two different components that help users to navigate the documents - the Intent Radar and the Global Visualization Map. To relate the keyword information shown on the existing Intent Radar component to the information in the new Global Visualization Map, we added a new feature in which the unigrams relevant to our search query are colored according to the colors of the corresponding keywords in the Radar. The grid cells in the whole grid based layout are also colored using color strengths of the

unigrams present in that grid. The transparency of a grid cell is determined using the number of unigrams matching with the keywords present in the Radar. The color and transparency provide an indication to the user about the occurrences of matching unigrams in a particular grid cell.

4. It was needed that the new Global Visualization Map should provide an additional way of sending feedback to the system. The original SciNet system has a feedback based system which works using the Intent Radar. In the Map, we can select the relevant unigrams from the list which is displayed upon clicking a grid. The unigrams can be sent as a feedback to the system. Using the feedback one could investigate further in a particular topic.

The features added to the system provide the users an option to search within a large corpus using typed queries and keyword manipulation.

2. THEORETICAL BACKGROUND

2.1 Information Retrieval

The research in this thesis is based upon the principle of Information Retrieval (IR) and it plays a major role in most the work accomplished in the thesis. IR is defined as the activity which aims at extracting information relevant to a particular search query from a large pool of information sources. This thesis concerns search within corpuses of text documents such as news articles. Searches for such a corpus would generally be based on upon the meta-data contained in the page or the document, or it can be the full text content of the whole document. The meta-data could be the keywords related to the article, title of the article, or information about author who published the article. This research will mostly work with the full text content of the document. Since the research concerns a search engine, IR is an important theme of this work.

2.2 Models of Information Retrieval

This section discusses the two most popular models used in IR - the Boolean Model and the Vector Space Model.

2.2.1 Boolean Model

The first model is the Boolean model which is used by traditional IR systems. The Boolean model is formed by the help of user queries which are formulated into Boolean expressions such as "(bioinformatics OR biotechnology) AND classification". The user queries have to be given as input to the system which it interprets and proceeds to determine the results based on the Boolean expressions obtained. There could be several ways to provide the user queries as input. It could be, for example, through check-boxes or radio buttons in the user interface or by simply typing a text phrase which is interpreted into a Boolean expression.

2.2.1.1 Advantages and Disadvantages

The advantages of using the Boolean model:

- It provides a relatively simple way for providing the user input to the system

and fetching the results back. This is simply done by getting the documents which satisfy the given boolean expressions.

- The simplicity means ease of use for the developer as well as for the user. This can be a good starting point, if one needs to understand the basics of searching mechanisms.

On the other hand, some of the drawbacks of the Boolean model:

- The results generated by this type of model are typically not ranked, because documents are considered to either satisfy the Boolean query or not. Such a model does not provide an indication of what are the most relevant documents documents in the results obtained.
- The basic Boolean model does not support weighting of the search terms in the query and the documents which may lead to poor-quality results since even a low-weight occurrence of a term in a document would be considered a match.
- The results cannot be obtained on the basis of partial matches, since documents that do not match any term required by the Boolean query do not satisfy the whole query. Boolean queries may therefore leave out documents that are only near matches to the Boolean query but would still have been considered relevant by the user.

Instead of the Boolean model, it would be preferable to have an alternative retrieval model that could provide a ranking of documents in order of their relevance and would therefore avoid the problems present in the Boolean model. The procedure for calculating the relevance scores is explained in later sections. The ranking of the documents in order of relevance makes the searching experience better for users. In such a model, the search engine would compute a score for each document representing how well it matches a given query.

In such a model, a search query typed on a web interface would be interpreted as a free-form set of words without Boolean operators. The intuition is that the greater the proportion of query terms present in a document, and the greater the frequency of the terms in the document, the more relevant the document should be to the user. The overall score of a document could then be computed, for example, as a sum of scores over individual query terms. The next subsections 2.2.2-2.2.3 detail a simple example of such a retrieval model, known as the Vector Space Model which aims to overcome the drawbacks of the Boolean Space model.

2.2.2 Vector Space Model

Instead of a Boolean model, a popular alternative model to represent documents and queries is the vector space model [10]. In the vector space model, a document can be

represented as a document vector where each element of the vector represents how strong a particular document feature (term) is in the document. The strength of a feature in the vector can be either directly proportional to a count of occurrences of the feature in the document, or a more complicated equation as discussed in the next section. Given a common set of possible features, the possible feature combinations then form a vector space, and a set of documents can be represented as vectors in it.

We represent the documents and the query using the Bag-of-Words approach, where word order is not considered and features represent the occurrence of individual keywords throughout the document. The query string is treated as a very short document. A vector representation is then computed for each document and the query. We then compute the cosine similarity measure between the documents available in the document corpus and the query vector. The cosine similarity is computed using the following equation,

$$cos\theta(\overrightarrow{d}, \overrightarrow{q}) = \frac{\overrightarrow{d} \cdot \overrightarrow{q}}{\left\|\overrightarrow{d}\right\| \|\overrightarrow{q}\|}$$

Where, $\overrightarrow{d}$ is the document vector which contains term frequencies or importance of terms as features, $\overrightarrow{q}$ is the query vector which is representation of the query string in document vector form, $\cdot$ is the inner dot product and $\|\|$ is the euclidean vector norm.

The value that we obtain using the cosine similarity is the score for the documents corresponding to the given query. The larger the score, the more similar the document is to the query. Using the scores, we can rank the documents in order of relevance.

2.2.3 Term Frequency - Inverse Document Frequency

Term-Frequency - Inverse Document Frequency (TF-IDF) is a term weighting method used in the areas of information retrieval and text mining applications. The weight given by TF-IDF is a statistical measure, which is used to compute the importance of a particular word in the given corpus. The importance of a word for a particular document is proportional to the frequency of the word in the document but is controlled by its frequency in the whole text corpus [6]. Thus, the importance increases if a word occurs many times in a given document but would decrease at the same time if it is common across the whole corpus.

Generally in text mining applications, A TF-IDF matrix is built which contains TF-IDF scores for all the unique terms occurring across all the documents in a

corpus. The following is a sample TF-IDF matrix which shows 'n' documents $d = \{d_1, d_2, ...d_n\}$ and 'm' unique terms $t = \{t_1, t_2, ...t_m\}$

$$\begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \dots & t_m \end{array} \\ \begin{pmatrix} 0.25 & 0 & \dots & 0.18 \\ 0 & 0.54 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0.34 & 0.12 & \dots & 0.08 \end{pmatrix} \end{array}$$

Each element corresponds to the TF-IDF score for a particular term in a specific document.

2.2.3.1 Calculation of scores

The Term Frequency (TF) in its natural notation or raw form is just denoted by the occurrence of a specific term in a document

However, a term which occurs 50 times is unlikely to be fifty times as important than a word which might occur just one. One of the techniques to attenuate the impact of terms occurring frequently, is sublinear TF scaling [7] which has been known to improve the results significantly[8]. Mathematically, it is given as,

$$TF(t,d) = \begin{cases} 1 + \log(tf_{t,d}) & \text{if } tf_{t,d} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Where, $tf_{t,d}$ is the frequency of a term t in document d.

Next, we calculate the Inverse Document Frequency (IDF) which measures how important a term actually is. It weighs down the frequent terms and boosts the rare terms across the corpus.

$$IDF(t) = log_e \left(\frac{N_D}{N_D^t} \right)$$

Where, N_D is the total number of documents present in the whole corpus, N_D^t is the total number of documents with the term t in the whole corpus

Once we have both the TF and IDF , the final TF-IDF score for a particular term in a specific document is given as,

$$TF - IDF(t,d) = TF(t,d).IDF(t)$$

In systems which have varying lengths of the document vectors, short documents would have short vectors and the longer documents will be represented by a larger vector. To eliminate the bias towards longer documents which would have more chance of getting retrieved than the shorter ones, we use a normalization factor to

equalize the length of the document vectors. This is also called L2 Norm.

If w is the TF-IDF weight of the term t in a document with n terms, then the final weight or score for a specific term would be given as,

$$w/\sqrt{\sum_{i}^{n}(w_i)^2}$$

The TF-IDF techniques have been employed in many applications , most notably in searching applications where the relevance of a document is determined with respect to a query from the user.

We have used the scikit-learn [9] python library to calculate the TF-IDF scores as one of the steps during the generation of the Global Visualization Map. The library helps us in building a clean and robust pipeline for all the natural language processing work, which is needed for generating the Map. The equations for IDF are slightly different in this case.

For calculating the IDF, the numerator and denominator of the logarithmic fraction is added with a constant value of "1". By doing this, it appears as if the whole collection had an extra document, which contained every term of the corpus,exactly once. This is done to prevent the divisions by zero. Also, constant "1" is added to the equation in the end, which means that the terms that have the IDF value as 0 (i.e. for the terms occurring in all the documents), will not be getting ignored completely.

$$IDF(t) = log_e\left(\frac{1+N_D}{1+N_D^t}\right)+1$$

2.2.4 Keyword Extraction

Keyword extraction is a technique which is primarily used to represent the important themes in a given document using different phrases or keywords. We used Maui[23] to automatically extract the relevant tags for a particular document. Each tag represents a specific keyword, and the set of keyword tags for a particular document are together intended to describe the topical content in the document. Maui is an algorithm which which is used for determining the tags or the topics. It has two phases of operation namely, candidate selection and machine learning based filtering.

2.3 Exploratory Search

According to Diriye et al.[34], exploratory search task is defined as "an open-ended, ill-defined and multi-faceted search problem that seeks to foster some knowledge product, or inform some action". To explain it further, when the user is not sure about his search, his query might not be able to fetch relevant results from the search

engine. Exploratory search becomes useful here to provide additional help when the query formulation is not straightforward. Using the exploratory search interfaces, a user is able to direct his search using the given options in an interactive manner.

2.3.1 Exploratory Search Interfaces

Usability is an important aspect for any interactive system, and proper design principles can help avoid designs that have poor usability. According to Hearst[33], there are certain design guidelines which are as follows:

1. Provide feedback - For every user action there should be informative messages which can be followed easily. Also, choices or options should have an indicative message to help the user decide better.

2. Reduce short term memory load - This involves having a clear view of the interface, making all the options and messages clearly shown to the user. Requiring the user to remember the different components or features of the interface which might have been shown earlier is not recommended.

3. Provide shortcuts - There should be proper shortcuts to exit, for example if the user wants to quit any particular operation in the middle.

4. Reduce errors - This involves eliminating conditions which involve errors occurring repetitively. Present a bug free interface to the user.

5. Balance user control with automated actions - The interface should be smart enough to perform actions according to the user inputs. It should complement the user actions well, so that it gives a friendly experience.

6. Recognize the importance of smaller details - The interface should clearly highlight the features which are less important but still useful. For example, the help functionality which could explain how the overall interface works.

7. Recognize the importance of aesthetics - The graphical details i.e. the components of the interface such as images, control buttons, boxes, should be presented in an appropriate format (for eg, usage of colors which do not strain the human eyes) to the user has a positive impact on him.

8. Simplicity - Follow the 'less is more' principle to provide the user less choices but with more control to search the information he needs. The formulation of query for search as well as the presentation of the results should be simple.

9. Pleasurability - To make the interface simple yet exciting for user, simulate pleasure inside the user to make them engaged

10. Customizability - This implies we give the user an option to customize the interface according to different parameters or choices

2.3.2 Evaluation of Exploratory Search Interfaces

Evaluating an exploratory search interface is done by assessing the user's behavior and the decisions made during the search. We conduct user experiments where the user is given a task to perform using the search interface. The experiments use the measurements of user satisfaction, the outcomes of the tasks given to the users, user's behavior, the cognitive load and learning. [35] The setting of the experiments and the topics of the tasks can be controlled. Experiments should be conducted in a naturalistic environment and should remain the same for all the users. There should not be any kind of disturbance in the environment. The designed tasks should be across a variety of topics. The users should be assessed first using surveys for their knowledge in the particular area. Based on the user's assessed knowledge, the topic of the task should be provided to the user, the topic should not be too familiar to him and should not be completely new to him.

2.4 Visual Data Analysis

According to Thomas et al.[36], "visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces". The concept is primarily based on using visual representations of data to express meaningful patterns elaborately, allow examination of data for better understanding, and communicate the findings which we think are meaningful in context to the data to others.

Visual data analysis can help make better sense of the data for understanding and explaining it. It helps in analyzing relevant findings and patterns which in turn helps in productive thinking and reasoning about the data. This thesis borrows ideas from the concept of Visual Data Analysis to understand the news data that we have. The visual tools help in exploring large amount of news documents better. By exploring the documents, we can find out the different themes and news events corresponding to some specific news category easily.

We discuss the Focus-plus-Context principle which enables us to see selected parts of the interface while keeping the overall view in the background. We have utilized this principle in the Intent Radar as well as the Global Visualization Map, to help users focus on selected keywords or documents for in-depth exploration. Then, we discuss a concept called Visual Information Seeking mantra , a summarizations of various design principles which helps in designing of interactive visualization tools. We have based our application to follow this concept, where we ensure that our interface provides a good overview of the whole data with respect to a search query. Then, we enable the users to zoom using a Fisheye lens in the Radar and using mouse-over in the Map

2.4.1 Focus-plus-Context

The idea is to display both the focus (local area of the data in detail) and the context (overview of the data) at the same time. In other words, this is a principle of information visualization used to enable users to view components (several parts or features of the user interface) which are of more interest in greater detail but while maintaining a global overview of the information[37]. For a given visualization, the first part of the principle, 'Focus', tells us to display the most important and relevant data at a particular focal point. The data around focal point is generally displayed at full size and with in-depth detailing. The area which is present around this focal point is termed as 'Context'. The aim of the Context is to convey the importance of information with respect to the whole data, thus preserving the global overview of information with lesser details.

The Focus-plus-context principle aims to solve the presentation problem which is occurs often in information visualization when it comes to the management of screen space. It is often noted that many times the visualizations may not allow the user to show detail and context together efficiently. By using the Focus-plus-context principle we allow the user to explore the information related with context and provide him a way to focus on detailed information in a specific area.

2.4.2 Visual Information-Seeking Mantra

Considering there could be numerous ways of seeking information and following different paradigms to do so, one of the design principles for design of the work-flow in an interactive analysis system is based on a mantra which says - "Overview first, zoom and filter, then details-on-demand"[38].

2.4.2.1 Overview

This design principle makes an overall view of the whole data to understand the data set in a general context. This is generally a good starting point when it comes to exploring the data. Using this view we could then select some of the relevant and important features and subsets of data for in-depth analysis.

2.4.2.2 Zoom and Filter

Zoom: After getting an overview of the data and spotting interesting patterns in it, we can start examining the data further. We select specific components of the view and analyze them more closely in this step. There are several types of basic zooming operations. First being the Geometric zooming which stresses on the fact that we

can specify a scale of magnification to zoom into the image and then focus on a specific area to zoom into, essentially discarding the information present outside. The next technique is called as Fisheye zooming which does not discard the information present outside after zooming in, but it would rather show it in a small space using some distortions. Finally, semantic zooming transforms the context in which we are visualizing the given data. For example, dots in a map of a city could transform into 3-Dimensional box shapes showing the actual buildings which are located at given points.

Filter: It is important from user's point of view as it allows the user to control which data or feature ranges are shown in the plot interactively.

2.4.2.3 Details-On-Demand

The techniques used here requested by the users to gain more in-depth information of a specific dataset. The techniques provide greater detail and progressive refinement. A popular example using this approach could be displaying of additional and detailed information by mouse hover in a visualization. This allows display of detailed information without changing the context of presentation or modifying the information that is currently present on the user interface.

2.5 Dimensionality Reduction

In many machine learning applications, the data which is being used for analysis, can contain a large amount of features (dimensions) describing the data samples. If the data is represented as matrix, it would contain a large number of columns representing the dimensions. This is problematic as it can increase the computational cost required to perform tasks such as classification or regression with this kind of data. It could give rise to a problem which occurs in high dimensions known as 'curse of dimensionality' where for such a high-dimensional data the classifier would require enormous amount of training samples to obtain a good prediction [57]. This is done to obtain sufficient samples for different combinations of feature values, because with increasing number of features, simple training algorithms would require to see at least one example from each of the different combinations, to make the correct prediction. Given that the number of combinations would be extremely high for a large number of features, it is unlikely that one could obtain sufficient training samples. Considering, how the curse of dimensionality can affect the performance of the task when the number of features increase in the data, it is often beneficial to reduce the dimensions of the data. The method benefits the computational efficiency of the analysis task we are carrying out and also improves the accuracy of the same.

The techniques used in dimensionality reduction are divided into two parts. We could either use them to perform the supervised and unsupervised classification or we can also use them for feature extraction and feature selection aspects. In this thesis we are dealing with exploratory tasks which requires the input data in form of a high dimensional TF-IDF matrix, where dimensionality reduction can be used to reduce the high dimensional TF-IDF matrix into a lower dimensional one. The output low-dimensional matrix could be then used to perform approximate nearest neighbor classification tasks. The nearest neighbor matrix is then used to perform another dimensionality reduction into a two dimensional output space, so that we could get a 2-D plot of the documents in a map based layout. In this way, we use dimensionality reduction for our exploratory tasks.

Some of the common techniques used for dimensionality are Principal Component Analysis[12], and Linear Discriminant Analysis [20].They produce linear data transformations. There also exist non-linear methods like Multi-Dimensional Scaling[18], t-distributed stochastic neighbor embedding[43], and the Self Organizing Maps[19].

2.5.1 Principal Component Analysis

We shall discuss some commonly used methods here, starting off with Principal Component Analysis(PCA)[12], which is a statistical procedure to represent the main directions of variation in a high-dimensional data set in terms of a smaller number of uncorrelated variables. The method uses orthogonal transformation to transform the variables into uncorrelated linear combinations. This method allows us to visualize and analyze the data on the basis of as few variables as possible. essentially performing dimensionality reduction.

Given n variables which might be correlated, the principal components are interpreted as follows. The first component is the linear combination of the standardized original variables which explains the highest possible variance in the data. Afterwards, each successive principal component is the linear combination of variables which has greatest possible variance and is not correlated with previous components. The eigenvectors of the correlation matrix for n variables are the principal component directions, and the vector of the n variables can be projected to each principal component direction to yield the value of that principal component. Correspondingly, the eigenvalues of the matrix represent variance of each component.

2.5.1.1 Singular Value Decomposition

Singular Value Decomposition[15] is a technique of linear algebra which is used in factorization of matrices and can be used as part of various dimensionality reduction methods. In particular, it is one of the ways to perform principal component

analysis, if computing an eigen-decomposition of a covariance matrix is too computationally difficult. It has been used as a dimensionality reduction method in numerous applications like information retrieval, gene expression analysis. The aim of SVD is to represent a high dimensional matrix using a smaller number of variables.

Mathematically it is represented as,

$$X = USV^T$$

Where, U is an $m \times m$ matrix, S is an $m \times n$ diagonal matrix V^T is an $n \times n$ matrix

The diagonal entries of S are singular values of X. The columns of U and V are left and right singular vectors for the corresponding singular values.

2.5.1.2 Truncated Singular Value Decomposition

Using the traditional SVD could be expensive in terms of time and memory requirements. We choose to use a variation of SVD in this case which is known as Truncated SVD. The method is an approximation rather than an exact decomposition of the original matrix. The method only computes t column vectors of U and t row vectors of V which correspond to the t top largest singular values in S. This is useful when t « r (Rank of matrix X). As a result of selecting a portion and discarding the rest of the matrix during calculations, we get time and memory efficient solutions.

Then mathematically it would be represented as,

$$\widetilde{X} = U_t S_t V_t^T$$

Where, U is an $m \times t$ matrix, S is an $t \times t$ matrix, V^T is $t \times n$ matrix

The method is an approximation rather than an exact decomposition of original matrix. But, as a result of discarding the rest of the matrix during calculations, we get time and memory efficient solutions.

In applications of text mining SVD provides significant improvements over other methods [16].

2.5.2 Random Projections

Using PCA is a traditional way to perform dimensionality reduction tasks. However, with large data sets, it becomes computationally demanding to use PCA in that particular application. We then have to look for simpler dimensionality reduction methods which do not significantly bring out distortions in our large data set.

Random projections[17] are known to provide comparable results to conventional

dimensionality reduction methods like PCA, and the similarity of the data vectors is known to be preserved by this method. However, one of the big advantages of using Random Projections is that it is computationally less demanding than traditional methods, which makes it suitable for large datasets.

2.5.3 Neighborhood Embedding

Our work uses a scalable version of the t-distributed stochastic neighbor embedding which is based on the work of Z.Yang et al.[41]. The work in [41] shows how, in different nonlinear dimensionality reduction methods, the optimization of low-dimensional output coordinates of samples can be made scalable. In each such nonlinear dimensionality reduction method, the cost function typically considers some statistics such as distances or neighborhoods, and measures their preservation from the original space to the output space. This method addresses the problem that many current state of the art non-linear dimensionality reduction methods are not able to scale to large data sets due to high computational complexity. This is relevant to our scenario where we wish to visualize a very large corpus of documents. Old speedup solutions like applying the methods to the subset of the data tends to ignore important relationships present in the data. In contrast, the method of Yang et al. does not ignore any relationships but instead provides a fast and scalable version of the current NE methods by applying an efficient approximation to their gradient terms. The interactions between far-away samples do not contribute much to the gradient which allows the contribution of the methods considered here computed using much stronger approximations.

2.5.3.1 t-SNE

We use Yang et al.'s scalable version of a particular dimensionality reduction method, t-SNE. It is a probabilistic approach to map objects which are given as high-dimensional vectors, or represented as pairwise-dissimilarities, into a low dimensional space by preserving neighborhoods of the given objects. The embedding is optimized so that the probability distribution over all the potential neighbors of a specific object is well approximated by the corresponding distribution on the display.

The t-SNE technique is used for visualizing high dimensional data in a 2-D or 3-D coordinate system, which can be easily visualized using a scatter plot. It reduces the tendency to crowd points together in the center of the map. The process constructs a probability distribution over objects represented as high-dimensional vectors: the probability of object j to be picked as a neighbor of object i is high if the objects are similar and low if they are dissimilar. Then, it defines a similar probability distribution in the lower-dimensional output space and optimizes the projection of

the high-dimensional data to minimize the difference between the distributions.

For a set of multivariate points , the neighborhood matrix P is constructed in such a way that P_{ij} is proportional to the probability that x_j is a neighbor of x_i. Neighborhood embedding will try to find a mapping $x \rightarrow y$ for i = 1,...,n in to preserve the neighborhood in the mapped space. Then, let the neighborhood be encoded in Q in the mapped space such that Q_{ij} will be proportional to the probability that y_j is a neighbor of y_i. Now, the task of the neighborhood embedding is to minimize $D(P||Q)$ over $Y = [y_1, ..., y_n]$ for some divergence D.

It uses the Kullback-Leibler divergence[44] as the cost function and tries to minimize it with respect to the location of points in the map. The cost function is given as, $min_Y D_{KL}(P||Q)$ Where, $P_{ij} = p_{ij} / \sum_{kl} p_{kl}$ and $Q_{ij} = q_{ij} / \sum_{kl} q_{kl}$ with q_{ij} proportional to the Cauchy distribution[42].

2.5.3.2 Kullback-Leibler Divergence

The Kullback-Leibler divergence is a measure of relative entropy i.e. the extra number of bits needed for data encoding when expecting a particular distribution but the data obtained is from a different distribution. In other words, The KL Divergence gives the measure of information, which is associated with two probability distributions involved in a particular scenario. The measure provides the KL divergence which tells how different or similar two probability distributions are actually.

For discrete probability distributions for example if we have $p = \{p_1, p_2, ...p_n\}$ and $q = \{q_1, q_2, ...q_n\}$ then the Kullback-Leibler distance is given as,

$$KL(p, q) = \sum_i p_i \log_2 (p_i/q_i)$$

2.6 Approximate Nearest Neighbors

Our document data is first represented as a large sparse TF-IDF-valued matrix, from which we create a reduced dense data matrix by principal component analysis. Next, we had to to find out the nearest neighbor documents for each document. This was needed to generate our global visualization plot of neighboring documents.

Because the number of documents in our dataset is large and the dimensionality (vocabulary size) present in the documents is also large, searching for exact neighbors is too costly. We thus opt for an approximate nearest neighbor approach. Approximate nearest neighbor approaches can reduce computational cost and hence be time-efficient while still providing sufficiently accurate results..

We use the methodology proposed by Muja et al. [39] to find five nearest neighbors of a given document. Priority Search K-means algorithm is used to find the approximate nearest neighbors.

2.6.0.1 The Priority Search K-Means Tree Algorithm

The Priority K-means algorithm [39] is used to find the approximate nearest neighbors of a data point by making full use of the natural structure existing in the data. The data points are clustered across all dimensions using full distance. The algorithm works in two stages, the first stage where a partitioning tree is constructed and second stage where the constructed tree is explored.

For constructing the tree, the data points are partitioned into K distinct regions using k-means clustering technique. Then the k-means algorithm is applied recursively to the points in each of the regions. We can say that the process is creating levels by clustering the regions recursively. The stopping criteria for recursion is when the number of data points in a particular region becomes less than the value of K. Thus, a tree is formed where the data points are partitioned at each level into K regions recursively.

For searching the tree using a given query point, we initially traverse from the root of the tree to the nearest leaf. A leaf is a node of the tree which does not have any child nodes. The nearest leaf is reached by selecting the branches at each level which have cluster centers nearest to the query point. All the remaining unexplored branches along the path are added in a priority queue. The priority queue is then sorted by the distance between the boundary of the unexplored branch and the query point. The sorting is done in increasing order of distances. One by one, the branches are selected from the sorted priority queue and the algorithm starts traversing the tree again.

The algorithm uses K as a parameter to determining the number of regions while performing the clustering. It is called as the branching factor. Another parameter is the maximum number of points to be examined while searching the tree. Number of iterations during the k-means algorithm is also a parameter in the algorithm. As the fewer number of iterations will build the tree in quicker time but will result in non-optimal clustering.

This thesis uses the Priority search k-means tree algorithm provided by Muja et al.[39] to find five approximate nearest neighbors of a specific document. The Nearest Neighbor matrix is generated based on this information. The matrix is then used as the input for the t-distributed stochastic neighborhood embedding method [41] which generates the global visualization plot in a 2 dimensional space.

2.7 Cumulative Gain

Cumulative Gain measures the usefulness of a result set containing scored results [45]. The results are rated by their degree of relevance for a given information retrieval task. Given a list of graded results, cumulative gain is defined as the sum

of all the graded relevance values in that list. This is given as,

$$CG_p = \sum_{i}^{p} r_i$$

Where, CG_p is the cumulative gain at rank position 'p' i.e. considering top 'p' number of documents , r_i is the graded relevance of a result placed at a position 'i'.

We have used the concept of cumulative gain while calculating the final results. The result set in our case consists of the graded answers written by the participants, for various tasks defined for the user experiments. The answers are graded based on their degree of relevance, according to the task.

2.8 Wilcoxon signed-rank test

The Wilcoxon signed-rank [47] is a nonparametric test used to verify the similarity of two dependent data samples if they were obtained from populations of the same distribution. This is different from t-test [49] which assumes the data to be normally distributed. Since, we cannot assume the data to be normally distributed, Wilcoxon signed-rank would be able to work with the ordinal data in our case.

We used the Wilcox signed-rank test to check the similarity of samples obtained during user surveys for the two systems i.e. The baseline system (Radar) and the full system (Radar + Map), for this purpose, we used a paired version [48] of the test to compare the results of the two systems. During the surveys, different questions concerning the user's experience with the system were asked for both the systems, and this test helps us to analyze the differences statistically. The significance of results is measured using the p-value, for which the threshold was set at 0.05. When the p-value is less than the threshold value of 0.05, we can deduce that the comparison is statistically significant.

3. THE SCINET SYSTEM FOR EXPLORATORY SEARCH

3.1 Overview

This thesis is based on extending SciNet, which is a feedback based search engine [1]. It uses publication papers as the backend data and searches for the same upon querying. It is based upon interactive intent modeling where user has to direct his search by providing estimates of search intent via feedback. The system then uses an Intent Radar to visualize the estimated intents which are represented by relevant keywords extracted from articles. The Radar displays the intent in a radial layout where relevant intents are closer to the center and similar intents tend to have angles which are similar. The user is able to provide feedback to the system by changing the intents in the layout according to the relevance of the results. The feedback is used by the system to learn and provide improved results and intent estimates.

SciNet supports the user in exploratory search, meaning search tasks where the user doesn't have sufficient knowledge about his search or he doesn't know how to reach his search goals. The aim is to help the user explore, investigate, and learn at the same time when using the system, better than when using a traditional system. The approach combines a standard querying process with effective and novel interactive feedback strategies to reach the goals, by allowing the user to effectively inspect search results and intents and to give feedback to them.

3.1.1 Interactive Intent Modeling

3.1.1.1 User interface

The user interface introduced by Ruotsalo et al. [1] consists of a query box, the result box and an interactive Intent Radar for giving the feedback. The radius of the keywords denotes their relevance, the most relevant keywords tend to be near the center of the Intent Radar and those with less relevance far off. Angles here denote similarity of intent for the keywords, keywords whose relevance behaves similarly with respect to feedback are given similar angles. Generally, there's a concept of semi-local and local keywords. The local keywords fall inside the center most area of the Radar and semi-local keywords are placed in the outer region of the Radar and

tend to form several clusters. The semi-local keywords are also colored according to clusters of the angles. Most of the keywords within a cluster are shown as points, one of them is shown with a label to define that cluster.

3.1.1.2 Interaction and Feedback

As noted earlier, the user can drag a keyword towards a center to give it a positive feedback (meaning the user is more interested in the keyword than the system estimated) and drag it away from the center to provide a negative feedback (which means that the user is less interested in the keyword than the system estimated). The user can also click the keywords beneath each of the documents in the result box to provide a positive feedback to the system. For the first iteration of the search, the documents and the keywords are retrieved on the basis of a pseudo-feedback from top ranked documents.

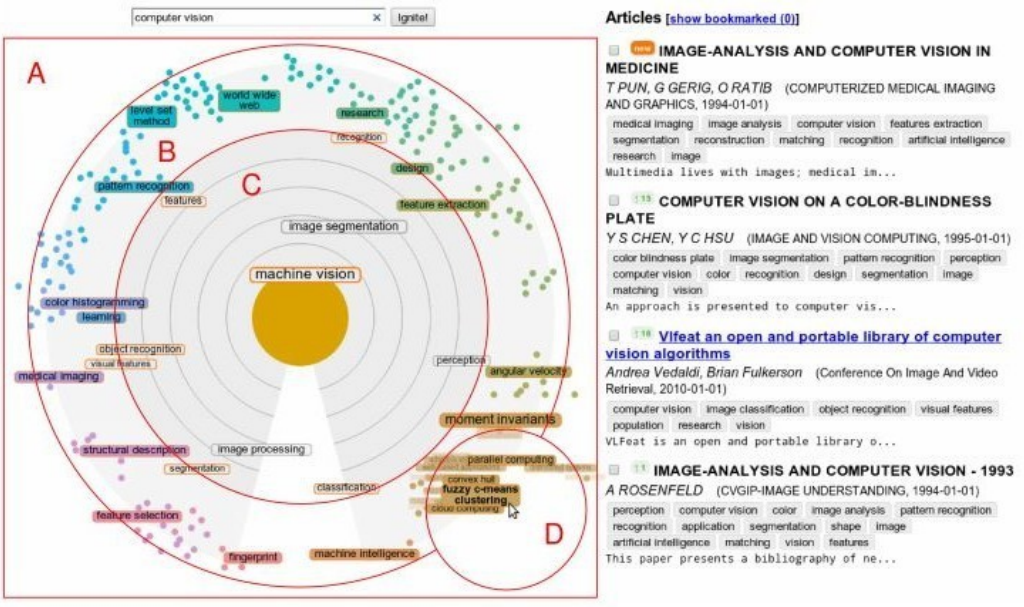


Figure 3.1: Left: The Intent Radar Interface A) Radial layout which shows the keywords i.e. the search intents. The user is represented as the center. B) The intent model which is used for document retrieval is visualized in the inner circle as keywords. A particular keyword is shown near to the center if it is relevant to the user's estimated intent and far from the center if not. C) Future intents are projected using keywords and visualized in the outer circle D) The fisheye lens feature for inspection of keywords. This image is taken from Ruotsalo et al. [1]

3.1.2 Document Retrieval and Relevance Estimation Model

The work of Ruotsalo et al.[1] uses a multinomial unigram language model to rank the documents in order of relevance. This is decided based on the estimate of the search intent of a user.

A keyword weight vector $\widehat{\mathbf{v}}$ is obtained from the intent model, which has the weight $\widehat{v_i}$ for some keyword k_i and d_j denotes a specific document. All the documents are ranked by the probability that $\widehat{\mathbf{v}}$ is going to be observed as a random sample from the document language model M_{d_j}. The equation given by maximum likelihood estimation is,

$$\widehat{P}(\widehat{\mathbf{v}}|M_{d_j}) = \prod_{i=1}^{|\widehat{v}|} \widehat{v_i} \widehat{P}_{mle}(k_i|M_{d_j})$$

. Bayesian Dirichlet smoothing is used to get a smoothed estimate, in order to avoid zero probabilities in the above equation.

The documents are then ranked by $\alpha_j = \widehat{P}(\widehat{\mathbf{v}}|M_{d_j})$. Here, α_j is used for ranking the document d_j. Instead of just showing top documents to the user, a subset of top documents is selected on the basis of sampling. The subset of documents are presented to the user which increases the chance of showing not only top but also novel results. This is favorable for the documents whose keywords have obtained positive feedback from the user.

The model has two representations in the form of current estimate of the search intent and the alternative future intents which are likely to occur based on the feedback given by the user. The current estimate of search intent is denoted by relevance vector $\widehat{\mathbf{r}}^{current}$ over the keywords. The set of alternate future intents are represented in the same way by $\widehat{\mathbf{r}}^{future,l}$ which are relevance vectors predicted in future. They are called future relevance vectors. Each vector $\widehat{\mathbf{r}}^{future,l}, l = 1, ..., L$ is a projection of current search intents into future intents by using a set of L feedback operations which can be given by the users in future. The user gives feedback to current search intents in the form of relevance scores $r_i \in [0, 1]$ to a subset of keywords $k_i, i = 1, ..., J$. A score of 1 for r_i denotes that the keyword is highly relevant to the user and a score of 0 means that the keyword is not relevant to the user.

The relevance of the keywords is estimated by representing each keyword k_i into a $n \times 1$ vector $\mathbf{k}_i$. The vector provides information about the occurrence of specific keyword k_i in the given set of n documents. The significance of the documents is boosted by using TF-IDF representation. The relevance score of the keyword k_i is denoted as r_i and is assumed to be a random variable with expected value $E[r_i] = \mathbf{k}_i^\top \mathbf{w}$. The $\mathbf{w}$ here is the unknown weight vector which provides the relevance of the keywords. The value of w is estimated using the LinRel algorithm, which uses

the feedback provided by the user in the search session so far.

For selection of keywords to the user, the system does not pick the keywords with the highest relevance scores but tries to pick the keywords with the largest upper confidence bound for the score. This is because when only little feedback has been given so far, the scores may not be well predicted yet and this can make the choice to take the keywords with highest relevance scores suboptimal. The system also provides a set of alternative future search intents which are basically the estimates of the relevance score vector for the future. At each search iteration, the future search intent is estimated by providing a pseudo-relevance feedback of 1 to each of the alternate feedbacks and then adding it to the feedback from previous search iterations. The future relevance vector $\widehat{\mathbf{r}}^{future,l}$ for keywords is then estimated using the LinRel algorithm.

3.1.3 Layout Optimization

The keywords are arranged on an Intent Radar based layout. The inner circle represents the current intent and the outer circle represents the future intents. The system uses probabilistic modeling-based nonlinear dimensionality reduction to optimize the locations of the keywords. The keywords which can become relevant to the user in future are shown in the outer circle by using their potential future relevances to determine their radius and angle.

3.2 Previous User Experiments on the SciNet System

In the original SciNet paper[1], user experiments based on given tasks were carried for SciNet. The aim was to measure natural user interaction while retaining advantages of a controlled experiment.The previous studies were based on the ability to direct search, system are a follow-up. We discuss results of the previous experiments to provide context for our experiments.

3.3 Results and Conclusions of the Previous Experiments

- Task performance - The results from the evaluation measures showed that the Intent Radar attains better performance than the Typed Query system, a baseline system that only allowed typed queries without visualizations or feedback

- Quality of Displayed Information - The Intent Radar system based on interactive intent modeling had better performance as compared to the Typed Query comparison system.

- The results showed that the Intent Radar was used more by the users as compared to the Typed Query and another baseline system. The Typed Query did not help in finding novel information as compared to Intent Radar. The intents visualized on the Radar helped the users to fetch more novel and relevant information.

3.4 Software Implementation of SciNet Search

This section discusses the software technologies which are used in implementation of the SciNet search system. The technologies used are open source and run on a Linux platform. The software technologies used here makes the development process of the system much more simplified and cleaner. This way the whole debugging process becomes easier as we could debug a specific component in the whole system. The following are the technologies which have been used in SciNet along with their role in the system.

3.4.1 Apache Lucene

Apache Lucene[30] is a Java library which is designed to support the functionality of using full text search across a collection of documents. It is suitable for many applications which require full text searching and is designed to have cross-platform support. It relies on the notion of inverted index for searching. The inverted index is meant to form a data structure where the tokens serve as keys with the documents as the values. The tokens here are depending upon the need, and they could be unigram,bigram or trigram. They are optionally stemmed, removed if they are stopwords and added with extra information (position) which is called meta-data. For ranking of the documents, Lucene uses a language model similarity measure based on SciNet's document retrieval and relevance estimation model as described in section 3.1.2. Lucene stores the given document in a Lucene index where each document is stored as key/value pairs. The keys here are called fields and they store all the relevant information which we want to keep. This process of storing the documents is also called indexing.

3.4.2 Spring Model View Controller Framework

Spring is an open source framework developed for Java programming language, which lets us perform a variety of operations for building different kinds of applications. It has various modules which provide different functionality. The SciNet system uses the Model-view-Controller framework (MVC) module. The documents stored in the Lucene index are queried using Spring MVC Framework which provides a cleaner approach to gathering and communicating the data further. The

results queried are then communicated towards a Python based backend developed in Django REST Framework, which continues the further steps.

3.4.3 Django REST Framework

Django is a Python library with numerous features to be used for web based applications. One of the components of Django is the REST framework which allows the development of REST based applications. The REST [29] is an architecture which is used by modern web browsers to send and receive data from remote servers on a regular basis. Owing to the usage of a lightweight universal protocol HTTP, REST applications have been used in a variety of domains for data transfer applications. REST based web services can send and return data in different formats, in SciNet and the work of this thesis the most common format JSON is used. After receiving the results from Lucene index via Spring MVC, Django based Python backend picks top keywords based on the current user intent model.

3.4.4 MongoDB

We use MongoDB in Scinet to keep a log of user activities. The profiles of all users who participate in the user experiments, the search queries made by them and the resulting documents and keywords obtained based on their searches. MongoDB[31] is an open-source Document Database based on the 'NoSQL' principles.

4. SCINET FOR NEWS ARTICLES

The SciNet system was originally implemented to work on a database of scientific publications. In our work, we are working with news articles. The SciNet backend was modified to be able to import XML-format news articles whose information fields differ considerably from those of publication articles, and to use those fields in the information retrieval process and in the user interface. Technically, modifications were done to the way data is stored as an inverted index, to the module that queries articles using the keywords and to the way feedback was processed.

SciNet gathers user feedback through their interaction with keywords on a Radar based visualization. The Radar uses the keywords relevances and displays them to the user. The feedback given by the user through the Radar is then processed by the SciNet backend and as a result the user intent is estimated and the documents and keywords corresponding to the user intent are obtained. In order for this interaction to succeed in directing exploratory search, the keywords extracted for an individual document should be well represented and should be meaningful for the document content. Without representative keywords, the system would not be able to display keywords on the Radar that could correspond to the user interest, which would lead to a poor feedback mechanism and a flawed system overall.

4.1 Dataset

The dataset that our version of SciNet searches is generated from news articles. The news articles are online published articles covering a variety of topics. They are crawled from the editorial section of several popular news websites spread across the world using web crawlers. The articles were crawled and stored from the websites over a period of 1 month during September 2013. Only English-language news articles were crawled from numerous English-language websites over the world. Currently, SciNet is focused on being able to explore only English articles. The articles are being stored in a MySQL database and some of the following fields are relevant to us :
title - The title of the news article
URL - The web address of the news article
processed_keywords - The keywords of the article which are obtained using a keyword extractor

plainTextContent - The text content of the article
category - The category of the news articles
contentLanguage - The language of the news article
authorName - The news website where the article was published
region - The region where the news website is located from which the data was crawled
publishDate - The date when the news article was published

The dataset was provided by a company called Mbrain[21], which is a global information, technology and consulting services company dedicated to providing media, business and market intelligence solutions. The original data from Mbrain contained around 8 million news articles. We used a subset of 1.6 million articles to speed up processing for carrying out our data transformations, experiments and analyses.

4.2 Keyword Extraction

The news articles are generated by crawling through various news websites over the internet. To represent an article in the SciNet system, keywords must be extracted from the content of the webpage containing the article. Some of the articles contain keywords in the meta field of the webpage but many of them do not. In addition to any keywords available in the metadata of the webpage, keywords can be extracted from the full text of the article itself. In some of the cases, it was also observed that they contain advertising keywords inside the webpage's meta data. As discussed earlier, in order for the SciNet system to work successfully we need to have precise keywords. Keyword extraction is a well researched topic [26; 27; 28] and aims to extract words/phrases that are strongly descriptive about the topical content of the text. There were two tools tested for our purpose - 'KPMiner'[24] and 'Maui'[23]. KPMiner is a popular tool which is known to have provided good results with a variety of documents, but for our usage it doesn't yield satisfactory results. In earlier publications on SciNet [1; 2; 3; 4] , KP-Miner was extensively used to extract the keywords from news articles, and the results were satisfactory. On trying the same tool for our data, there were many keywords that either did not contain topical content, were grammatically unsuitable as a descriptive keyword, or were too short to be understandable as a descriptive keyword, for example - 'said , told, said yesterday, claim, unlink, israel said' In our view the resulting keywords were not of sufficient quality to be used in the search system. Hence, we used the keywords generated by Maui[23] in this work, whose results seemed to be satisfactory for our application.

Maui tries to augment the keyphrase extraction algorithm (KEA) [25] by using semantic knowledge from Wikipedia. In order to obtain the semantic knowledge,

Maui utilizes Wikipedia Miner which provides Wikipedia based features to its classification model. Wikipedia Miner utilizes a Wikipedia dump containing all the data from Wikipedia and summaries mentioning relationships and patterns in the dump [53] for extraction of information.

4.3 SciNet for News in Action

There are series of steps which were done in order to make the existing system of SciNet work with the news articles. Since the schema of the news articles is different from the publication articles, we need to change the way how data is stored and retrieved. First we store the news articles in a Lucene[30] index. Next, we setup the web services for querying and retrieving the results for displaying in the user interface.

4.3.1 Indexing Process

The backend uses a Lucene index to store and search the documents. Lucene stores the document in key/value pairs where the key is a specific field for example title, URL and the values are the actual data for the fields. Lucene uses a language model similarity measure which is based on SciNet's document retrieval and relevance estimation method. Relevance scores are then calculated using Lucene's similarity measure. The scores are used to rank the documents from the most relevant to least relevant documents for the given query. The higher the score, the more relevant a particular document is to our given query.

For storing the documents in the index, we modified an existing indexing application which reads the documents in the XML format. The documents with their extracted keywords are stored in the central database server. We developed a script to dump the documents from the database in an XML format in a folder, to be read by our indexer. Once the XML documents are dumped in a folder, we run the indexer by specifying the location of the folder where the files are kept. The XML documents are read one by one and the indexer parses the XML fields to read the data along with their data types. After parsing the fields and extracting the data, a number of cleanup operations are done on the data, such as, formatting and removing garbage characters and extra whitespaces from the title. Finally the data is stored into the Lucene index with the required field names.

4.3.2 Querying Process

For querying, the backend makes use of Model View Controller paradigm based on the Spring framework[51] which takes in the query keywords and their weights according to the user intent model along with optional other parameters. The query

keywords are used to query the Lucene[30] index and return the results. The result documents are displayed via XML or JSON format. The documents and their keywords obtained from Lucene are then sent to the Python backend which picks the top keywords, as ranked by the current user intent model, computes a layout for them on the Radar, logs the results and sends them to be displayed in the frontend user interface. We also modified the web services code to work according to the schema of news articles. The results are communicated from the Python backend to the frontend using the JSON format.

The user interface presented to the user is then populated with the results. The resulting documents are ordered by their relevance scores and the Intent Radar is visualized with important keywords. The JSON obtained from the Python backend includes a list of local and semi-local keywords and their respective relevance weights. The local keywords are the ones which are visualized inside the central area of the Radar and semi-local keywords are visualized in the outer boundary area of the radar in clusters. As discussed earlier, the local keywords are the current intents and semi-local keywords are the future intents.

The user feedback is given using the Radar by dragging keywords the user wishes to indicate as relevant towards the center of the Radar and dragging the less relevant keywords away from the Radar. The backend re-estimates the user intent model based on the feedback and uses it to estimate the relevance of multiple keywords. The most relevant keywords and their weights are then used to search in Lucene for a new set of documents and the process begins again.

5. EXTENDING SCINET: INTERACTIVE VISUAL OVERVIEW FOR LARGE NEWS DATA

In this thesis the original SciNet is extended by introducing new features. We visualize the whole data stored in the backend by scalable neighborhood embedding methods. During this process, we have used several scalable solutions to work with our large dataset in order to generate the final visualization. We have added a new functionality to the SciNet user interface by building an interactive Global Visualization Map which has provided additional ways of exploring the data and sending user feedback.

5.1 Global Visualization Map

The addition of a global corpus visualization Map helps the users in many ways which were not possible only by using Radar. When used along with the Radar, it provides an overall view of the search results. The user is able to identify different themes present in the content of the search results from the Map. The Map helps the users in exploring the search results quickly compared to viewing the result documents as a long list. By sending feedback, users can also refine their search to understand how comprehensively they have searched so far.

The Global Visualization Map is generated using the Mbrain data which consists of news articles previously described in section 4.1. For the results in this thesis in order to reduce computational complexity, we used a subset of 1.6 million articles to build our model. The process of the Map generation follows a series of steps.

5.1.1 Unigram Language Model

The language model described here is used to create the Global Visualization Map. We are specifically building a language model using unigrams because the keywords obtained are too sparse to yield enough information about the patterns in the data. The language model helps us in summarizing the content in each area of the Map.

To build the model, we follow the standard set of steps followed in a Natural Language processing task:

1. Data pre-processing: The data was cleaned, special characters were removed. We also set a condition to filter out words whose length is less than three

characters. This helps in making data sane for our usage before we actually start applying algorithms and methods.

2. Choice of n-grams: The text content is then split into n-grams using whitespace as a delimiter. Our model is build with unigrams where each unigram represents a single word in the text.

3. Stopwords Removal: This step involves removal of common words which occur very frequently in the corpus but do not convey important information. Some examples of stop words are - 'the','is','an','this'. We will consider only English stopwords.

4. Stemming : Various kinds of stemming algorithms are present for our usage. Stemming allows the reduction of words to their basic word root. We choose to go with the Snowball stemmer[55] for English words.

5. TF-IDF Matrix : Generate a high-dimensional sparse TF-IDF matrix for 1.6 million documents by calculating the TF-IDF score of each term using section 2.1.3.1

5.1.2 Dimensionality Reduction

A first simple Dimensionality Reduction step is needed to reduce the computational complexity before applying the final Map creation step in our work due to a high dimensional input matrix. For reducing the original TF-IDF input matrix to a smaller number of features, we are going to use a dimensionality reduction method as discussed in the previous chapters. For our purpose, we first tried the Random Projections[17] approach to reduce the dimensions of the given original sparse matrix to 100 dimensions. We evaluated the performance of our choice by using a K-means[14] clustering method to cluster the data points in 100 different clusters. The results showed that most of the data points were clustered in 1 large cluster, therefore leaving most of the remaining clusters with just 1 or 2 data points in them. This clearly shows that the random projections have not sufficiently preserved interesting structure of the data and we therefore need to look for other options to perform the dimensionality reduction.

We switched to Singular Value Decomposition(SVD)[15] approach to reduce the high-dimensional TF-IDF matrix to 100 dimensions then. SVD approaches are generally known to work well with text data [16]. Since we have a large dataset, we used a Truncated SVD[13] approach which is known to be economical in terms of time and memory requirements. We performed the K-means clustering again with the reduced matrix from Truncated SVD, and the results produced clusters which had a uniform distribution of data points across all 100 clusters.

Since this new approach worked well for our case, we stored this final output from the dimensionality reduction method in a separate file, to avoid doing the redundant work again. After this, we proceeded to calculate the nearest neighbors of the data points.

5.1.3 Approximate Nearest Neighbor Search

Our ultimate visualization depends on the data neighbor relationships and in order to compute them efficiently for 1.6 million documents with 100 dimensions, we need to use an approximate version of the nearest neighbor search technique. We used the method proposed in *Marius Muja et. al.*[39] for calculating the approximate nearest neighbors. We are using the approximate neighbor approach owing to the fact that we have a massive dataset and calculating the exact nearest neighbors would be too expensive in this scenario. In the previous section, we used the Truncated SVD approach to reduce the data to 100 dimensions. Performing the dimensionality reduction provides us the dense matrix which is required by the approximate nearest neighbor algorithm. We have initially used 5 nearest neighbors to build the model. We used the FLANN [40] library which uses the method proposed in [39]. The algorithm provides 5 approximate nearest neighbors for each of the documents that we are going to use next.

5.1.4 Neighborhood Embedding

The results obtained after performing the approx NN search were in a format like this:
13456, 194856, 45, 245, 1
1857, 138, 10, 49, 4444
149603, 90, 4, 39586, 3005
58935, 67, 8, 39485, 3994
.....
28, 78, 3848, 2395, 83747

Where each line corresponds to the indices of the approximate nearest neighbors of a particular document. The line number is the id of that specific document.

We converted the results obtained from approximate NN approach into a sparse neighborhood matrix. The matrix is a 5-NN graph which is basically a nxn matrix with only binary values. An exemplary NN graph matrix is shown below:

$$
\begin{array}{c}
 \\
d_1 \\
d_2 \\
d_3 \\
d_4 \\
d_5 \\
d_6 \\
d_7 \\
d_8 \\
\vdots \\
d_n
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc}
d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & \dots & d_n
\end{array} \\
\left(\begin{array}{cccccccccc}
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & \dots & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \dots & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \dots & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & \dots & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & & & & & & \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots & 0
\end{array}\right)
\end{array}
$$

Where d denotes the documents and n denotes the total number of documents. Each document has 5 nearest neighbors, a value of 1 denotes that the two documents are neighbors while 0 denotes they are not.

Next, we used the scalable version of t-distributed stochastic neighbor embedding to create the global visualization. We used the KNN graph as an input for the neighborhood embedding (NE) software based on the work of *Z.Yang et al.* [41]. We used a weighted tsne method for building the neighborhood embedding. All the other parameters were set to their default values while running the software. The output from the NE software is an output matrix of document coordinates reduced to a 2-dimensional space which consists of X and Y coordinates. Each point in the plot represents an individual document. The aim here is to preserve the neighborhoods that were computed from a higher dimensional space in the lower dimensional (here 2-dimensionsional) space.

Figure 5.1 shows the case where random projections were used for the initial dimensionality reduction step before neighbor embedding, as the original choice. Figure 5.2 shows the case where truncated SVD was used for the initial dimensionality reduction step, as the final choice.

5.2 User Interface

The user interface for SciNet is extended by adding new functionalities for the user. The aim is to provide additional ways for exploring and investigating the data while browsing. Our target was to build interactive components which can aid the user in learning more about a particular news area or topic. This was done by dividing the whole visualization Map into a grid based interactive system. We used a 20x20 grid based system where each of the grid cells were programmed to respond to a user click. Each grid cell will provide information about relevant keywords and stemmed

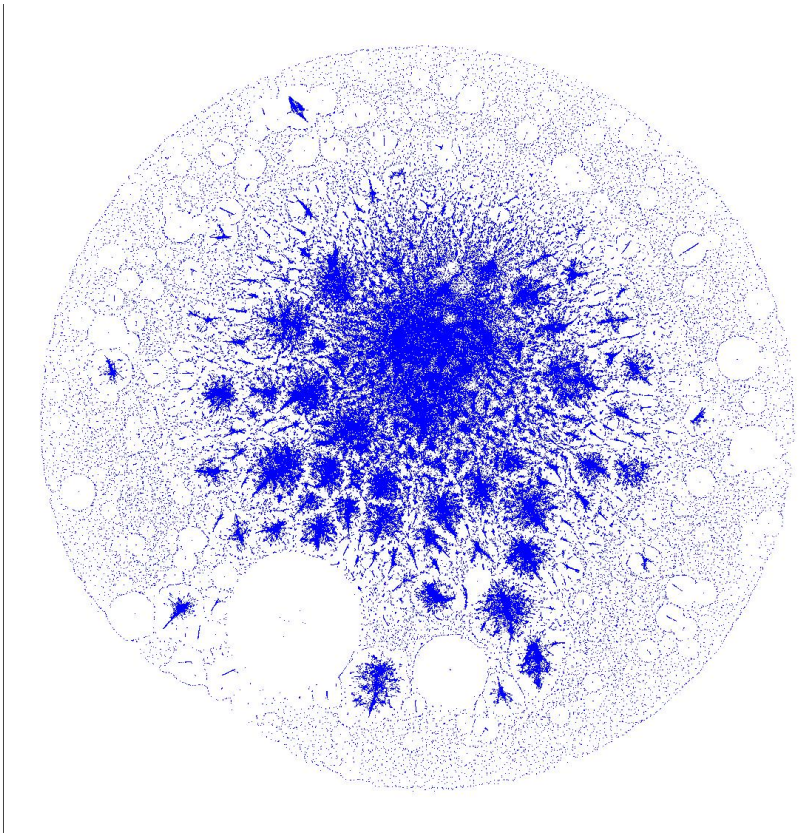


Figure 5.1: The visualization plot generated by performing dimensionality reduction (Random Projections) on the original TF-IDF matrix, followed by Approximate Nearest Neighbor Search and finally using the Scalable Neighborhood Embedding

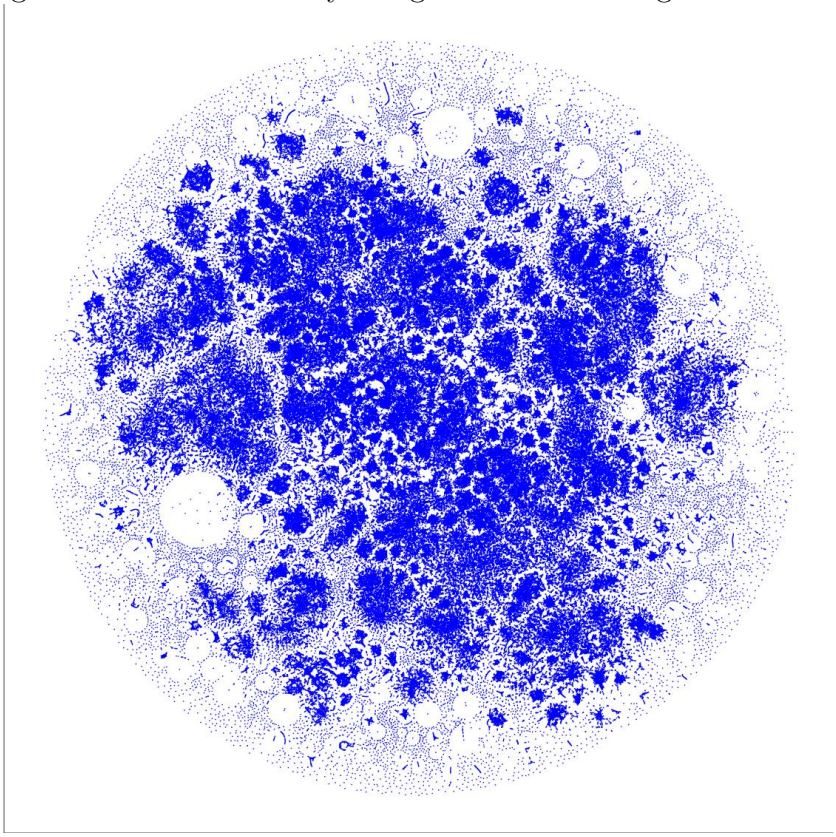


Figure 5.2: The visualization plot generated by performing dimensionality reduction (Truncated SVD) on the normalized TF-IDF matrix, followed by Approximate Nearest Neighbor Search and finally using the Scalable Neighborhood Embedding

unigrams (single-word tokens) prevalent in documents inside that cell. The Map also shows the location of search results on it by the help of image based markers. The markers also have a functionality to show the details of the corresponding news articles.

Figure 5.3 shows the Full Scinet System containing the Intent Radar and the Global Visualization Map, whereas Figure 5.4 shows only the Global Visualization Map of the full system. The components and functionalities of the Global Visualization Map will be discussed in the following subsections

5.2.1 Grid Cells

We used the neighbor embedding plot generated for 1.6 million articles to build an interactive system which the user could use to interact from the browser. The grid cells represent a 20 x 20 matrix comprising of different regions. Each of the grid cells are dynamically programmed to respond to user click. The click of the user is location dependent, based on the current location of the user's pointer in the plot. The click with the location information selects the grid cell relevant to the cursor. The current mouse coordinates are used to query the grid cells which are stored in a database with initial and final coordinates for both X and Y axes. The grid cell which fits the current coordinates within its window is the one which is relevant to us.

5.2.2 Local Unigram and Keyword Lists

Clicking on the grid fetches a subset of the top unigrams present in that area. The information for the unigrams is stored in the backend and is retrieved based on which cell of the grid was clicked. The top unigrams are generated based on the principle described below. The resulting top unigrams are shown as a list at the left edge of the Map window. The unigrams are by default colored white but when a search query has been made and relevant keywords from an intent model have been displayed in the Radar visualization, the unigrams in the Map window will be colored according to which keywords in the Radar they occur in; this links the unigram concepts shown on the Map visually to the keyword concepts shown as interaction options on the Radar. Since we have unigrams in the list and colors in the Radar correspond to the keywords, we use reverse stemming described in Section 5.2.2.1 to build a relationship between both. In addition to coloring unigrams, also the grid cells of the grid themselves will be colored. The coloring will be described in Section 5.2.4.

Figure 5.3: Full Scinet System: Intent Radar + Global Visualization Map

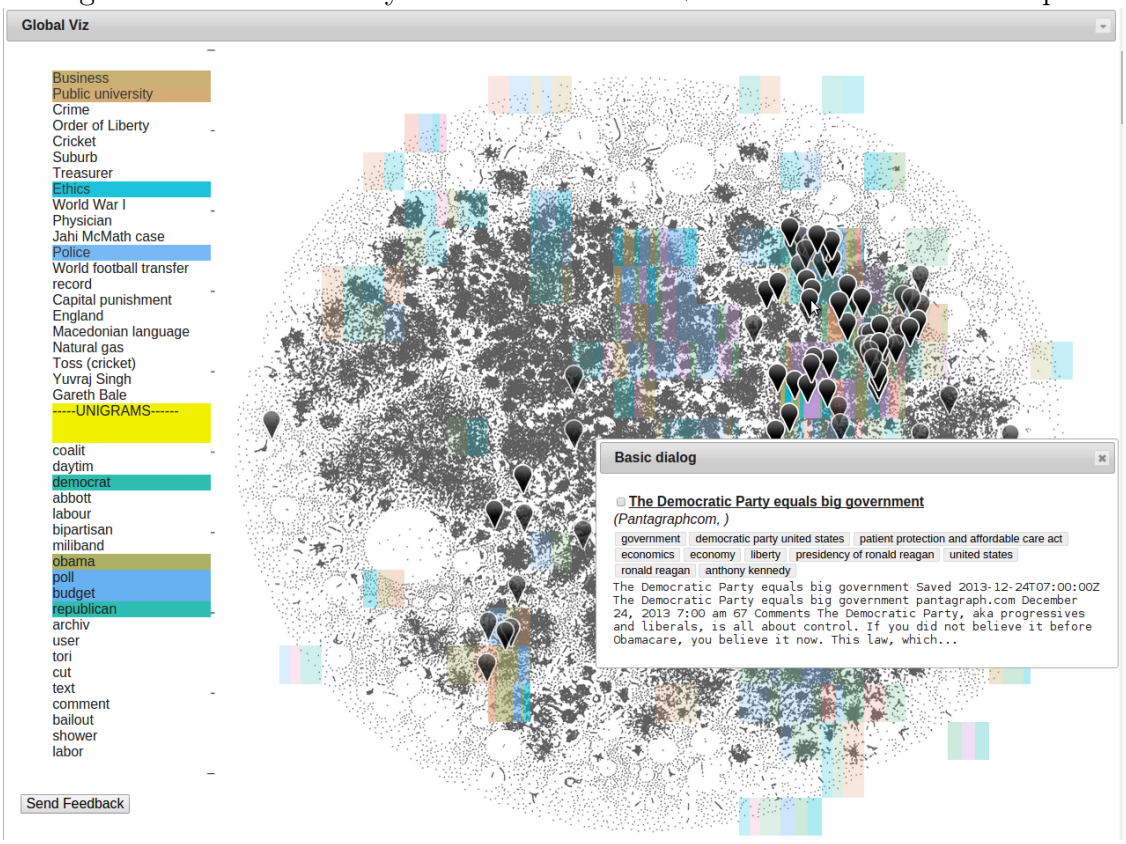


Figure 5.4: Global Visualization Map of the Full System

5.2.2.1 Reverse Stemming

Since the SciNet Radar display is based on keywords whereas the model for the global visualization is based on unigrams, we will use a process which we are going

to call as reverse stemming. The SciNet backend is used to compute the stemmed versions of the different unigrams present in a particular keyword. We then build a key-value based data structure where we will keep a track of different unigrams and the keywords which contain the particular unigram.

5.2.3 Top Unigrams

For each cell of the grid, the top unigrams are generated using the neighborhood embedding coordinates generated in Section 5.1.4. We use the pre-calculated grid cell location information to retrieve the indices of the documents present in different grid cells. To get the top unigrams we select a particular grid cell and fetch the relevant documents for the same. We use the TF-IDF vocabulary matrix containing the TF-IDF scores for all the unigram features. We aim to rank unigrams in the grid cell by average weight across documents in the cell. However, a simple average would rank uninteresting common words near the top even if their TF-IDF scores are low in each document they appear in. To solve this, we only take the top 10 unigrams which have the maximum TF-IDF scores from each individual document and set the scores of the rest of the features to zero for that document. To get the final unigrams, we take the mean of the features over all the documents and select the top 20 unigrams with the best mean scores to be shown in the unigram list.

5.2.4 Coloring

Coloring will be used to relate the contents of the Map to the current search intent model, that is, to the current keywords on the Radar. This is done as a two-stage process. We have to first color the unigrams present in the unigram list according to the colors of keywords on the Radar. Secondly, we need to color the grid cells accordingly. We tried different approaches to tackle this and they are listed below.

We receive the color information from the Radar keywords and unigrams present in the list by 'Reverse Stemming' described in section 5.2.2.1. Using reverse stemming we can identify which keywords contain a particular unigram and then we can identify which of those keywords are on the Radar and which colors they have.

Since we have directly received for each unigram the keywords containing that unigram, and some of those keywords have a color on the Radar, we can assign the color of the unigram to be the average color of keywords on the Radar containing the unigram. However, there were few variations which can be tried out now on the coloring aspect of the grid cells as well as the unigrams present in the unigrams list.

5.2.4.1 Color Variation of Grids

First, we tried using an average of the colors to provide a final color to each unigrams in the unigram list. The final color gives an indication to the user that which color(s) are dominant for the current query. However, for visual representation this is not the most effective solution, it can be difficult to identify the different colors involved in a unigram by just seeing a particular average color.

Then, we opted for a new scheme for coloring the grid, where we could take out the different colors for matched unigrams from the unigram list and define the width of color according to the counts of the occurrences of the different colors. Hence each cell in the final grid that we see is overlaid with stripes of colors of different width percentages which are dependent upon the occurrence of each particular color.

5.2.4.2 Transparency

Along with the coloring scheme which gives the user an indication where unigrams on the Map match keywords of particular color from the Radar, we also define the transparency of the colors overlaid on the grid cells. If the unigram matches the keywords from Radar, we then say that the unigram has 'matched'. The transparency or opacity score provides an indication for the percentage of the total matched unigrams which occur in a specific grid cell.

We use the simple feature scaling with upper and lower limits to determine the opacity of the grid cell by the following equation.

$$Opacity = \begin{cases} 0, & \text{if } UM_{grid} = 0 \\ c1 + \frac{UM_{grid}}{UM_{total}} \cdot c2, & \text{otherwise} \end{cases}$$

Where, UM_{grid} is the number of matched unigrams in the specific grid cell, UM_{total} is the total number of matched unigrams in the Map, $c1$ is equal to 0.05, so that the opacity of a grid is never below 0.05, $c2$ is the maximal bonus opacity (set to 0.90) so that the upper limit of grid opacity is c1+c2

If the opacity is close to 1 that means a large proportion of all the matches occur in that grid cell a value closer to 0 means very few matches. We added the offset c1 to ensure that the minimum value is always 0.05 so the colors of matches can be seen even for few matches and the maximum is set to 0.95 so that the color is not totally solid and the shape of the data can be seen from under the color. This is done because fully transparent grid cells are made to represent areas where we have absolutely no matches whereas overlaying the cells with totally solid colored stripes would block the view of the figure in background. In case, when there are no matching unigrams for a grid, UM_{grid} becomes 0 and the cell is rendered transparent because there is no color in the first place to set the opacity for.

5.2.5 The Document Location

The location of the documents in the Global Visualization Map space is indicated using image based markers. Each query returns a list of 100 top documents from the SciNet backend. Each document contains a unique id. The global visualization plot was generated by projecting each of the documents into a lower dimensional space (2-Dimensional in this case), hence we used the document id to locate the coordinate of the document in the 2-Dimensional Map. We created markers over specific points in the Map which indicate the presence of current results.

Upon hovering the mouse over a marker, a pop-up window opens where we can see the details of the news article such as the title, abstract, keywords and the URL to the original page. Clicking the marker keeps the window open, and we then have the option of sending user feedback by clicking on any of the keywords, this complements the ability to give feedback by dragging the keywords present on the Radar or by clicking the keywords under the documents in the result list. A pop-up window can be more convenient for viewing document details and giving keyword feedback than, e.g., highlighting the document corresponding to a marker in the whole list of document results. The documents are generally arranged in the form of clusters i.e. similar documents are likely to be in the same cluster. The arrangement of documents in the clusters therefore enables us to explore a particular area properly in the Map.

6. TASKS, EXPERIMENTAL SETUP, EVALUATION MEASURES

6.1 Tasks

We defined several tasks for the users to explore our systems. For each user we choose two news areas to work with. The users are given two news areas from a list of news areas. Table 6.1 shows the news areas which are used for the experiments.

For each news area we give the user one of the two systems, which they must use and write down the answers to the given experiment questions. For the first question, we ask the user to list down the 'main topics' of the given news area. Then, for at least 2 main topics, we ask the user search for a main topic and write down at least 2 'themes' based on his observations of the search results. A theme is basically a categorization of several news articles as a subgroup under the main topic, which are similar in their content, in user's own words. Next, for each theme the user has to list down at least two important news events (news articles) from the search results which were relevant to the written theme.

Consider for example, if we give the news area as Sports, the users might list down the main topics as 'Football', 'Basketball', 'Tennis', 'Cricket'. Then the user searches for 'Football' in the search box and obtains a list of results. After seeing the search results, the user observed that there were several news articles which were similar in some way. For example, some articles talked about the FIFA world cup from the year 2010 while some of them were about English Premier league. We can then write one of the themes as "FIFA World Cup 2010" and mention news events like "2010 Fifa World Cup stadiums", "Spain win World Cup 2010" which were relevant to the theme the user wrote.

Table 6.1: The list of news areas used in the user experiments

American Politics
Sports
Entertainment
Finance

6.2 Experimental Setup

We bring the user to a separate experiment room where the user can focus on the tasks without distractions. The room features a computer with a large screen with the systems setup ready to use, and all users performed the test on the same system. The modified SciNet system is setup on one of the servers of the Revolution of Knowledge Work [22] research project, where one could log in and use the system. The users can start using one of the two systems for exploration.

6.2.1 SciNet Variant Systems

Sections 3.1.1.1 and 3.1.1.2 describe the user interface details of SciNet and how a user can use it for performing search queries, interpreting the results and keywords and providing feedback to the system. In this thesis we aim to create an extended version of SciNet, the following sections summarize the interaction possibilities of the original SciNet search system and the version which was created in this thesis.

6.2.1.1 Intent Radar Based (Baseline System)

The first system is the original system with Intent Radar based layout. The user can type in their query in the query box and start exploring by seeing the news results in the right side. Based on the initial query, all the important keywords generated on the basis of the search results are visualized on the Intent Radar. The user can explore the different keywords and use them to learn more about the particular topic. The users are able to drag the keywords, which they think are important in context of their search to the center of the Radar and drag the keywords away which are irrelevant. Based on the user feedback, the next search round is able to provide more specific and refined search results back which the user can explore further.

6.2.1.2 Intent Radar + Map Based (Full System)

The new system contains the Global Visualization Map in addition to the Intent Radar. We refer to this system as the full system because it contains all the features listed in our work. Using this system, the users can again type in their query keyword and start searching. The Intent Radar part works in the same way as in described in Section 6.2.1.1 and Chapter 3. Using the map, the users can see the search results being visualized on the map in the form of clusters. We use markers to represent the documents. The users can then hover over the markers to see which news article a particular marker represents. The visualization (Map) may show clusters of similar search results. This can provide the users information about important news events in a particular topic. Then, we have a grid based system on the map which displays

a list of top unigrams and keywords in an area. It also colors the unigrams and grids according to the colors of the keywords from Intent Radar. In this system the user is able to provide feedback in four ways, by clicking on the keywords of a news article in the overall result list, by clicking on the keywords in a pop-up windows of an article selected from the global map, by selecting some of the unigrams from the list that is shown for a selected grid cell, or by using the Radar to give keyword feedback.

6.3 Participants

We sought over 20 participants from different backgrounds to participate in the user studies. The participants were mostly students from the Helsinki Area and Tampere. They were from mixed nationalities and mostly were not native English speakers. They were casual news readers who browse news articles using search engines and news websites regularly. The users are first required to fill a questionnaire with their personal personal details: name, email, gender, native English speaker or not, previously used SciNet system or not. The users who haven't used SciNet before are taken into consideration here. Then, as a part of survey to decide the suitable topic for a user, we ask the users to rate their expertise levels from the given list of news areas. We rank the expertise levels based on a 5 point scale, where 5 means the user is has a lot of knowledge of that area and is an expert, 4 means user has a very good knowledge in that news area, 3 implies that the user has moderate knowledge in that news area, 2 means that the user has some/not a lot of knowledge about that area, and 1 means that the user has absolutely no knowledge in that area. Based on the results we pick news areas with ratings 2-4, that is, areas with which the user is not too familiar with but which are not completely unfamiliar.

The users are first given a demonstration of the usage of the system using suitable examples. The demonstration is done to ensure that the participants can utilize the system for our tasks. The users are given 35 minutes for 1 news area in which they have to explore the system and write down the answers for the questions provided. The total time thus taken for the whole experiment for one user becomes 10 minutes (demonstration) + 35 minutes (news area 1 using baseline system) + 35 minutes (news area 2 using full system) + 10 minutes for user feedback when the user has used the baseline system + 10 minutes for user feedback when the user has used the full system. We tell half of the users to use the Intent Radar based (baseline) system first, followed by the Intent Radar + Map based (full) system. For the rest, we tell them to use the full system first followed by the baseline system. This helps ensure that we get a roughly uniform number of completions for each task, on each system, in each time order (first task of the user or second task of the user). To gather a balanced amount of result data from the different systems and tasks, we

made sure that, over all users, the tasks included each news area on each system an equal amount of times, which means, for example, 'Sports' will be used across all the user experiments for exactly 5 times using the baseline system and 5 times using the full system.

Before starting with the actual experiments, we had performed a feasibility study to check that our selected tasks and other experimental settings were suitable: we chose 2 participants and gave them 2 news areas - "American politics" and "sports". The first participant was supposed to use the baseline system first and then the full system while the second participant was supposed to use the full system first followed by the baseline system.

We logged all the user interactions for future research. A comprehensive list of all the types of logged actions is mentioned in Table 6.2. For each action, the timestamp and relevant details of the action were logged (for eg. query string for typing a query)

Table 6.2: The complete list of logged user interactions

Typing a query to search
Dragging of Keywords on the Intent Radar
Sending feedback using keywords on the Intent Radar
Clicking on the keyword beneath each article
Clicking on the link of the article
Bookmarking / Un-bookmarking an article
Clicking on the Markers to see the article
Sending the feedback from the Global Visualization Map

6.4 Performance Assessment

First, we ask the users to write down their response in an excel sheet shown in Figure 6.1. The excel sheet contains a selected news area for each participant and asks them to write at least 5 main topics for it. Then for at least 2 main topics, the user searches for a main topic and writes at least 2 themes based on his observation. Then, for each theme, the user writes two important news events or news articles from the list of search results which were relevant to the theme. This process is repeated once for the baseline system and then for the full system.

The answers are then graded on a relevancy scale from 0 to 5 (relevance of main topic to the news area of the task, relevance of theme to the main topic, relevance of the news event to the theme), where a rating score of 5 indicates that the answer is fully relevant to the given topic or news area and a rating 0 indicates that the answer is not at all relevant to the given topic or news area.

For the expert grading, all the themes and news events written by all the users were collected in one single sheet. So, the expert does not know from which system

News Area:	
List of main Topics (List 5) -	
Main Topic 1 -	
Theme 1 -	**Theme 2:**
How is it related to given news area:	How is it related to given news area:
News event 1	News event 1:
How is it related to the theme:	How is it related to the theme:
News Event 2	News Event 2:
How is it related to the theme:	How is it related to the theme:
(News Event 3)	(News Event 3)
How is it related to the theme:	How is it related to the theme:
(News Event 4)	(News Event 4)
How is it related to the theme:	How is it related to the theme:
Main Topic 2 -	
Theme 1 -	**Theme 2:**
How is it related to given news area:	How is it related to given news area:
News event 1	News event 1:
How is it related to the theme:	How is it related to the theme:
News Event 2	News Event 2:
How is it related to the theme:	How is it related to the theme:
(News Event 3)	(News Event 3)
How is it related to the theme:	How is it related to the theme:
(News Event 4)	(News Event 4)
How is it related to the theme:	How is it related to the theme:

Figure 6.1: Figure shows an empty excel sheet where the participants were asked to write their responses. The news area was given to them and they had to fill the remaining sections. The text in black color was mandatory and the text in gray color was optional

the answers came from. This eliminates the bias towards a particular system. In order to make the grading process easier, we generalize similar main topics and themes into clusters. For each news area, we would have several main topic clusters. Then, for each main topic cluster, we would have different theme clusters.

For the first part of the grading process, we provide a relevance rating to each theme with respect to its main topic cluster. Next, we provide a relevance to each news event with respect to its theme cluster.

In order to calculate the final results, we calculate the following:

1. **Cumulative gain per main-topic cluster**: The main-topic clusters contain the graded themes corresponding to the individual main topics. For each user we calculated the sum of theme scores, there will be one sum for each main-topic cluster; the score is marked as N.A (not applicable), if no themes were matched for a specific main-topic cluster.

$$CG\left(MTC_m^{a,s}\right) = \sum_n \left(RTheme_n^{m,a,s}\right)$$

Where, $CG\left(MTC_m^{a,s}\right)$ = Cumulative gain per main-topic cluster 'm' for news area 'a' using system 's', $RTheme_n^{m,a,s}$ = Relevance score of news theme 'n' with respect to main-topic cluster 'm' for news area 'a' using system 's'

The total cumulative gain per main-topic cluster (for news area 'a' using system 's') is given as,

$$CG_{total}\left(MTC_{a,s}\right) = \sum_m CG\left(MTC_m^{a,s}\right)$$

2. **Cumulative gain per news-theme cluster and main-topic cluster**: The main-topic clusters contain graded themes and the themes are further clustered into news-theme clusters. The news-theme clusters now contain the themes with their graded news-events. For each user we calculated the total sum of news-event scores multiplied with its theme score, where the multiplication is done in order to emphasize relevant-graded themes in the total sum of news-event scores. There will be one sum for each news-theme cluster; the score is marked as N.A, if no news-theme from that cluster was provided by that user

$$CG\left(TC_{x,m}^{a,s}\right) = \sum_y \left(\sum_z \left(RNewsEvent_{z,y}^{x,a,s}\right) \times RTheme_{y,x}^{m,a,s} \right)$$

Where, $CG\left(TC_x^{a,s}\right)$ = Cumulative gain per news-theme cluster 'x' and main-topic cluster 'm' for news area 'a' using system 's', $RNewsEvent_{z,y}^{x,a,s}$ = Relevance score of news-event 'z' (written for theme 'y') with respect to its news-theme cluster 'x' for news area 'a' using system 's', $RTheme_{y,x}^{m,a,s}$ = Relevance score of news-theme 'y' (under theme cluster 'x') with respect to its main topic cluster 'm' for news area 'a' using system 's'

The total cumulative gain per news-theme cluster and main-topic cluster (for

news area 'a' using system 's') is given as,

$$CG_{total}\left(TC_{a,s}\right) = \sum_{x} CG\left(TC_{x,m}^{a,s}\right)$$

6.5 User Feedback

We also took a user feedback in the form of a questionnaire at the end of the user experiment for each of the two systems. We provided a defined set of statements and questions to user, for which they had to select a score on a scale of 1 to 5. A score of 5 means that the user strongly agrees with the statement and a score of 1 means that the user strongly disagrees with the statement. Based on the results, we listed means and standard deviations of the user score for each of the questions asked and generated histograms for the same.

7. RESULTS AND DISCUSSION

The results of the thesis have been computed on two different levels. First, we grade the answers written by the participants of the user experiments and compute the different scores as described in section 6.4.

Table 7.1 shows the cumulative gains for the main-topic clusters i.e. the relevance scores of news-themes with respect to their main-topic clusters for a particular news-area using a specific system. The horizontal label denotes the news-areas and the vertical label denotes the system (Radar is the baseline system and Radar+Map is the full system)

Table 7.2 in turn shows cumulative gains for news-theme clusters and main-topic clusters i.e. the relevance scores of news-events with respect to their news-theme clusters, multiplied by the score of the news-theme with respect to its corresponding main-topic cluster, for a particular news-area using a specific system

Based on the scores from table 7.1, we can calculate the total sums for both the systems across all news areas, and averages per news area.

Similarly, Based on the scores from table 7.2, we can calculate the total sums for both the systems across all news areas, and the averages per news area.

We can clearly see in both the tables, that the Radar+Map performs better with an average of **91.5** vs 84.75 for Radar only in table 7.1 and an average of **874.75** vs 752.25 for Radar only in table 7.2. Radar + Map performs better in particular for 3 news-areas (American Politics, Entertainment, Sports). The margin of difference is considerably higher in table 7.2 where we conducted more in-depth grading. By the analysis of the scores, we saw that the full system i.e. Radar+Map has an advantage over the baseline system i.e. Radar.

The next level of computation for results was done using the feedback taken in the form of a questionnaire by the users who participated in the user experiments.

Table 7.1: Cumulative gains for the main-topic clusters : Relevance scores of news-themes with respect to their main-topic clusters for a particular news-area using a specific system

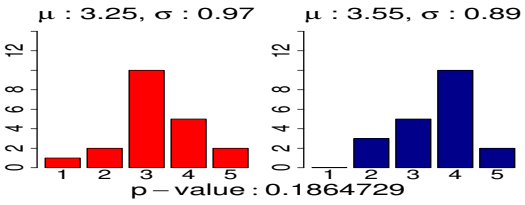
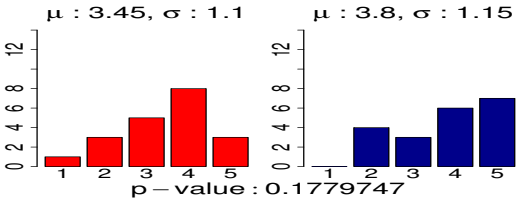
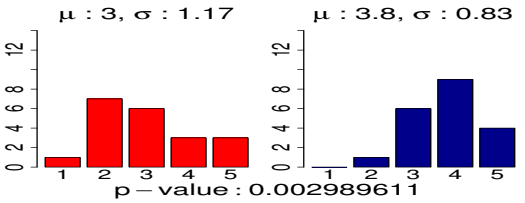
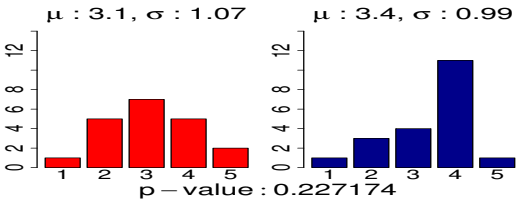
	Finance	American Politics	Entertainment	Sports	Total	Average
Radar	97	68	96	78	339	84.75
Radar+Map	86	**83**	**99**	**98**	**366**	**91.5**

Table 7.2: Cumulative gains for news-theme clusters and main-topic clusters : Relevance scores of news-events with respect to their news-theme clusters, multiplied by the score of the news-theme with respect to its corresponding main-topic cluster, for a particular news-area using a specific system

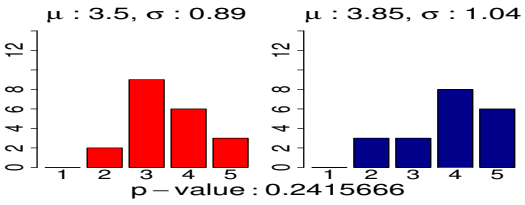
	Finance	American Politics	Entertainment	Sports	Total	Average
Radar	909	666	880	554	3009	752.25
Radar+Map	846	**815**	**942**	**896**	**3499**	**874.75**

Based on the feedback, we summarized the distribution of the resulting scores in the form of histograms. We then used a paired version of the Wilcoxon signed-rank test [48] to test the statistical significance of the difference between the results of the two systems for each of the questions. The significance threshold for the p-value is set at 0.05.

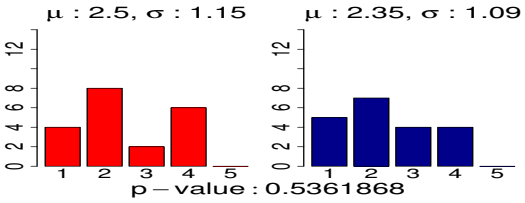
All questions are listed in Table 7.3 showing histograms for the distributions of answers for both the systems, including the mean and standard deviation and the p-value of the difference. Questions with $p<0.05$ are shown in bold.

Statement / Question	Radar (Red) vs Radar+Map (Blue)
1. This system provides adequate way to express preferences	
2. This system provides adequate support to revise preferences	
3. This system helps me to understand why the suggested articles should be important	
4. The information provided by the system is sufficient to make decisions	

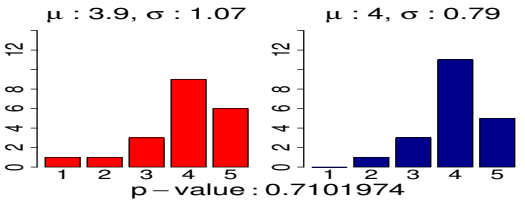
5. The labels / keywords / information provided by the system are clear



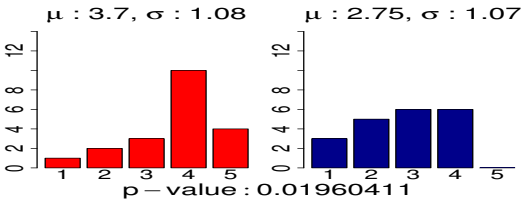
6. The layout of the system is not very clear



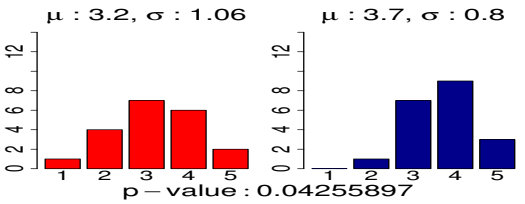
7. I learnt to use the system quickly



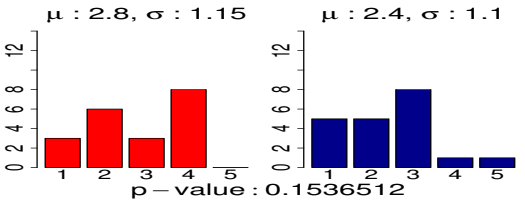
8. It took too much effort to find useful articles



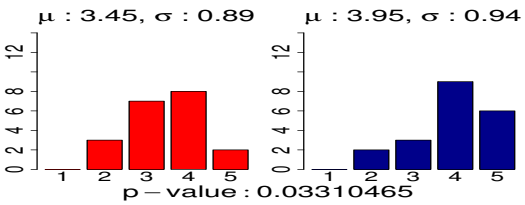
9. I found it easy to express information need and preferences



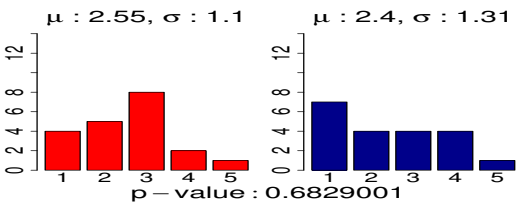
10. I found it difficult to train the system with updated preferences



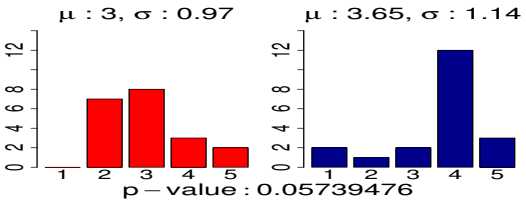
11. With this system it is easy to alter the outcome of results



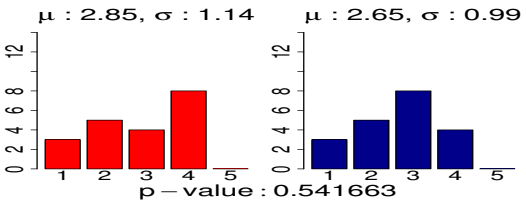
12. It is difficult to get new set of items instead of what I already have



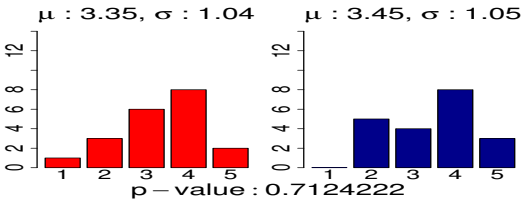
13. The system offered me useful options and avoided me from getting stuck when I could not think of a proper query to express information need



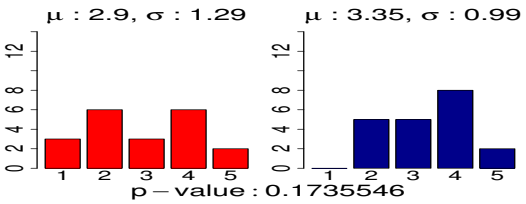
14. I found it difficult to explore the related areas without getting stuck



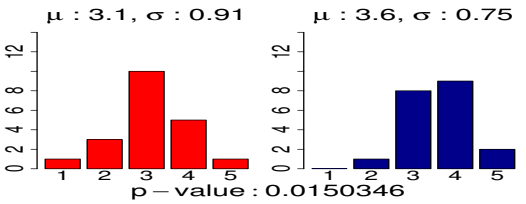
15. I feel in control to tell what I want



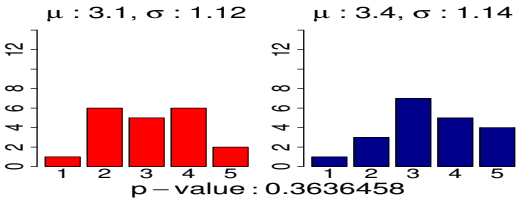
16. The system helps me to understand and keep track of why the items were relevant and offered for me



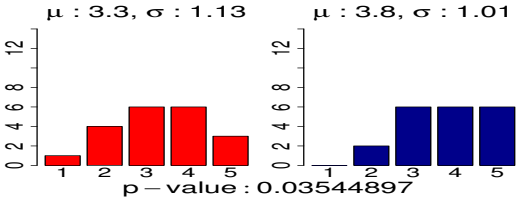
17. I'm satisfied with the system



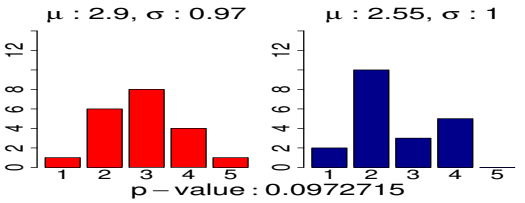
18. I am convinced that I found the right articles



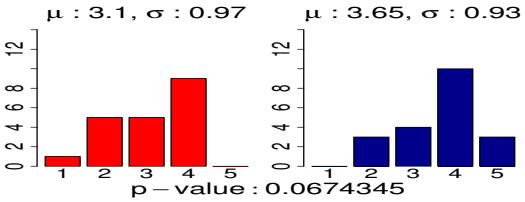
19. I would like to use the system, if offered for me



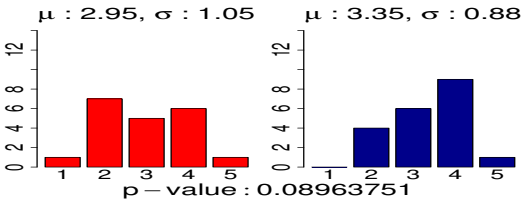
20. With this system it is difficult to find answers to my information needs



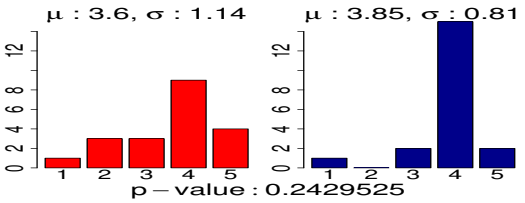
21. I was able to take advantage of the system easily



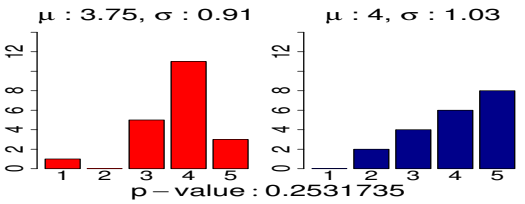
22. I quickly became productive by using the system



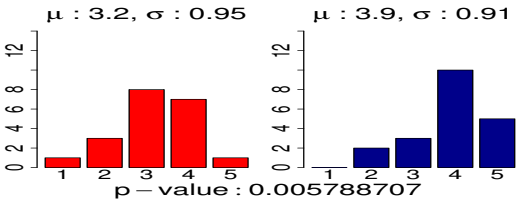
23. The system influenced my choice of items



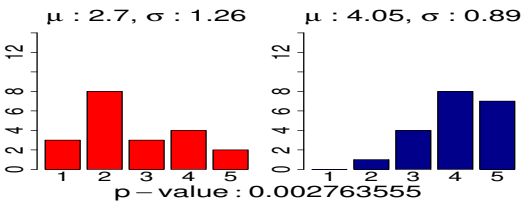
24. The system helps me to get an overview of the available information



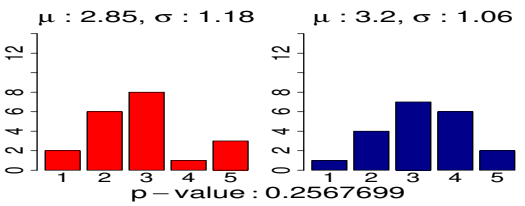
25. I felt I was able to explore the available articles



26. The system helps me to understand which articles are related to each other



27. I feel I achieved a comprehensive understanding of the articles



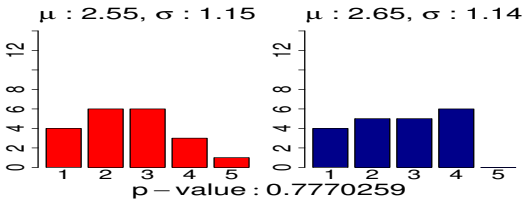
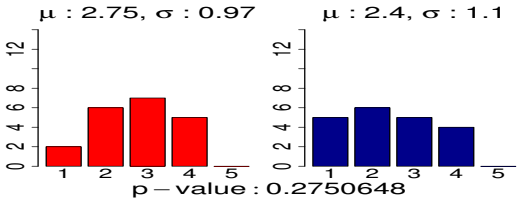
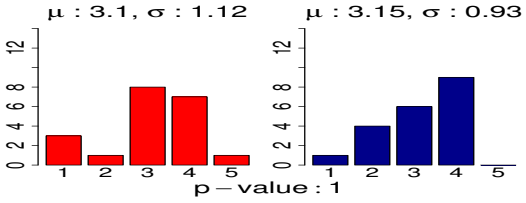
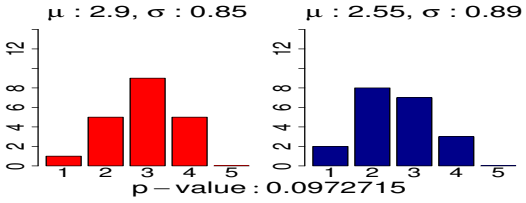
28. I did not feel supported by the system to find what i like	
29. I found it difficult to know what is available in the news.	
30. I felt very confident using the system	
31. How well you knew the topic of this task before?	

Table 7.3: Side by Side comparison of the Radar and Radar+Map based systems based on the user feedback obtained from the participants who performed the user experiments. The entries for which the p-value is less than 0.05 are highlighted in bold.

We selected the entries for which the p-value is less than the threshold value of 0.05 in order to make sure that we focus our analysis on the comparisons that are statistically significant. Table 7.4 shows the list of selected statements.

As we can see in table 7.4, the Radar+Map has a clear advantage over Radar, all the statements that show statistically significant difference are in favor of the Radar+Map. This along with our previous result from relevance grading of the user's answers, shows that the Radar+Map clearly helps the user in finding the relevant results as well as improving their user experience.

No.	Statement / Question	Mean (Radar)	Mean (Radar + Map)
3.	This system helps me to understand why the suggested articles should be important	3.0	**3.8**
8.	It took too much effort to find useful articles (negative statement)	3.7	**2.75**
9.	I found it easy to express information need and preferences	3.2	**3.7**
11.	With this system it is easy to alter the outcome of results	3.45	**3.95**
17.	I'm satisfied with the system	3.1	**3.6**
19.	I would like to use the system, if offered for me	3.3	**3.8**
25.	I felt I was able to explore the available articles	3.2	**3.9**
26.	The system helps me to understand which articles are related to each other	2.7	**4.05**

Table 7.4: Selected statements from the list of feedback results (table 7.3) where the p-value is less than 0.05, showing there is a statistically significant difference between the results of Radar based survey and Radar+Map based survey

8. CONCLUSIONS AND FUTURE PROSPECTS

Considering the large amounts of data present everywhere these days, the task of information seeking becomes important and necessary, in order to explore and find the information needed across a variety of different domains. In our thesis we extended the original SciNet system to make it run for large set of news articles which are published online. We contributed to the system by adding new features to the user interface which helps in exploring the news articles better. The new interactive exploratory search system now supports searching via the original Intent Radar and the added Global Visualization Map.

There were a series of user experiments carried out to test the performance of the new system. From the experiments we can conclude that preliminary results suggest that our interactive map serves as a useful aid to the users in finding the subtopics or important news events of a particular topic. It could be seen that by using the Global Visualization Map with Intent Radar, the cumulative gains for a specific news area improved, in most cases. We would also like to mention that we logged every participant's actions while they were performing searches during user experiments. The list of all user interactions logged could be seen in Table 6.2. They could be used in future work to analyze user patterns when interacting with our system.

Based on the answers written by the users who participated in the user experiments, we calculated the relevance scores and on comparison, found that the full system (Map) performs better than the baseline system (Radar). To confirm the same, we also compared the scores of feedback taken from the users. This shows, that the visual aids based on our research improved the experience of SciNet for the users and helped them to explore more relevant news articles for their search queries.

However, the performance can be significantly be improved further in order to make the system even more useful. There are few challenges which need to be overcome. Currently, the keyword extractor used - Maui produces a lot of general keywords in context to the given topic. It would be beneficial to have a trained keyword extractor for news articles in order to produce more specific keywords. Next, we could have better web crawling mechanisms which are able to discard garbage articles like advertisements or ambiguous articles which talk about multiple things.

Currently, the image used for the Map is static and generated once via MATLAB for a large amount of data. If we are able to explore quicker algorithms which can run on distributed environments for our use case, then it could be beneficial for a real time data. Right now, this kind of a process will have to be run periodically to update the image used for the Map as the corpus is updated. Next, with the current system it is possible to see the clusters of the search results with the help of document markers, as well as clusters of documents in the background image that are not in the current search result, it would be even more helpful if a cluster could be tagged with a keyword which summarizes the documents contained within that region.

Overall, we conclude that our work holds a lot of potential for further research in this field of exploratory search. It also opens up possibilities for developing novel techniques when it comes to exploring and investigating large data sets.

REFERENCES

[1] Ruotsalo, T., Athukorala, K., Głowacka, D., Konyushkova, K., Oulasvirta, A., Kaipiainen, S., Kaski, S. and Jacucci, G., 2013. Supporting exploratory search tasks with interactive user modeling. Proceedings of the Association for Information Science and Technology, 50(1), pp.1-10.

[2] Ruotsalo, T., Peltonen, J., Eugster, M.J., Glowacka, D., Reijonen, A., Jacucci, G., Myllymäki, P. and Kaski, S., 2014, April. Intentradar: search user interface that anticipates user's search intents. In CHI'14 Extended Abstracts on Human Factors in Computing Systems (pp. 455-458). ACM.

[3] Ruotsalo, T., Peltonen, J., Eugster, M.J., Głowacka, D., Jacucci, G. and Reijonen, A., Lost in Publications? How to Find Your Way in 50 Million Scientific Documents.

[4] Kangasrääsiö, A., Głowacka, D., Ruotsalo, T., Peltonen, J., Eugster, M.J., Konyushkova, K., Athukorala, K., Kosunen, I., Reijonen, A., Myllymäki, P. and Jacucci, G., 2014. Interactive Visualization of Search Intent for Exploratory Information Retrieval. In ICML 2014 workshop "Crowdsourcing and Human Computing.

[5] Marchionini, G., 2006. Exploratory search: from finding to understanding. Communications of the ACM, 49(4), pp.41-46.

[6] Term Frequency - Inverse Document Frequency `www.tfidf.com` Retrieved 05/01/2015

[7] Schütze, H., Manning, C.D. and Raghavan, P., 2008. Introduction to information retrieval (Vol. 39). Cambridge University Press, pp. 118-128

[8] Paltoglou, G. and Thelwall, M., 2010, July. A study of information retrieval weighting schemes for sentiment analysis. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 1386-1395). Association for Computational Linguistics.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825-2830.

[10] Schütze, H., Manning, C.D. and Raghavan, P., 2008. Introduction to information retrieval (Vol. 39). Cambridge University Press, pp. 120-124

[11] Muja, M. and Lowe, D.G., 2014. Scalable nearest neighbor algorithms for high dimensional data. IEEE transactions on pattern analysis and machine intelligence, 36(11), pp.2227-2240.

[12] Principal Components Analysis http://www.jmp.com/support/help/Overview_of_Principal_Component_Analysis.shtml Retrieved 01/02/2015

[13] Halko, N., Martinsson, P.G. and Tropp, J.A., 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions.

[14] Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), pp.100-108.

[15] Weisstein, Eric W. "Singular Value Decomposition." From MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/SingularValueDecomposition.html Retrieved 10/10/2016

[16] Landauer, T.K., Foltz, P.W. and Laham, D., 1998. An introduction to latent semantic analysis. Discourse processes, 25(2-3), pp.259-284.

[17] Bingham, E. and Mannila, H., 2001, August. Random projection in dimensionality reduction: applications to image and text data. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 245-250). ACM.

[18] Multidimensional Scaling http://forrest.psych.unc.edu/teaching/p208a/mds/mds.html Retrieved 10/10/2017

[19] Self Organizing Maps http://davis.wpi.edu/~matt/courses/soms/ Retrieved 10/10/2017

[20] LDA http://chem-eng.utoronto.ca/~datamining/dmc/lda.htm Retrieved 10/10/2017

[21] M-Brain - Global Business & Market Intelligence www.m-brain.com Retrieved 10/1/2015

[22] Revolution of Knowledge Work www.reknow.fi Retrieved 10/1/2015.

[23] Medelyan, O., Frank, E. and Witten, I.H., 2009, August. Human-competitive tagging using automatic keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3 (pp. 1318-1327). Association for Computational Linguistics.

[24] El-Beltagy, S.R. and Rafea, A., 2009. KP-Miner: A keyphrase extraction system for English and Arabic documents. Information Systems, 34(1), pp. 132-144.

[25] Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C. and Frank, E., 1999. Improving browsing in digital libraries with keyphrase indexes. Decision Support Systems, 27(1-2), pp. 81-104.

[26] Liu, F., Liu, F. and Liu, Y., 2008, December. Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In Spoken Language Technology Workshop, 2008. SLT 2008. IEEE (pp. 181-184). IEEE.

[27] Wan, X., Yang, J. and Xiao, J., 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In Proceedings of the 45th annual meeting of the association of computational linguistics (pp. 552-559).

[28] Van Der Plas, L., Pallotta, V., Rajman, M. and Ghorbel, H., 2004. Automatic keyword extraction from spoken text. a comparison of two lexical resources: the EDR and WordNet. arXiv preprint cs/0410062.

[29] Fielding, R.T. and Taylor, R.N., 2000. Architectural styles and the design of network-based software architectures (Vol. 7). Doctoral dissertation: University of California, Irvine, pp. 76-105

[30] Apache Lucene `http://lucene.apache.org/core/` Retrieved 14/01/2015

[31] MongoDB for GIANT ideas `http://mongodb.com` Retrieved 01/10/2017

[32] Wikipedia Miner. http://wikipedia-miner.cms.waikato.ac.nz/ . Retrieved 15/02/2015

[33] Hearst, M., 2009. Search user interfaces. Cambridge University Press.

[34] Diriye, A., Wilson, M.L., Blandford, A. and Tombros, A., 2010. Revisiting exploratory search from the HCI perspective. HCIR 2010, p.99.

[35] White, R.W., Marchionini, G. and Muresan, G., 2008. Evaluating exploratory search systems: Introduction to special topic issue of information processing and management.

[36] Cook, K.A. and Thomas, J.J., 2005. Illuminating the path: The research and development agenda for visual analytics.

[37] Bjork, S. and Redstrom, J., 2000. Redefining the focus and context of focus+ context visualization. In Information Visualization, 2000. InfoVis 2000. IEEE Symposium on (pp. 85-89). IEEE.

[38] Shneiderman, B., 2003. The eyes have it: A task by data type taxonomy for information visualizations. In The Craft of Information Visualization (pp. 364-371).

[39] Muja, M. and Lowe, D.G., 2014. Scalable nearest neighbor algorithms for high dimensional data. IEEE transactions on pattern analysis and machine intelligence, 36(11), pp.2227-2240.

[40] FLANN - Fast Library for Approximate Nearest Neighbors `http://www.cs.ubc.ca/research/flann/` . Retrieved 15/02/2015

[41] Yang, Z., Peltonen, J. and Kaski, S., 2013, February. Scalable optimization of neighbor embedding for visualization. In International Conference on Machine Learning (pp. 127-135).

[42] Cauchy Distribution `http://mathworld.wolfram.com/CauchyDistribution.html` Retrieved 10/05/2017

[43] Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.

[44] KL Divergence `http://www.cs.buap.mx/~dpinto/research/CICLing07_1/Pinto06c/node2.html` Retrieved 01/02/2015

[45] Järvelin, K. and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS), 20(4), pp.422-446.

[46] Liu, T.Y., 2009. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval, 3(3), pp.225-331.

[47] Wilcoxon Signed-Rank Test using SPSS Statistics: `https://statistics.laerd.com/spss-tutorials/wilcoxon-signed-rank-test-using-spss-statistics.php` Retrieved 22/04/2017

[48] Wilcoxon Signed-Rank Test `http://www.r-tutor.com/elementary-statistics/non-parametric-methods/wilcoxon-signed-rank-test` Retrieved 22/04/2017

[49] Paired Sample T-test - Statistics Solutions http://www.statisticssolutions.com/manova-analysis-paired-sample-t-test Retrieved 28/04/2017

[50] Frakes, W.B. and Baeza-Yates, R. eds., 1992. Information retrieval: Data structures & algorithms (Vol. 331). Englewood Cliffs, New Jersey: prentice Hall.

[51] Spring Framework http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html Retrieved 16/01/2015

[52] Model-View-Controller http://c2.com/cgi/wiki?ModelViewController Retrieved 16/01/2016

[53] The csv summary files https://github.com/dnmilne/wikipediaminer/wiki/The-csv-summary-files Retrieved 11/02/2016.

[54] Cavnar, W.B. and Trenkle, J.M., 1994. N-gram-based text categorization. Ann arbor mi, 48113(2), pp.161-175.

[55] Snowball: A language for stemming algorithms. http://snowballstem.org/ Retrieved 01/10/2017

[56] Evaluation of Unranked Retrievel Sets http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-unranked-retrieval-sets-1.html Retrieved 20/01/2015

[57] Hughes, G., 1968. On the mean accuracy of statistical pattern recognizers. IEEE transactions on information theory, 14(1), pp.55-63.