

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Masterarbeit

Visual Tracking and Analysis of Web Content Dissemination

Leslie Tso

Course of Study:	Softwaretechnik
Examiner:	Prof. Dr. Thomas Ertl
Supervisor:	Dr. Dennis Thom, Johannes Knittel, M.Sc.
Commenced:	February 6, 2019
Completed:	August 6, 2019

Abstract

Due to the rising popularity and necessity of information today, it stands to reason that the enormous amount of information needs to be filtered and organized in order for humans to quickly and accurately retrieve the most important information from it. Furthermore, it is also important to track the changes in the information to discover how information about specific topics change over time. This thesis focuses on assessing and evaluating possible machine-learning algorithms in order to help automatically determine the similarity of documents and topics as well as visualization methods that allow the user to intuitively and accurately retrieve and track news article topics across multiple documents. Based on the evaluation of said machine-learning algorithms and visualization methods, a system using fundamental visualizations to promote understandability and the tracking of relationships between words and articles at the expense of requiring more user interaction was proposed. The proposed system has the main goal of helping analysts determine the significance and validity of textual content from multiple documents and sources as well as help determine other relevant documents and the possible origin of specific news article content. The system was then be evaluated through a user study where half the participants used a basic search-engine-based system and the other half used the proposed system. The results of the study was used to assess whether the proposed system can be used as an effective and efficient way for analysts and journalists to discover the relationships between different articles as well as track the provenance and evolution of the topics over time. From the results of the study, the participants using the proposed system did significantly better in terms of time and correctness of the answers in comparison to the participants who used a search-engine-based system.

Contents

1	Introduction	13
2	Related Work	15
2.1	Summary	20
3	Modeling	23
3.1	Theory	23
3.2	Design Approach	35
3.3	Design Implementation	45
3.4	Summary	53
4	Study	55
4.1	Participants	55
4.2	Apparatus	56
4.3	Tasks	56
4.4	Procedure	59
4.5	Difference between Control and Treatment Groups	60
4.6	Summary	60
5	Study Evaluation	61
5.1	Results	61
5.2	Discussion	76
5.3	Summary	81
6	Conclusion	83
	Bibliography	85

List of Figures

2.1	TopicStream Visualization	18
2.2	StoryFlow Layout Pipeline	19
3.1	Word2Vec Model	26
3.2	Doc2Vec Paragraph Vector	27
3.3	Doc2Vec Paragraph Vector wit Bag-of-Words	27
3.4	Stacked Bar Visualization in Proposed System	42
3.5	Ring-Network visualization in Proposed System	44
3.6	Heat Map Visualization in Proposed System	45
3.7	Bubble Map Visualization in Proposed System	46
3.8	UML Activity Diagram for the Proposed System	47
3.9	Proposed System - First Layer	50
3.10	Proposed System - Second Layer	51
3.11	Proposed System - Third Layer	52
3.12	Proposed System - Fourth Layer - Details	53
5.1	Total Times Required for Task Set 1	62
5.2	Times Required for Task 1	63
5.3	Times Required for Task 2	64
5.4	Times Required for Task 3	65
5.5	Times Required for Task 4	67
5.6	Times Required for Task 5	68
5.7	Times Required for Task 6	69
5.8	Times Required for Task 7	70
5.9	Times Required for Task 8	71
5.10	Times Required for Task 9	72
5.11	Times Required for Task 10	73
5.12	Likert-Scale Survey Results	76
5.13	Average Time per Task (s)	77
5.14	Average Time per Task (%)	78
5.15	Combined Scores from Both Groups	79

List of Tables

5.1	Table of NASA-TLX Results	75
-----	-------------------------------------	----

List of Abbreviations

IDF Inverse Document Frequency. 24

LDA Latent Dirichlet Allocation. 29

LSA Latent Semantic Analysis. 35

TF Term Frequency. 24

1 Introduction

Since the beginning of human history, information has played an important role in society ranging from farming knowledge to astrophysics. In fact it could be said that information is currently our largest and most important resource. For many today, the main way they receive information is through the internet. This especially holds true for current news in particular, whether it comes from traditional digital news sources such as online newspapers and magazines to more unconventional news sources such as YouTube¹ and blogs. However, due the massive wealth of information we have at our fingertips today together with the various opinions and interests of the topics contained in the articles from multiple news sources, it can be incredibly hard to filter and search for relevant data as well as understand the information from all the data.

This raises the question: Is there a way to automatically summarize, organize and present the information contained in the data in order to massively reduce the workload of the analyst? One way to solve this problem is through information visualization. The visualization of relevant information is one of the most important and basic methods to present interesting and important information to the viewer due to the high relationship between perception and cognition [War12]. By visualizing the information, humans can easily detect patterns and understand otherwise complex information. This is possible as visualizations can be used to present patterns found in the data or summarize the data for the viewer so that they do not need to manually do it themselves. This can be especially useful when an analyst needs to track the evolution of data or just get an understanding of data sets containing hundreds or even millions of documents. The ability to track how data changes over time is very important for users such as business and news analysts to help predict future trends as well as understand why the data has changed. However, while there are projects such as TopicStream [LYW+15] which shows the changes in the grouping of topics over time and the paper by Alshaymi [Ali17] which focuses on tracking the provenance of topics and articles, none focus on the ability to use both features simultaneously. By using both types of features, an analyst would be able to determine the starting point for a or for a combination of topics and quickly and intuitively determine the differences or similarities between past topics and documents.

The objective of this thesis is to develop methods to identify, extract and visually consolidate related pieces of information from online sources with a focus on news. Recent works on word- and document-embedding algorithms such as Word2Vec [Mik13] and Doc2Vec [LM14] in conjunction with various similarity functions have proven that machine-learning embedding algorithms have arrived at a point where it can be seriously considered to help analysts break-down and organize textual information. To provide a platform allow analysts to identify topic and keyword trends as well as the ability to determine the provenance of topics from multiple sources and track the evolution of the topics over time, the information from the automatic analysis of the documents and the thorough use of visualizations and interactions was combined. To fulfill these goals, a system

¹<https://www.youtube.com>, last visited July 28, 2019.

to automatically extract the content of the data as well as the relationships between the various content was proposed and implemented. With the extracted information, the system automatically creates visual tales which can be further explored using interactions while taking into account the relationships between the spatial and temporal facets of the data, especially in context of discovering the provenance and tracking the change in topics overtime. Using a user study, the proposed system was evaluated on whether it can effectively and efficiently allow an analyst to analyze a large set of data and perform real-world-based tasks through the use of topic modeling in conjunction with visualizations. Our contributions are as follows: i) analysis and evaluation of existing methods used for the visual analysis of documents; ii) implementation, improvement and selection of current word and document modeling algorithms and foundational visualizations to be used as components of the proposed system; iii) conduct empirical evaluations on the methods proposed in ii); and iv) determine the effectiveness and usability of said methods to create a software-based tool for the visual analysis of news articles.

Outline

The thesis is structured the following way:

Chapter 2 – Related Work: This chapter describes previous papers and projects which aim to use document embedding and interactive visualizations to automatically simplify and organize data for the users.

Chapter 3 – Modeling: This chapter explains the theory and the implementation of the methods used for the study.

Chapter 4 – Study: This chapter describes the conducted study. Here the setup, limitations and study procedure are described.

Chapter 5 – Study Evaluation: This chapter illustrates the results from the study as well as provides a brief explanation of the correlation between the different results.

Chapter 6 – Conclusion Last of all, this chapter summarizes the thesis as well as provides possible future steps and applications for the system.

2 Related Work

Due to the importance of information gathering and the ever increasing amount of information we have at our fingertips that needs to be sifted through in order to obtain the necessary information, many developers and researchers have been developing and investigating new methods to help analysts do so quickly and efficiently with minimal human effort. These developments have proven valuable in our research with many of these projects influencing the direction of this thesis.

One work that heavily inspired this thesis was the paper by Wang et al. [WLC+16]. In their paper, they proposed a system called IdeaFlow with the main goal of tracking the relationships between ideas between different social groups. By tracking these relationships, the proposed system is able to help analysts understand how the information, opinions and thoughts change as they pass through different social groups. Using IdeaFlow, the authors attempted to solve two major challenges: 1) modeling ideas and tracking their lead-lag relationship over time [WLC+16]; and 2) designing a visualization which can be analyzed by the analyst without any training and can describe the evolution of ideas over time. To model the ideas, they proposed using a random-walk-based correlation model which they then used in conjunction with Bayesian conditional co-integration and tensor-based machine-learning. The entire process of IdeaFlow can be summed into four main steps. The first step is to use Bayesian conditional co-integration to determine the association between word time series. The system then uses the said associations as input for the random-walk-based correlation model in order to consolidate the relationships between the word time series, tweet content and meta-date in order to discover the word-time relationship with the best accuracy. Afterwards, tensor decomposition is used to cluster important words as a basic skeleton of each idea and combined into a single element which shows how the words change over time to determine how the idea changes over time. This information is then visualized through a novel visualization which attempts to combine the advantages of Voroi-treemap-based bubble trees, flow maps as well as timelines in order to allow the analysts to easily and efficiently gain an outline of the various ideas. A layout algorithm based on correlated-clustering is then used in order to reduce the ambivalence within the flow of ideas as well as give the system the ability to generate and show multiple flows of ideas in parallel. In order to show the details of an idea while allowing the analysts to see the entire timeline as a whole, the authors decided to use a focus+context interaction design pattern in their timeline. To validate their concept, the authors ran a quantitative evaluation of their system and two case studies. With the quantitative evaluation, two different Twitter data sets were used. The first dataset consisted of nearly 2 million tweets posted by over a thousand Twitter accounts from the members of the 114th U.S. Congress. Each account was then manually classified into four categories: Democratic or Republican Senators or House Representatives. The second data set contained nearly 17 million tweets regarding “Ebola” from more than 320 thousand accounts between July 2014 and February 2016 which were classified based on their tweet locations. Using a machine with an Intel Xeon E52630 processor and 64GB RAM, the authors obtained 300 ideas and 27,802 lead-lag relationships from the first data set and 100 ideas and 2,568 lead-lag relationships from the second dataset which took 17 and 9 hours respectively. The experiment population was consisted

of two PhD students majoring in data mining and were experienced with the two data sets so they themselves could determine if the ideas were classified correctly. The results of the study were then compared to two baselines. The first baseline was developed by Zhong et al. [ZLW+16] which also makes use of co-integration and tensor representation to track the evolution of ideas. However, as the method proposed by Zhong et al. [ZLW+16] did not analyze the lead-lag relationships between ideas, the authors exchanged the tensor representation algorithm with the one developed for IdeaFlow. The second baseline was simply IdeaFlow without the use of co-integration between the word time series in order to determine the effect the co-integration has on the overall accuracy of the system. From their quantitative evaluation, the authors determined that IdeaFlow was a step forward in terms of idea and lead-lag flow analysis. Across both data sets, IdeaFlow proper had an average accuracy of 86.65% when tracking the flow of ideas and 80.15% when tracking the evolution of lead-lag relationships. In contrast, IdeaFlow without co-integration came in second place with an average accuracy of 79.15% and 75% for idea and lead-lag tracking respectively for both data sets. In last place was the proposed system by Zhong et al. [ZLW+16] with a significantly lower accuracy for 66.65% and 63.65% for idea and lead-lag tracking respectively. As for the case studies, The first case study was conducted with a professor of media and communications using the first data set containing tweets from the members of the 114th US Congress. Using IdeaFlow the professor was able to identify various clusters of ideas and how each idea had a different patterns of evolution as well as strengths and weaknesses in every social group. The second case study was conducted with a different professor using the second data set. Here, the professor was quickly follow the history and different phases' of health crises within the specific time period of the data set. However, it should be noted that due to the needs of the professor, the authors implemented the word-embedding-based sentiment classification method from Tang et al. [TWY+14].

The primary goal of the paper by Leskovec et al. [LBK09] is to track topics, ideas and memes pulled from multiple data sources. To do so, the authors developed a framework which learns short phrases with unique characteristics and track how they change over time. With their approach, the framework attempts to automatically find a set of unique phrases which diversify over time and track these changes. Using this set of phrases, the framework can split a topic into multiple collections of threads or memes. However, discovering and clustering variants of distinct phrases can prove to be a challenge. To solve this issue, the authors first attempted to transform the set of phrases into a direct phrase graph where each phrase is a node and the directed edges determine if a phrase is a variant of a previous phrase. The system then attempted to remove the lowest number of edges, so that each node is connected to a node with no outgoing edges somewhere along the line. To cluster the phrases, the framework compares the phrases in the collections to a longer phrase or collection of phrases and clusters the phrases to the longer phrase or other collection if they pass a similarity threshold. The clusters are then visualized in a stacked area chart to show the topics in a particular time period. The individual clusters are visualized as threads or rather a stack on the area chart and the amount of times the phrase appears in a time period can be tracked.

In the paper by Liu et al. [LYW+15], the authors proposed a system called TopicStream with the main goal of helping analysts explore and interpret how topics evolve over time. To do so, the application can be generalized into two foundational steps. First of all, with a set of topic trees and a stream of documents as input, the authors used a dynamic Bayesian network model to create new tree cuts in increments. For this to work, the analyst manually selects nodes containing information they may be interested in. Afterwards, based on the selection, one or more "tree cuts" or a set of tree nodes containing a layer of topics that the analyst is interested in based on the nodes they selected are generated from the streaming tree cut algorithm from the authors. Generally, a tree cut

contains the set of nodes from the root of the tree to the child node that the analyst has marked as interesting. One major challenge of this proposed method is the fact that the system is constantly receiving information in real time. This is an issue as it is very time-consuming to determine every tree cut every time a new document is received which is an issue when receiving information in real-time. Furthermore, every time a new document is received, existing tree cuts may be altered based on the information from the new documents. To attempt to solve this problem, the authors use the aforementioned dynamic Bayesian network to hypothesize all future possible tree cuts from the new incoming documents. In the second step, a time-based visualization is used to display the information from the resulting tree cuts. The authors had three main requirements for the visualization: 1) provide an overview of the entire text stream; 2) visualize how new documents from the text stream will merge with the existing ones already processed in the system; and 3) have the ability to allow analysts to compare the content of documents from different time periods. The visualization itself can be split into four sections as can be seen in Figure 2.1. The left-most area in the visualization is called “Archive”. As its name implies, it contains the k time steps oldest topics and documents of the stream set by the analyst. As this information might no longer be as important to the analyst in comparison with the newer documents from the stream, the area is visualized as a stacked bar chart to maximize the usage of space while keeping the area required relatively small and reduce clutter. In the stacked bar itself, every bar represents a topic with the height representing the average number of documents in that archived time step. The “Stack” area of the visualization shows the older topics which are next to be transferred into the archive. Here only the main stream of each topic in that time period is shown. The “River” region displays the recent topics as well as how they split off and merge into new or existing topics. The right-most area on the graph labeled “Streaming” simply contains all the topics from the documents which have just been streamed in. To illustrate the effectiveness of TopicStream, the authors did two case studies with real-word scenarios and datasets. For the purposes of the case studies, two datasets were used: 1) a news and tweet dataset containing 15,558,371 documents regarding “Ebola”; and 2) news dataset containing 495,151 documents regarding “Obama”. The first study using the Ebola dataset was done with a professor who focused on public opinion analysis in health care. Using the proposed system the participant was able to quickly find the different new topics in the stream based on color and whitespaces. The participant then was able to follow a stream containing a topic of particular interest to determine the evolution of the topic over time and how new topics were generated from it. The second case study was conducted with a professor of media and communications in conjunction with the Obama dataset. The results from this case study were quite similar to that of the first other than the type and content of the topics. The participant was quickly able to determine the different topics and from the splitting and merging of the topic streams, was able to analyze and interpret the reasons for topics such as “low economic confidence index”.

Rather than focusing on tracking the relationship between documents to determine the evolution of topics over time, the paper by Alshaymi [Ali17] focuses on using the relationship and similarity between documents to determine the reliability of news articles as well as whether the article has been partially or fully plagiarized. The proposed system can be split in to four main steps. Given a search query made of keywords, titles or document contents, in the first step is collecting relevant news articles across all online news sources using the Topic Detection and Tracking technique used in Google Custom Search¹ and the Google Search API². The second stage is to discover the

¹<https://cse.google.com/cse/>, last visited July 8, 2019.

²<https://developers.google.com/custom-search/>, last visited July 8, 2019.

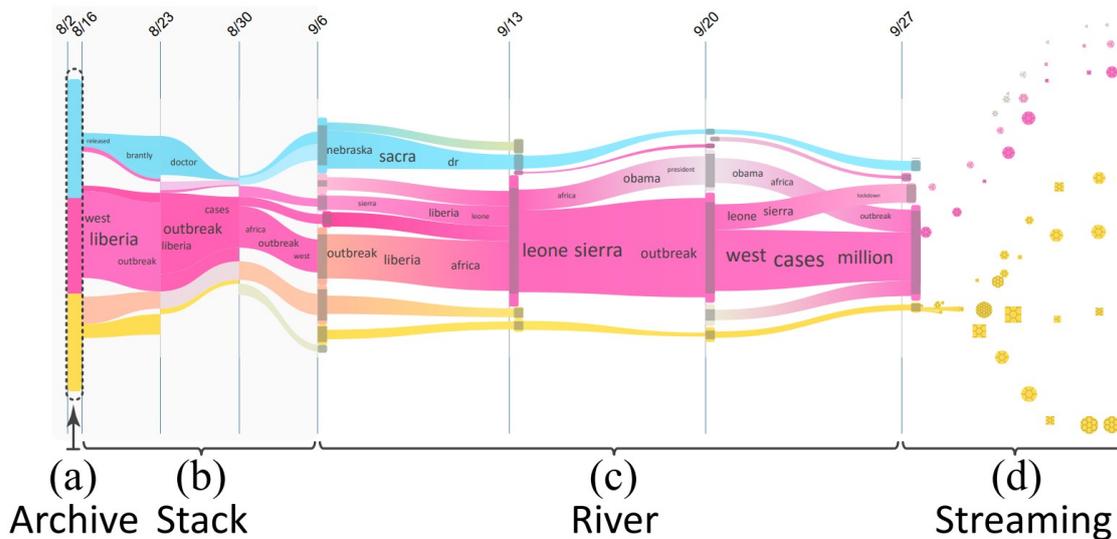


Figure 2.1: Time-based visualization proposed by Liu et al. [LYW+15]. The x-axis denotes the time and the visualization itself is split into four main sections depending on the difference in times [LYW+15].

oldest article in the collection of relevant articles which is assumed by the author to be the origin article in the collection of articles. The other articles in the collection are then compared with the origin article using the Google API to determine their similarity and articles with a similarity over a certain threshold are tagged with “Plagiarism”. Finally, the results are displayed in a table format from the oldest to newest article regarding the search query.

As with the previous two papers mentioned in this chapter, the paper by Liu et al. [LWW+13] also focuses on tracking the evolution of documents over time. However, rather than focusing on the change in ideas in social groups or the progression of topics in day-to-day news, the authors focus on visualizing the progression of movie storylines but can easily visualize other types of textual data. Using visualizations, the authors attempted to show how the relationships between entities in the storyline change as the movie progresses with the main goal of optimizing the layout of the visualization to maximize the effect of visual analysis. Furthermore, the proposed visualization has the additional goal of allowing the analyst to tweak nearly every aspect of the visualization in real-time whether it be adding, dragging or bundling entity line relationships. In order to achieve this goal, the authors propose a novel method of mapping user interactions to the layout of the visualization. The layout pipeline to generate the default image can be split into four steps. In the first step, the relationship trees is generated for each time frame. Afterwards, the trees are used to create an order for the lines for each entity. To maximize the readability and aesthetics of the visualization, the lines between each time frame are aligned to make sure that each line is as straight as possible. In the last step, a quadratic optimization algorithm is applied in order to maximize the usage of the available space. The optimization works by minimizing the amount of whitespace, wiggles in the lines as well as the number of times the lines cross with each other. The user interactions are then mapped to each step in the layout pipeline to streamline the layout modifications in real-time as can be seen in Figure 2.2. The authors make the assumption that there are four main cause and effects from interacting with their proposed visualization. By adding and deleting entities, the hierarchy of the tree can change leading to the first step of regenerating

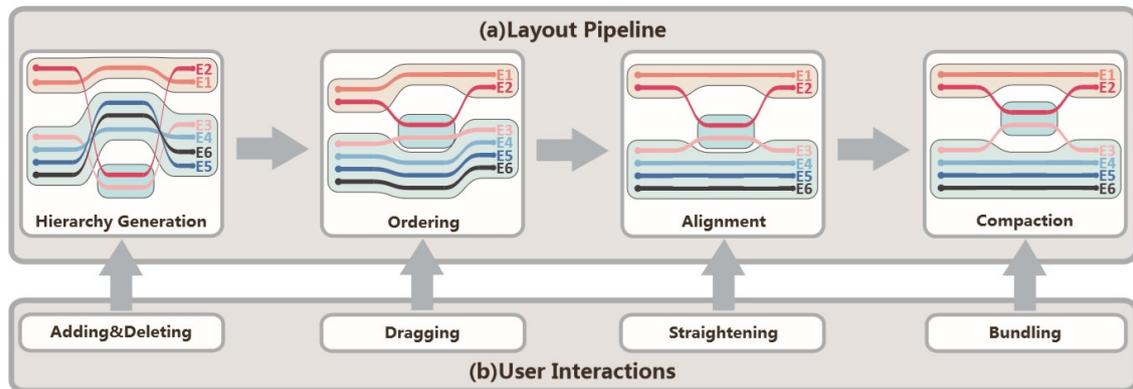


Figure 2.2: A visualization showing what effect each user interaction with the visualization has on the layout of the visualization [LWW+13].

the relationship trees. By moving the existing entities to new positions, the order of the entities are changed leading to the second step being reprocessed. By straightening a line, the entire visualization then needs to be realigned to make sure that the straightened line is consistent. Finally, by bundling multiple lines together, the visualization needs to be re-optimized so that the modified images can to make full use of the available view space.

The paper by Wu et al. [WCS+18] proposes a system called StreamExplorer with the goal of allowing an analyst to visually analyze, track and compare data streams for social media at a macroscopic, mesoscopic and microscopic level that can run on low-performance computers. The proposed system consists of three main components. The first is a online algorithm that can continuously detect important time periods from the social streams [WCS+18]. To do this, the authors proposed a framework with three components: 1) a subevent detector that can detect critical time periods based on the content and frequency of a topic in a certain time period; 2) a preprocessor that cleans the tweets found by the subevent detector; and 3) a map generator which creates topical and geographical maps of the tweets detected by the subevent detector. The second is a Self-Organizing Map method that can quickly analyze the text streams and return relevant and important topics with the assistance of the the power from the graphics card in the computer. The labels for each cluster of topics is then generated used TF-IDF. The last component is an innovative visualization that combines the advantages of glyph-based, timeline and map visualizations. The visualization itself consists of seven components. The tree visualization and line chart provide the analyst with a macroscopic level of information by giving them an overview of the stream. The tree visualization focuses more on giving the analyst an overview of the past events whereas the line chart shows the analyst the current and past tweet activity which is then translated to the amount of data being processed by the system. If a period of time is categorized as critical, normally when the amount of tweets on a topic suddenly spike, a DICON glyph is used to center the attention of the analyst and provide a summary of the tweets in that critical period. As the stream progresses, the oldest data and critical time period events are removed from the line chart and then added to the tree which serves as a history of past events. On a mesoscopic level, StreamExplorer also shows three topic maps which summarize the topics or the geological location of the tweets when clicking on a DICON glyph in one of the visualizations used to display the macroscopic information. By giving the analyst interactive lens which can then be dragged on to any of the visualizations in

StreamExplorer, the analyst is then provided with the details of the area the lens is focusing on such as getting the keywords of tweets in a specific geographical location or the content of tweets about a particular topic.

As with many of the other projects mentioned in this chapter, the paper from Krstajić et al. [KNMK13] focus on the implementation of a system called StoryTracker which allows for the analysis of documents over time, new stories in particular. However, instead of concentrating on the evolution of topics over time like the many of the current literature, Krstajić et al. focuses on exploring the relationships between new stories themselves. Similar to the paper by Liu et al. [LYW+15], the authors show this relationship by allowing analysts to explore how new stories evolve over time, how they relate to one another, as well as how the topics mentioned in the stories merge and split off from other new stories. Furthermore, the system was developed with real-time processing of data streams in mind. To improve performance, the authors worked under the assumption that news articles regarding the same topic discussing the same real-life event normally appear close to one another in terms of time. Therefore, StoryTracker clusters documents in 24 hour intervals and then compares neighboring clusters with each other to discover documents with similar stories. In order to calculate the similarity of stories, StoryTracker breaks down the news story into a set of essential keywords from the title and description of the story. The Jaccard distance is then used to calculate the similarity between different sets of keywords. To visualize this information, the authors utilized a mixture of parallel coordinate plots with stacked bar charts as the vertical axes to visualize the evolution of the story over time and the set of keywords for each cluster of documents in a 24 hour time frame. A zoomed view was also implemented in order to give the analyst a more detailed view of each cluster which showed the titles for the top five documents in the cluster as long as a summary of the cluster itself. Further interaction with the zoom view would result in an article view which presents the raw article to the user as well as metadata such as the most important words in the article.

2.1 Summary

In conclusion, the current state-of-the-art regarding topic tracking over time as well as the various algorithms and visualizations that were used in the respective seven papers were described. The works from Wang et al. [WLC+16], Leskovec et al. [LBK09], Liu et al. [LYW+15], Liu et al. [LWW+13] and Krstajić et al. [KNMK13] were especially helpful in forging a general path on how to determine the relevancy between documents, as well as suggesting possible visualization and interaction methods which could help analysts understand the use the data. Additionally, the paper from Liu et al. [LYW+15], Wu et al. [WCS+18] and Krstajić et al. [KNMK13] helped inspire me in terms of creating effective systems that were intuitive and simple while maximizing the amount of information the analyst would need to complete their tasks through the generous use of interactions. Furthermore, all papers summarized in this chapter were immensely helping for me in learning about the different methods used today to cluster and organize documents. While this thesis does not focus on processing the data in real-time, the knowledge gleaned from these various works provide a foundation which can be built on my current work in the future. However, these projects mainly focus on providing analysts and users the ability to visualize information from large data sets and track the evolution of documents over time. This thesis hopes to provide an alternative way of doing so by using basic visual elements at the expense of requiring more user interaction in order to maximize the understandability of the visualizations and reducing the need for training.

Furthermore, the proposed system also provides a method to allow users to easily detect the origin of a topic in a data set as well as the perceived relationship between the articles. In addition, due to their recent advances and promising evaluations, machine-learning-based embedding algorithms such Doc2Vec [LM14] were assessed for their benefits and eventually used in the proposed system instead of rule-based systems utilized by many of the aforementioned papers and projects.

3 Modeling

This chapter is split into 2 main parts. The first section describes the theoretical advantages and disadvantages of the algorithms used for document and word embedding as well as visualizations used for visual analysis. The second section describes the steps and methods used to implement the specific algorithms and visualizations used in the proposed system as well as the reasoning behind why they were selected above the others.

3.1 Theory

Here the theory behind the various considered word and document embedding algorithms as well as the visualizations for the visual analysis in the proposed system will be described. For each algorithm and visualization, a description of how it works as well as the various advantages and disadvantages of it will be detailed.

3.1.1 Word/Document Embedding Algorithms

Since the invention of the written word, text has been humanity's the primary mean to create and store information. However, due to the massive amount of text that has already written, as well as those that are in the process of being written, it can be difficult and time-consuming for users to sift through all the text to find the data they need as well as make sense of the said information once it is found. One solution to do this is through information visualization and visual analysis. These two methods are used to improve and accelerate the extraction of relevant information from the database in addition to giving assistance to the user when it comes to making sense of the filtered information. However, this does come with several challenges. First of all, text is relatively unstructured in the context of computational pattern matching. Without the use of a rule-based system or machine-learning, it can be difficult to automatically determine the similarity and relationship between the various words and documents. There are also many challenges which attempting to analyze natural language such as automatically understanding ambiguities as well as context sensitive meanings. In order to help solve these problems, word and document embedding algorithms are used.

Now what are word and document embedding algorithms? Word and document embedding algorithms have the main goal of converting the essence of the information in textual data to a machine-understandable format for machine-learning [DOL15]. In this section, four algorithms used for word and document embedding are described: Bag-of-Words; TF-IDF; Word2Vec; and Doc2Vec. Furthermore, the strength and weaknesses of each algorithm will be evaluated.

Bag-of-Words

Bag-of-Words in conjunction with some type of similarity function such as cosine similarity or Euclidean distance is currently one of the most popular methods for word and document embedding, especially in regards to natural language processing. [ZJZ10]. The Bag-of-Word model consists of two main parts: 1) the vocabulary of all known words and 2) a histogram of each known word for each document. The vocabulary itself is simply all unique words from all the documents in the data set. The frequency of each word in each document is then enumerated to create the histogram.

Bag-of-Words itself has many uses in the context of word and document embedding, first of which is feature extraction. Feature extraction can be used as a form of preprocessing for textual data. This is important, especially when attempting to use machine-learning for word and document embedding as machine-learning algorithms often do not work well with inputs of unequal number of features as well as different set of features. By preprocessing the data, the user makes it much easier for the machine-learning algorithm to determine if the input data accurately follows an established pattern. By preprocessing the documents using Bag-of-Words, a simple and flat representation of text documents as the output is received. This means that the rather than text, the individual documents are represented as a series of document vectors which is perfect as inputs for both machine-learning algorithms and rule-based systems. Another advantage of Bag-of-Words is that it is relatively easy to implement as it simply counts the frequency of each word in each document for all documents.

Despite the many advantages, the Bag-of-Words model also has many drawbacks, most important of which is the loss of context or the meaning of word and structure. As the sentence structure, grammar as well as the relationship between the current sentence and past and future sentences are all removed and ignored, much of the information contained in the text as well as the meaning of individual words can be lost. This is particularly important as words can have different meanings and with the loss of semantic information, it can be difficult to automatically determine the relationship between different words. In addition, the Bag-of-Words model determines the importance of the words based on frequency which can be a drawback as words are often not equal with some being more important than other based on focus or context. Another disadvantage is the size of the vocabulary. With a large amount of documents, it stands to reason that the number of words in the vocabulary increases as well. While this can be partially managed through additional preprocessing of the text such as stemming or even the manual removal of unimportant words to create a sparser matrix, the opposite issue can also occur when the resulting Bag-of-Words model is too sparse and contains little relevant information in correlation to the size of the representational space.

TF-IDF

Term Frequency (TF)-Inverse Document Frequency (IDF), can be seen as an extension of Bag-of-Words and is a statistical model that shows how important a word is in a document [Ram+03]. As can be seen by the name, it consists of two main parts: Term Frequency and Inverse Document Frequency. The first part simply calculates how frequently a word appears in the document similar to Bag-of-Words. However, due to differing document lengths, it may be possible for terms to appear more often in documents with longer length. Therefore, the frequency of each word in each document is also divided by the number of words in the document. The second part of the algorithm determines how important each word is in context of the document. In the first part of the

algorithm, all words are seen as equal, including stop words such as “a”, “the” and “and”. This is a problem as these words may appear often in documents but are individually unimportant. To solve this issue, the Inverse Document Frequency calculates the logarithm of the number of documents in the corpus divided by the number of documents containing the specific word. The results of the first and second part of the algorithm are then multiplied together to get the TF-IDF value where the higher then number, the stronger the relationship between the document and the word.

As with Bag-of-Words, one advantage of TF-IDF is the relative simplicity of the algorithm. The computation of TF-IDF does not require large external libraries during implementation and outputs vectors which are easy to read and understand. In addition, despite the relative simplicity of the algorithm, the results of using TF-IDF as part of a query engine regularly returns highly relevant documents.

On the other hand, TF-IDF does come with several drawbacks as well. First of all, as with Bag-of-Words, the semantics and the structure of the documents are not retained. One major drawback of this is that the algorithm cannot determine the relationship between words which may be relevant to one another but rarely written together such as synonyms of words. In addition, if stemming or lemmatization is not used on the documents before being processed by TF-IDF, plurals, gerunds as well as other grammatical variations of a word are all processed individually by the algorithm and seen as separate words.

Word2Vec

Word2Vec by Mikolov et al. [GL14] is a tool for generating word embeddings. Using a text corpus as input, the goal of the software is to output a set of word vectors []. To do so, it creates a vocabulary based on all words in the training text corpus and learns the possible vector representations of each word. Word2Vec currently has two methods to train the text corpus so that it can generate the word vectors: continuous Bag-of-Words and continuous skip-gram. It can be said that these two methods are quite similar as they both attempt to solve the same problem but simply from opposite directions. Whereas the continuous Bag-of-Words method attempts to learn specific words based on the words surrounding those specific words in the training text corpus, the continuous skip-gram model does the opposite and attempts to discover the context of the words from a word [MLS13] as can be seen in Figure 3.1. Therefore, rather mapping specific clusters of words to individual relating words, the skip-gram methods learns specific words and uses this to predict clusters of words.

As a tool for the generation of word embeddings, Word2Vec holds many advantages over its competitors. For instance, unlike Bag-to-Words and TF-IDF, the context of the words are preserved increasing the likelihood of discovering the true importance of each word in a word sequence. Furthermore, by mapping a word to its context word, the relationship between the words are saved in the vector space leading to possible linear translations such as “king”-“man” + “woman” producing a vector similar to the word “queen” [MLS13].

On the other hand, the theory behind Word2Vec is unspecific. To date there is no clear algorithmic reason why Word2Vec produces good vector representations [GL14]. Therefore, it can be difficult to understand why the algorithm proposed the output representations. Furthermore, without retraining the data, Word2Vec does not work with words that are not already in the vocabulary. This means that comparing the similarities of new documents to trained documents is often not possible.

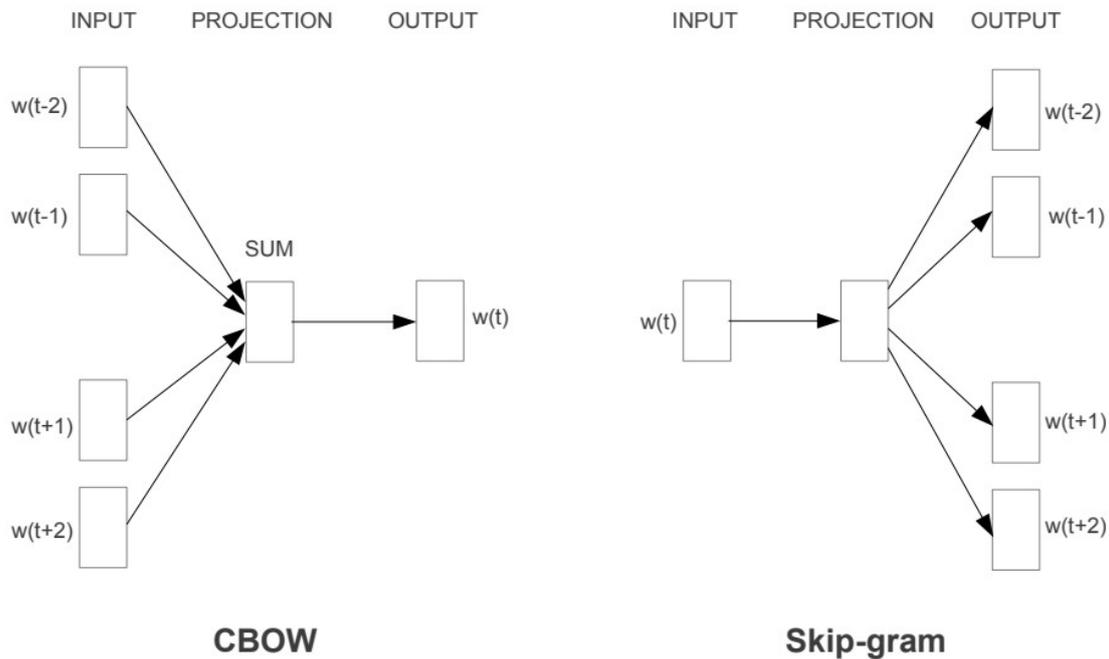


Figure 3.1: Graphical model of the process for both the continuous Bag-of-Words (left) and skip-gram (right) model in Word2Vec. [MLS13]

Doc2Vec

Like Bag-of-Words, Doc2Vec is an algorithm with the main goal of generating document as well as paragraph embeddings [LM14]. This algorithm was proposed in 2014 by Le and Mikolov and can be seen as an extension of Word2Vec by the same authors. In contrast to Word2Vec which focuses determining the relationship between individual words, Doc2Vec determines the relationship between word sequences. While there are many similarities between the two algorithms, many changes had to be made in order to address the additional challenges that appear with document-level embeddings. For example, whereas Word2Vec can rely on the logical structure of words when creating the embedding, Doc2Vec cannot. In order for Doc2Vec maintain the context of the document after document embedding, every document in the data set is mapped to a unique vector which is then saved as a column in Matrix A. Likewise, every word in the data set is then mapped to a unique vector and stored as a column in Matrix B. Using this method, identical words that appear in different documents have the same word vector. By averaging or concatenating the document and word vectors during machine-learning, the algorithm is able to “predict” the next word in context of the document as can be seen in Figure 3.2. In this case, the document vectors represents the possible missing information and serves as a type of memory which allows the algorithm to determine what is missing in context of topic of the document.

With the help of Bag-of-Words, it is also possible to generate document vectors while ignoring the order of the words in Doc2Vec. Rather than concatenating the document vectors with the word vectors as mentioned in the previous method, this method ignores the context of the words. Instead, it uses the document vector to classify words in each document. Afterwards, in the training phase,

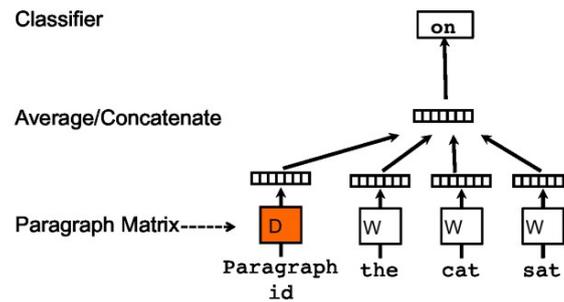


Figure 3.2: Framework for learning the paragraph vector in Doc2Vec. [LM14]

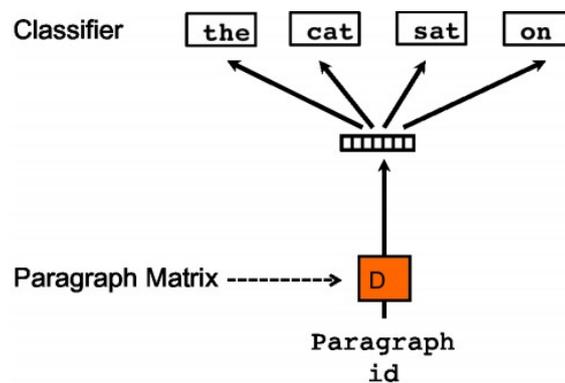


Figure 3.3: Framework for learning the paragraph vector without word ordering using a distributed Bag-of-Words model in Doc2Vec. [LM14]

the algorithm randomly samples word sequences from the document in order to form a classifier and forces a prediction based on the sample in order to check if a specific word belongs to a specific document as can be seen in Figure 3.3.

Doc2Vec holds many advantages over Bag-of-Words. For instance, in contrast to Bag-of-Words, Doc2Vec has the advantage of capturing the semantics of the document as the document vectors work in conjunction with the word vectors. Therefore, based on the data set, the algorithm can determine that the word “powerful” is more related to the word “strong” rather than the word “bread”. Furthermore, Doc2Vec takes word order into account similar. While the word order is not preserved for the entire document unless the document is extremely small, it is similar to the extent of a n -gram model with a large n .

While Doc2Vec holds advantages over algorithms like Bag-of-Words in terms of the ability to retain information, it is rather difficult to access the information itself. Unlike Bag-of-Words in which information contained in the vectors can be easily understood and read in conjunction with the features, with Doc2Vec, the document vectors are difficult to understand and interpret as they contain information from both the document and word vectors.

3.1.2 Topic Modeling

As one of the main goals of the proposed system is to compare the similarity of different documents in order to see the differences and similarities between articles as well as the ability to mark the most important information in the documents, it stands to reason that the algorithm used for topic modeling plays a large role in terms of the success and accessibility of the proposed system. These algorithms are used to automatically and efficiently summarize, organize and understand the information contained in the large library of documents.

For the purpose of the proposed system, three methods of topic modeling were considered: cosine similarity, Euclidean distance and LDA.

Cosine Similarity

Cosine similarity is one of the most popular methods for topic modeling today [RKA12]. Using this method, one needs to first create a set of word vectors using word/document embedding algorithms such as those mentioned in the previous section (Section 3.1.1 - Section 3.1.1). The cosine value between the word or document vectors of two documents then can be calculated in order to find the similarity measure between the two documents where the higher the score, the higher the relationship between the two documents or words. Given that \vec{d} is the document vector of document d and \vec{q} is the document vector of the query document q , the cosine similarity between document d and query document q can be measured as can be seen in Equation 3.1.

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{\|\vec{d}\| \cdot \|\vec{q}\|} \quad (3.1)$$

As it is a widely used algorithm for topic modeling, it obviously has many advantages. One of the largest strengths of cosine similarity is that it corrects for documents of unequal length [Hua08]. For example, with two documents of equal length, one can assume that if document A mentions “cheese” four times and document B mentions “cheese” twice, it can be said that document A is more relevant to the topic of “cheese”. However, it is also understandable that if document A is vastly longer than document B , document B could be more relevant to the topic “cheese” even though the word “cheese” appears more often in document A .

On the other hand, the main drawback of cosine similarity is that it does not reliably process the semantic meaning of the text as it simply strives to match the syntax rather than understanding the context of the words in the word or document vectors [RKA12].

Euclidean Distance

The Euclidean distance simply measures the distance between two word or document vectors within a Euclidean space. However, rather than normalizing the distance or similarity between documents like with cosine similarity, the raw distance between the two documents are calculated leading to a situation where the smaller the value, the higher the relevancy of the two documents [Hua08]. Given that \vec{d} is the document vector of document d and \vec{q} is the document vector of the query

document q and $w_{i,d}$ and $w_{i,q}$ are the i -th word vector in the documents d and q respectively, the Euclidean distance between document d and query document q can be measured as can be seen in Equation 3.2.

$$dist(\vec{d}, \vec{q}) = \sqrt{\sum_{i=1}^n |w_{i,d} - w_{i,q}|^2} \quad (3.2)$$

One advantage of using the Euclidean distance over other similarity and distance measures is that it uses raw data as the input [BJGK14]. Therefore, addition of new data to the data set such as outliers would not affect the overall clustering of the data to a large degree.

On the other hand, the dependency on raw data is also a drawback as scaling plays a large role in determining the accuracy of the distances. If different scales are used for different dimensions, then the accuracy can be inaccurate and misleading. For example, if one dimension is measured in millimeters and another in meters, the Euclidean distance will not adjust to the different scales and assume that they are all the same scale and therefore changing the resulting clusters.

LDA

Latent Dirichlet Allocation (LDA) is a statistical model proposed by Blei et al. in 2002 [BNJ02] which attempts to explain the reason for the similarity between documents. The algorithm assumes that documents are created from a probability distribution of topics which are then further defined as a probability distribution of words. LDA then attempts to determine the topics of a document by reverse engineering the previous assumption [BNJ03]. The core principle of the algorithm can be split into two basic steps. In the first step, given that there are a total of k topics for all documents in the data set which is manually defined by the user, the algorithm assigns a topic to every word in the document. Then in the second step, for each word in the document, it calculates the proportion of words in the document for each k topic as well as the proportion of topics for the specific word across all documents. By repeating these two steps, the LDA algorithm can determine which words in the data set are related to which topic.

There are many advantages of using LDA. One advantage is the ease of using and modifying LDA as a module of a more complex algorithm as LDA itself is generative. Furthermore, LDA allows the classification of multiple dimensions resulting in the simultaneous clustering and ranking of multiple variables. Additionally, LDA has the possibility of merging multiple models together to allow for the clustering and ranking of variables with using the merged vocabularies from each model.

However, despite the advantages due to the modular-ability of the algorithm, LDA on its own is not very accurate compared to other models. This is especially shown in the paper by Le and Mikolov [LM14] where they compared the clustering and similarity performance of multiple similarity and distance measurement functions. In their paper, they showed LDA as having an error rate of 32.58% which is 16.13% more than multinomial naive Bayes which was the second worst algorithm in the comparison.

3.1.3 Visualizations

The main goal of this thesis is to propose a system which allows the user to analyze the information within a large corpus of documents automatically and efficiently. One of the most efficient and descriptive ways of providing information to the user is through the use of interactive visualizations. Visualization in the most basic terms is a method to turn data into information that can be easily understood by a human user. To do this, the data is first structured via data transformation; then focused via actions such as filtering and extraction; and then mapped and rendered onto a suitable visualization [LCWL14]. Using this method, large amounts of normally incomprehensible data can be simplified and shown on a single compact image. Interactive visualization adds an additional layer to visualization by adding UI controls which allow for user interactions to modify the visual perspective of the user as well as allows the user to directly interact with the data itself [YKS+07]. There are many reasons to use visualizations in conjunction with machine-learning. The most obvious benefit is that visualizations can show us what is important, such as a pattern or the outliers. Furthermore, as there is a general overview of the data through visualizations, it is much easier to compare different data sets or parameters see the major differences between them.

Visualizations are used to attempt to fulfill three goals of the system: 1) to show the relationship between articles; 2) to show the relationship between words; and 3) to show a ranked list of words or articles. For the purposes of this thesis, eight different types of visualizations were considered to accomplish the above goals: 1) bubble maps; 2) heat maps; 3) lists; 4) node-link diagrams; 5) parallel coordinate plots; 6) area charts; 7) bar charts; and 8) word clouds.

In the following sections, the considered visualizations as well as the theoretical strengths and weakness of each proposed visualization are described.

Bubble Map

Bubble maps can be seen as an extension of the scatterplot chart [KK11]. However, instead of simple points, bubbles are used in place of dots or shapes to represent the data. Bubble maps visualize data through two main parameters, the location of the bubble based on the x- and y-axes; and the size of the bubble. The different bubbles can then be distinguished from one another based on these two parameters. Using this type of visualization, an additional type of value for each data point can be shown in comparison to the scatterplot charts. Furthermore, by differentiating different data sets by color and texture, data from multiple data series can be shown in a single graph.

Unlike many of the other visualizations described in this section, bubble maps are one of the few visualizations that work well with negative values. This allows for a better representation of the raw values without the need of preprocessing or normalizing the values which can alter the analysts' perception of the data. They also share many of the advantages found in scatterplot diagrams. For example, by how the bubbles are clustered as well as how the position and sizes of the individual bubbles increases or decreases, the analyst can see the growth or decline of values over time as well as determine if there is any correlation between the values. Furthermore, if the data has relatively low correlation, it has the possibility of making full use of the available whitespace.

On the other hand, while bubble maps may have the advantages of scatterplot charts, they also have the disadvantages of them. For example, with a large number of data points, overlapping due to the clutter can be a distinct problem. From an overview point of view, if a lot of points are

overlapping one another and it is very difficult to determine anything other than the outliers and a general correlation. This can be partially solved by using interactions such as zooming. However with zooming, while you can separate the values, the analyst is then unable to view the entire visualization at the same time. Last of all, labeling can also become an issue when a large number of data points as well as for data points with bubbles of small sizes.

Heat Map

The heat map is a powerful visualization used to display and compare data sorted into categories based on color and optionally, size [WF09]. It generally appears in a matrix format in its most basic form consisting of rows and columns where the cells in the matrix are colored based on the value of the specific cell. Furthermore, in order to show data multiple data series, it is possible to fill the cells by color based on the category of the value, then lighten or darken the color or use textures to represent the value in the cell. With cells with overlapping data, the cells can be split into separate segments with each segment representing a different category and value. Additionally, based on the ordering of the rows and columns, a general structure of the data can be shown to the analyst such as a ranking scheme. 3D heat maps are also a valid way to map and compare data sorted into categories. The advantage of 3D heat maps lie in the ability to show at least one more aspect of a data series through the use of height. For instance, regular heat maps are limited to visualizing the quantity of a particular value by color or the shade of a color. With 3D heat maps, this can be more clearly shown with height which allows for better comparisons, especially when there are multiple unique values.

There are many advantages of heat maps. First of all, they are very easy to understand and require just a glance to determine the information on it. For example given that there was a heat map where the higher the value the darker the value, an analyst would be able to register areas with large or small values simply based on the distinct difference in colors. Furthermore, as it generally appears in a matrix format, it reduces visual cluster as well as improves scaling as the available whitespace is efficiently used and by extension, gives a good overview of the visualization to the analyst.

However, when looking for a specific cell or cluster, the analyst will still need to read through all the column and row names in order to find the cell or cluster they are looking for. Furthermore, pattern detection is generally only available if the creator of the visualization ordered the columns and rows in such a way that the patterns can be seen in the first place. However, by this can lead to information bias as the analyst will only be able to easily see what the creator of the visualization wanted them to see.

Lists

Lists, and by extension tables, are very basic ways of showing relational data. They are simply the raw data values organized in such a way that they are human readable by adding column headers.

While simple, lists are also a very effective form of visualization. For example, there is no fear of visualization lies as the analysts can read the values directly as text. This allows the analysts to come to their own conclusion when reading the values in the list. Furthermore, by using a table format, analysts are able to see the relationship between the different properties of an entity.

However, the simplicity itself can also be a drawback. For example, as it just shows the raw values of the data, any additional information based on the comparison of different values have to be done manually by the analysts which can be very time- and resource-consuming when there are millions of lines of rows in the table itself. Furthermore, just reading and remembering the different values in the cells can be difficult when the number of rows and columns can be in the millions or more.

Node-Link Diagram

Currently, trees and namely node link diagrams, are a very popular method of visualization since they are very easy to understand and can organize and structure the data in a very readable and hierarchical way [KEC06]. Each node in the visualization represents an object and is connected by a directed or undirected link which represents the relationship between two objects. Using labels, colors and size, additional degrees of information can be added to the links to show more detailed information about the relationship between the objects.

The tree structure itself is very interactive and many of its limitations listed below can be solved via user interaction. For example, it can be difficult to get a detailed view of a large tree on a single screen or view. However, the use of interactions could solve this problem by allowing the analyst to just focus on the leaves that they are interested in. Furthermore, in an overview of the table, clustering and changing the leaf colors could play a role in helping the users gain an understanding of the data without having to view every node individually. As for the issue where large data sets can cause overlapping and labeling issues in node link diagrams, this can be partially solved with edge bundling methods which has the additional benefit of making the visualization more aesthetically pleasing. However, this does mean that details about individual edges would be lost and it can introduce tracing problems.

However, while it can make full use of the screen by spreading out the nodes, the whitespace is still used inefficiently due to the large amount of unused whitespace around the links and does not show the leaf size of the nodes and children without the use of labels or other visual encoding properties such as size. Another limitation is that it does not handle large sets of data. Due to the clutter, it is difficult to search for a specific node which expands to overlapping and labeling problems. Other than the clutter, due to the top to bottom or left to right hierarchy, it is very difficult to show a large tree on a single screen.

While using a circular layout uses the available whitespace more efficiently than a traditional layout, the hierarchical structure which can be important to the user can be lost [RMC91]. One solution to this is to use a cone tree, which is in essence is a 3D representation of a hybrid circular and traditional node-link diagram. It has all the advantages found in traditional node-link diagrams but uses the space more effectively due to an additional degree of motion. Furthermore, as it is three-dimensional, it is more intuitive for the users to interact with such as the ability to “spin” the cone.

However, with extremely large data sets, it still contains the same problems as the traditional layout such as overlapping and labeling problems and there are still large areas around the links where the space is not used making it less whitespace efficient in comparison to a visualization like the heat map.

Parallel Coordinate Plots

Parallel coordinate plots is a visualization technique used for analyzing multivariate data [KK11]. For this type of visualization, each variable in the data is mapped to a single vertical axis. Afterwards, for each data object, the value of each variable is then linked to their values on their respective axes.

There are many advantages for using parallel coordinate plots over the other visualizations mentioned in this section [Cha06]. First of all, as it is essentially a series of line charts, the outliers and clusters can be easily detected if available. Additionally, it serves as an effective tool to find similarities between data objects as they would have similar lines.

On the other hand, with large amounts of data, cluttering is an immediate issue where it can be difficult to differentiate between the individual lines. This problem is visualized in Figure ???. From the image it can be easily understood that it would be very difficult for an analyst to compare data from specific counties without the use of interaction where filtering, selection and zooming can be used to minimize the problem, but not remove it [Few06].

Area Chart

Like the other charts in this category, the area chart has the main goal of visualizing quantitative data [KK11]. It is based on the line chart visualization and simply fills the area beneath the line with visual elements such as colors or textures. Using such a visualization, an analyst can see the evolution or trend in the data over time. Visualizing multiple data series is also possible using multi-area charts and stacked area charts. With multi-area charts, the user can sort multiple area charts from the largest to smallest in terms of area, then overlap them in that order. With stacked area charts, the different area charts are simply stacked on top of each other instead of interpolated on top of one another. In addition, to make each area chart in the multi- or stacked area chart visually distinct, visual elements such as color, patterns as well as labels can be used.

As mentioned before, one advantage of the area chart is that the analyst can easily see the trend or evolution of the data over time. Additionally, one can easily determine and compare the values of specific data points by simply comparing the height or area of the particular data points.

Despite the simplicity of the idea, area charts also come with several disadvantages. Firstly, due to overlapping, information can be lost when creating multi-area charts. This can happen when there are area charts in the lower layers of the stacked area chart contains values in specific categories which are lower than those found in the area charts in the higher layers. This leads to the data point in the lower layers being overwritten by those in the upper layers which can lead to confusion during the visual analysis of the data. Furthermore, an issue with stacked area charts is that it is relatively difficult to compare the values in different area charts. This can be difficult as the starting point in the upper layers of the stacked area chart are dependent on the height of the area charts in the lower layers. Additionally, due to human perception, this can particularly be a problem as it can be difficult to differentiate the difference in sizes accurately especially if the areas are similar in height.

Bar Chart

The bar chart is currently one of the most popular and basic types of visualization, especially when it comes to mapping out discrete data [KK11]. It is normally visualized as a series of horizontal or vertical bars where the height or length of the bar respectively directly correlates to the perceived value of a data point. Different types of bar charts are available such as multi-bar charts which have multiple bar values for each category of the data and stack bar charts which visualize multiple categories of data in a data point by segmenting the bar into proportions based on the values for each category in the data point. These two types of bar graphs are two ways which enable the user to visualize multiple data series in a single visualization in contrast to the single bar chart which only visualizes a single data series.

Due to the popularity of bar charts, it stands to reason that there are many reasons why it is popular in the first place. First of all, the bar chart is simple to understand and is very suitable for the comparison between different values as the analyst only needs to compare the length or area of the bars to determine the difference between them. Furthermore, as mentioned before with the help of grouping and stacking as seen in the multi-bar chart and stacked bar chart respectively, multiple series of data in a single visualization can be easily visualized.

However, the bar chart also comes with its fair share of disadvantages. First of all, labeling can become an issue with a large number of bars which can make individual bars very thin. Due to the thinness, labels overwriting the boundaries of the bars or overlapping with other labels can be a valid problem. Furthermore, there is the issue of the requirement to use zero as the baseline. This is a problem for data series with generally extremely large values which requires the user to scale size of the bar chart to fit these values. Of course these problems can be solved with normalization as well as value scaling and weighting but this can lead to problems such as visualization lies and the change in perception and interpretation of the information by the analysts [RTB96].

Word Cloud

Word clouds are a relatively modern and aesthetically pleasing method of visualizing textual information [HLL14] in comparison to the aforementioned visualizations. As the name implies, word clouds are essentially clouds of words which consists of words jumbled together in a layout and order specified by the creator of the visualization such as alphabetical order or circular layout with the most frequent words in the middle. They are mainly used to give then analyst an overview of typically the most important or most frequent words. The importance or frequency of a particular word is then visualized through the size of the word.

One main advantage of word clouds is that they are aesthetically pleasing. Furthermore, as with bubble maps, the analysts can quickly determine the importance or frequency of a word simply by the size of the word. By linking together the words by size, the analyst can quickly get a basic summary of the text. In addition, the word cloud itself can function as a way to determine if a text is relevant to the needs of the analyst without the need to analyze the details of the text.

However, the simplicity is also a drawback. The word cloud on its own is simply a statistical summary of individual important words. There is no way to determine how the words are linked together as the context of the words in the text as well as the structure of the sentences are not

preserved. On the other hand, this only holds true for the fundamental variants of word clouds and there are more advanced variants in terms of layouts which minimizes this problem to a certain degree [BKP14].

3.2 Design Approach

In order to determine which algorithms and visualizations were the best fit in order to achieve the goals of the proposed system, they were compared to one another. To do so, the advantages and disadvantages of each algorithm and visualization mentioned in the previous section as well as the results of studies done by previous projects was taken into account.

3.2.1 Word/Document Embedding Algorithms with Topic Modeling

In this section, the different considered word and document embedding algorithms in combination with the different topic modeling algorithms, both which are described in the previous chapter, are compared.

Word Embedding Algorithms

First is the evaluation of word- and document embedding algorithms. For the purposes of the system, the accuracy performance would play the largest role in determining which algorithm to use. However, it can prove difficult to solely evaluate the accuracy of a word or document embedding algorithm from the output alone. Therefore, the accuracy evaluation will be made by combining the outputs from the word or document embedding algorithms and using said outputs as the input for the topic modelling algorithms in order to establish the combination of algorithms with the best accuracy for the test case. For word embedding algorithms, Bag-of-Words, TF-IDF and Word2Vec were evaluated in conjunction with cosine similarity and Euclidean distance as well as LDA on its own. In terms of accuracy, Bag-of-Words has an advantage over LDA in terms of accuracy. This can be seen in the paper by Zhao et al. [ZZM16]. In their paper, the authors compared different variations of the Bag-of-Words model with LDA and Latent Semantic Analysis (LSA). They then used these models to cluster 1,762 tweets from Twitter¹ and determine instances where the tweet contained traces of cyber-bullying. From their experiment, they determined that a Bag-of-Words model using cosine similarity had a F1 score of 76.6 compared to the lower score for LDA with 74.9.

Like with Bag-of-Words, it can also be said that Word2Vec has an advantage over LDA. In the paper by Shen and Rudzicz [SR17], the authors attempted to predict signs of anxiety disorders through social media posts in Reddit² and Twitter. To do so, the authors compared the effects of word embedding in Word2Vec, document embedding in Doc2Vec and topic modeling in LDA using a linear kernel support vector machine or neural network machine-learning algorithm as the classifier. They then did an experiment using the aforementioned algorithms together with a pre-classified

¹<https://twitter.com/>, last visited July 11, 2019.

²<https://www.reddit.com/>, last visited July 12, 2019

Reddit and Twitter data set containing 22,808 and 100,000 posts respectively. From the results of the experiment, the authors discovered that Word2Vec had an advantage over LDA in terms of accuracy for both data sets and classification models. On one hand, Word2Vec using the linear kernel support vector machine classification model had a mean accuracy of 85.95% (SD = 4.65%) and 84.3% (SD = 5.7%) for the neural network model. On the other hand, LDA had a significantly lower average accuracy of 80.8% (SD = 6%) and 78.35% (SD = 6.25%) across both data sets for the support vector machine and neural network models respectively.

There are also works that compare the effectiveness of Word2Vec and TF-IDF. One such work is the paper by Lilleberg et al. [LZZ15]. In their paper, the authors evaluated five variations of Word2Vec with LinearSVC for the classification and TF-IDF using cosine similarity on a data set consisting of 18,000 news articles on twenty topics. From the results of the evaluation, the authors noted that Word2Vec was significantly less accurate than TF-IDF when classifying data sets without the removal of stopwords with accuracy of 84.19% compared to the TF-IDF accuracy of 89.46%. However, the results are inverted after the removal of stopwords and the change from Bag-of-Words weighting to TF-IDF weighting with Word2Vec achieving an maximum accuracy of 89.59% and TF-IDF attaining a slightly lower maximum accuracy of 88.11%. Last of all, the authors attempted to combine the vector representations of Word2Vec using TF-IDF weighting and TF-IDF after the removal of stopwords. Using this method, they were able to achieve a maximum accuracy of up to 89.73%. However, the authors noted that in both instances of Word2Vec which achieved a higher maximum accuracy than TF-IDF without stopwords, on average TF-IDF without stopwords outscored the two Word2Vec variants in terms of accuracy.

In terms of cosine similarity versus Euclidean distance, cosine similarity was the most popular of all reviewed literature as there were none that used Euclidean distance as the similarity measure to determine document or word similarity. While Euclidean distance does have its own advantages and disadvantages, papers such as the paper from Huang [Hua08] show that Euclidean distance is relatively inaccurate in comparison to other topic modeling algorithms such as cosine similarity or the Jaccard coefficient. For the experiment done by Huang [Hua08], the author clustered data from seven data sets using multiple similarity and distance algorithms in conjunction with the K-means algorithm. From the experiment, Huang determined that the Euclidean distance only clustered on average over all data sets 41.86% of the datapoint accurately. In contrast, cosine similarity managed to cluster the data with an accuracy of 66.86%.

Document Embedding Algorithms

As for document embeddings, the advantages and disadvantages of the Bag-of-Words, TF-IDF and Doc2Vec algorithms were assessed. As above, the document embeddings in conjunction with the various topic modeling algorithms to determine their accuracy were evaluated. In the paper by Maas et al. [MDP+11], they do a comparison of multiple document embedding algorithms for the purposes of sentiment analysis. Of the algorithms they evaluated in their study, two of them were the Bag-of-Words algorithm using the cosine similarity for topic modeling and LDA. For their paper, they implemented the various algorithms and had them classify three data sets: a collection of 2,000 movie reviews; a data set containing 50,000 reviews on IMDB³; and lastly a sentence

³<https://www.imdb.com/>, last visited on July 6, 2019

subjectivity data set by Pang and Lee [PL04]. Using these data sets, they evaluated two versions of Bag-of-Words, one with a smoothed delta IDF during preprocessing and one with no IDF. From their experiment, Bag-of-Words without IDF had an average classification accuracy of 87% whereas the version using IDF had an average accuracy of 86.56% across all three data sets with the accuracy actually increasing with the number of documents in the data set. On the other hand, LDA did not perform as well with an average classification accuracy of just 66.92%. With the paper from den Brave et al. [BVF18], the authors analyzed and compared the differences between Bag-of-Words and TF-IDF algorithms in conjunction with cosine similarity. In their paper, they used Bag-of-Words and TF-IDF to classify three sets of data: 1) consists of questions, answers, comments and tags in regards to “gaming” from StackExchange⁴ containing 5769 documents; 2) consists of questions, answers, comments and tags in regards to the English language in StackExchange containing 7,704 documents; and 3) a partial Reuters-21578 data set containing 5,680 documents. It should be noted that while the data sets from StackExchange were human-readable, the Reuters-21578 data set was less so containing the use of abbreviations and specific news jargon. Furthermore, for all three data sets, exactly half the data set was used for training whereas the other half was used for testing. For the evaluation, the authors implemented two versions of Bag-of-Words, one which outputs the frequency of each word in the vocabulary as the vector and one which outputs whether or not a word in the vocabulary exists in the document. Furthermore, two versions of TF-IDF were also implemented, one from scikit-learn⁵ due to its popularity and a log based TF-IDF weighting scheme from [KPA08]. The results of the study by den Brave et al. [BVF18] were quite surprising as they determined that for the data sets they used, a standard Bag-of-Words algorithm using word frequency with an average accuracy of 83.67% was the most accurate of the four evaluated algorithms with the boolean Bag-of-Words algorithm appearing in second place with 81.33% and both TF-IDF algorithms appearing in third place with an accuracy of 79.33%. These results were surprising as they go against current literature as TF-IDF was developed as an extension of Bag-of-Words by the proposed improvement of IDF as it solves the TF problem where all words are treated equally [MRS10]. On the other hand, studies done by various authors such as Chen et al. [CWS+13] show TF-IDF with cosine similarity vastly out-performing Bag-of-Words with cosine similarity in terms of accuracy, especially when it comes to large data sets with an increased accuracy of up to 7% when trained and tested with data sets containing news and website documents respectively. Additionally, this is also shown in the paper from Le and Mikolov [LM14] show similar results with TF-IDF having a lower rate of error at 11.77% in comparison to the 12.20% they found with Bag-of-Words using the data set containing IMDB reviews.

In the paper from Chen et al. [CWS+13], it showed that for news article data sets, the LDA algorithm follows the same trend as the Bag-of-Words algorithm with cosine similarity with very similar performance but is still out-performed by TF-IDF in combination with cosine similarity. Furthermore, the paper from Le and Mikolov [LM14] exacerbate these results as their experiment showed that LDA had a significantly higher error rate of 32.58% in comparison with the other algorithms in the study such as Bag-of-Words and TF-IDF with cosine similarity with error rates of 12.20% and 11.77% respectively. On the other hand, the same paper [LM14] shows that using Doc2Vec is a significant improvement over Bag-of-Words, TF-IDF and LDA. With an error rate of just 7.42%, Doc2Vec shows a 4.78% improvement over Bag-of-Words, 4.35% over TF-IDF and an

⁴<https://stackexchange.com/>, last visited July 6, 2019

⁵https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, last visited July 6, 2019

enormous 25.16% improvement over LDA in terms of classification errors. This is also shown to a lesser extent in the paper from Dai et al. [DOL15] where the authors determined that Doc2Vec had an accuracy of 78.8% compared to the 67.7% for LDA and 78.3% for Bag-of-Words with cosine similarity. On the other hand, the paper from Shen and Rudzicz [SR17] show results to the contrary. In their study described in the previous section, the authors were able to achieve a higher average accuracy for LDA over Doc2Vec with an average accuracy of 80.8% (SD = 6%) over the Doc2Vec accuracy of 78.75% (SD = 1.55%) when using a support vector machine classifier. However, when using a neural network classification model, Doc2Vec had a higher accuracy with 81% (SD = 1.3%) in comparison to the LDA accuracy of 78.35% (SD = 6.25%). Furthermore, it should be noted that for both classification models, Doc2Vec scored higher than LDA when categorizing the Twitter data set with a 5.5% and 10.2% improvement in accuracy in comparison to the LDA results with the same data set.

Due to the contrasting opinions found in the literature research, the three promising document embedding methods: Bag-of-Words; TF-IDF; and Doc2Vec were tested. Due to the inherent advantages of cosine similarity over Euclidean distance, especially considering the variety of the articles in the data set, all three document embedding methods were clustered using cosine similarity. For the purposes of the system, given that the similarity values range from 0 to 1, documents with similarity values under 0.3 were ignored as they never provided documents relevant to the selected articles in the tests. As Bag-of-Words and TF-IDF simply transforms the matrix of words into a Bag-of-Words and TF-IDF matrix respectively, machine-learning is unnecessary. Therefore, the modeling was done relatively quickly at merely 53.4 and 58.46 seconds respectively. In contrast, it took over 12 hours to train all 63,450 articles using the discrete Bag-of-Words method [LM14]. However, it should be noted that the Doc2Vec parameters were tuned for accuracy over performance with a relatively high vector size of 400, the maximum distance between the current and the predicted word was set to a value of 15^6 , the minimum frequency of the word at 5, negative sampling of 5 for noise removal and a training epoch limited to 100. Furthermore, it should be noted that Doc2Vec was limited to a single process which would have influenced the training times. Furthermore, the entire data set for both training and testing was used as over-fitting would not be an issue since the data was expected to be fully retrained every time new documents were added to the data set. This was mainly to avoid constant use of the “infer_vector” function from Doc2Vec which is used to transform unknown documents into a document vector. However, due to the random nature of the function, it regularly produced different vectors for the same document. Furthermore, due to the wide variation in topics and the large difference in articles relating for topics, over-fitting the model would increase the accuracy substantially [SK96]. As for the accuracy of the various methods, all the articles related to the topics from four specific articles were reviewed. Due to large amount of articles and time constraints, only the titles of the articles were used to determine the degree of relevancy of each article. Furthermore, the results should be seen as established as the results were classified manually and based on the opinion of a single person with a limited test set. Rather, they are meant to be a representation of the general differences between the three algorithms. In the test, all three methods correctly found all the highly related articles. However, there was a large difference in terms of false positives. Of the three methods, Doc2Vec performed the best. Out of the 879 articles with a similarity value of between 30 and 100 across related to the four articles, the tester could not find any instance of false positives. With TF-IDF, 921 articles were found. However, from the titles of the 72 extra articles, they were classified as false positives. For example, with

⁶<https://radimrehurek.com/gensim/models/doc2vec.html>, last visited July 16, 2019

with an article discussing “Game of Thrones”, there were 19 articles regarding video games which had nothing to do with the Game of Thrones television series. Of the three, pure Bag-of-Words had the worst results with a total of 1,431 relevant articles found for the four articles. Additionally, the cosine similarity ratings largely did not match those from the Doc2Vec and TF-IDF methods. More than half the expected documents were given higher similarity values than those from the other two methods.

3.2.2 Visualizations

It can be said that visualizations are the most important component of the proposed system. Through the visualizations, the user needs to be able to intuitively, accurately and efficiently gather the relevant information. Therefore, the visualizations mentioned in the previous section are then evaluated based on their expected role and use-case in the system. From the goals of the proposed system, three main roles for the visualizations were extracted: 1) show the relationship between articles; 2) show the relationship between words; and 3) show articles or words based on their relevance or importance. Additionally, the visualizations will be evaluated based on the criteria set by Mackinlay [Mac86]. The criteria is as follows: [Mac86]

Expressiveness

- The design encodes all the facts in the set
- The design encodes only the facts in the set

Effectiveness

- The design can be interpreted accurately
- The design can be interpreted quickly
- The design has visual impact
- The design can be rendered in a cost-effective manner

Relationship between articles

The relationship between articles mainly deals with comparing a selected article with other related article. To display the relationship between different articles, the following information that needed to be conveyed to the analyst was presumed: 1) the similarity between the selected article and articles related to it; 2) the change in the topic of articles related to the selected article over time; and 3) the details of specific articles such as the content or publishing date. From these requirements, the four main variables that needed be shown were extracted, 1) the similarity value of the article in comparison to the selected article, 2) the date of the article; 3) the topics contained in the article and 4) the difference in topics between an article and the ones that came before it.

To show these variables, following visualizations were considered: bubble maps, node-link diagrams, parallel coordinate plots, area charts and bar charts. These were chosen as they were the most suitable based on the literature research from the previous section. Two types of bubble map visualizations were considered when attempting to visualize the relationship between articles. The

first was the standard bubble map visualization which can be equated to a scatter plot diagram using bubbles of various sizes instead of points. However without the addition of interactions such as zoom or dragging, the bubble map cannot pass the expressiveness criteria requiring the design to encode all facts in the data. The main reason it did not pass the expressiveness criteria was due to the issue of overlapping. Based on the size differences between the bubbles, it stands to reason that multiple bubbles may overlap or completely cover up other bubbles given enough data. The second considered bubble map visualization was a bubble map visualization in a radial form. However, while aesthetically pleasing, one drawback of this visualization is that the expressiveness of the visualization is reduced as again due to overlapping. However, this could be partially solved by making all bubbles equal size and taking away the temporal aspect of the visualization. Using this visualization, the center bubble would contain the selected article and the bubbles surround the central bubble are individual articles where the distance to the central bubble correlates to the similarity of the articles. As the bubbles are all of equal size, the overlapping issue can be reduced given enough spacing between the individual bubbles. However, this type of visualization comes with the cost of reduced effectiveness and analysts may have difficulties determining the design accurately and quickly if accessing articles with a low amount of related articles. Due to the sparsity of bubbles, it can be difficult to accurately and quickly determine the similarity between articles, especially if there is little to no articles that are highly similar to the selected articles due to the large amount of whitespace between the central bubble and the next closest surrounding bubble. Furthermore, this is added on top of the natural inaccuracy of area-based visualizations [Car99]. Of course the bubble distances could also be scaled where the most similar article in the set of related articles always appears next to the central bubble, but this would lead to visualization lies.

Node-link visualizations are also a good way of showing the relationship between articles. To reduce the amount of visual clutter, nodes could consist of groups of articles with similar similarities or visual elements such as size could be used with edges to visualize information such as the similarity value between two articles. Using these types of visualizations share many of the same advantages of bubble maps and also directly visualizes the relationship between the articles without needing the analyst to infer the information based on distance or differences in color. However, node-link diagrams can also share the same disadvantages as bubble map visualizations and has the additional problem of dealing with the visual clutter from the edges. There are many edge layouts which can be used to solve this problem such as bundling, but this comes with the problem of decreased visual information due to overlapping edges and the difficulty in differentiating between different edges. In addition, in all the proposed node-link visualizations, the temporal aspect is not displayed which is one of the requirements for the visualization. The visualization could naturally also be placed on a pair of horizontal and vertical axes, in essence turning the node-link diagram into a bubble map with edges between them. However, while this will allow the node-link diagram to visualize the temporal aspect of the data, it then also comes with all the drawbacks of the bubble map visualization such as overlapping.

Parallel coordinate plots, on the surface, are a very good method for visualizing the four variables required for the task. Using the horizontal axis, the visualization can show temporal data. With the vertical axes, the visualization can show the other variables required by the visualization such as similarity value and the top n topics. By comparing the top n topics of multiple articles, analysts can then see the differences between the topics. However, parallel coordinate plots has a fatal flaw, in particular when there can be thousands of articles that need to be visualized in one parallel coordinate plot visualization. The flaw is the amount of visual clutter. With thousands of intersecting and overlapping lines, it would make it near impossible for the analyst to find specific information

and therefore reducing the expressiveness and effectiveness of the visualization. As mentioned previously, these problems can be minimized through the use of suitable interaction design patterns such as hovering and filtering. However, while minimized, the problem still exists and the addition of hovering and filtering thousands of lines introduces new problems such as determining which lines to hover or what types of filters to set.

In many ways area charts, and in particular stacked area charts are an effective method for visualizing the required variables. With the axes, they can visualize the temporal aspect. Using the area charts in the stacked area chart, it can visualize the similarity of groups of articles. Furthermore, with the effective use of labels the topics of each group of similarity over time can be shown. Furthermore, it can be designed in such a way that it passes most of the expressiveness and effectiveness criteria set by Mackinlay [Mac86]. However, the effectiveness of stacked area charts can also be seen is inadequate for detailed visual analysis. Due to the fundamental nature of stacked area charts, area-based visualizations are not optimal for quantitative perceptual tasks when accuracy where accuracy is important [Car99]. Furthermore, it can be difficult to differentiate and measure the different heights of different area charts as each area chart will most likely not have a singular smooth start point.

Last of all is bar charts. Due to the amount of variables that need to be visualized, only stacked bar charts were considered. For this visualization, the articles were grouped by date and similarity. Each date in the visualization then consists of stacked bars where each bar consists of the group of articles with a similarity value between X and Y where $X = 30, 40, 50, 60, 70, 80, 90$ and $Y = 39, 49, 59, 69, 79, 89, 100$. Additionally, the height of each bar represents the number of articles in each group. In terms of visualizing the required variables, the stacked bar chart is the most promising of all visualizations mentioned so far. It has all the advantages of stacked area charts, but the use of position and length instead of area is major advantage in terms of accuracy for the analyst [Car99]. Additionally, as each category is visualized as an individual bar instead of a single seamless visualization found in stacked area charts, the effectiveness property set by Mackinlay [Mac86] is not violated. Furthermore, through the use of labels which visualize the most important word in the article granted that it has not appeared before in previous articles, analysts are able to quickly and easily see which topics have been introduced in later articles and which articles are in essence summaries of previous works. An example of this visualization can be seen in Figure 3.4. Furthermore, by adding simple interactions such as the ability to click on or hover over the bars, the analyst can see details about the bars such as the articles, the most important words for the articles and the most important words for the articles which have not appeared before. In addition, by allowing the analysts to filter the articles by year and month through the use of drop-down menus, the amount of visual clutter seen by the analyst was reduced. To further reduce the amount of visual clutter so that the analyst only needs to see the data set they are interested in, by clicking on the legend which consists of the different clusters of articles by their similarity value, the analyst is able to hide the bars containing the articles with the clicked similarity as well as the bars containing articles with lower similarity.

Relationship between words

To determine what types of visualization would be effective to show the relationship between words, it is important to determine what variables needs to be displayed. One obvious variable that needs to be displayed is whether or not there is a relationship between two words. Another important

3 Modeling

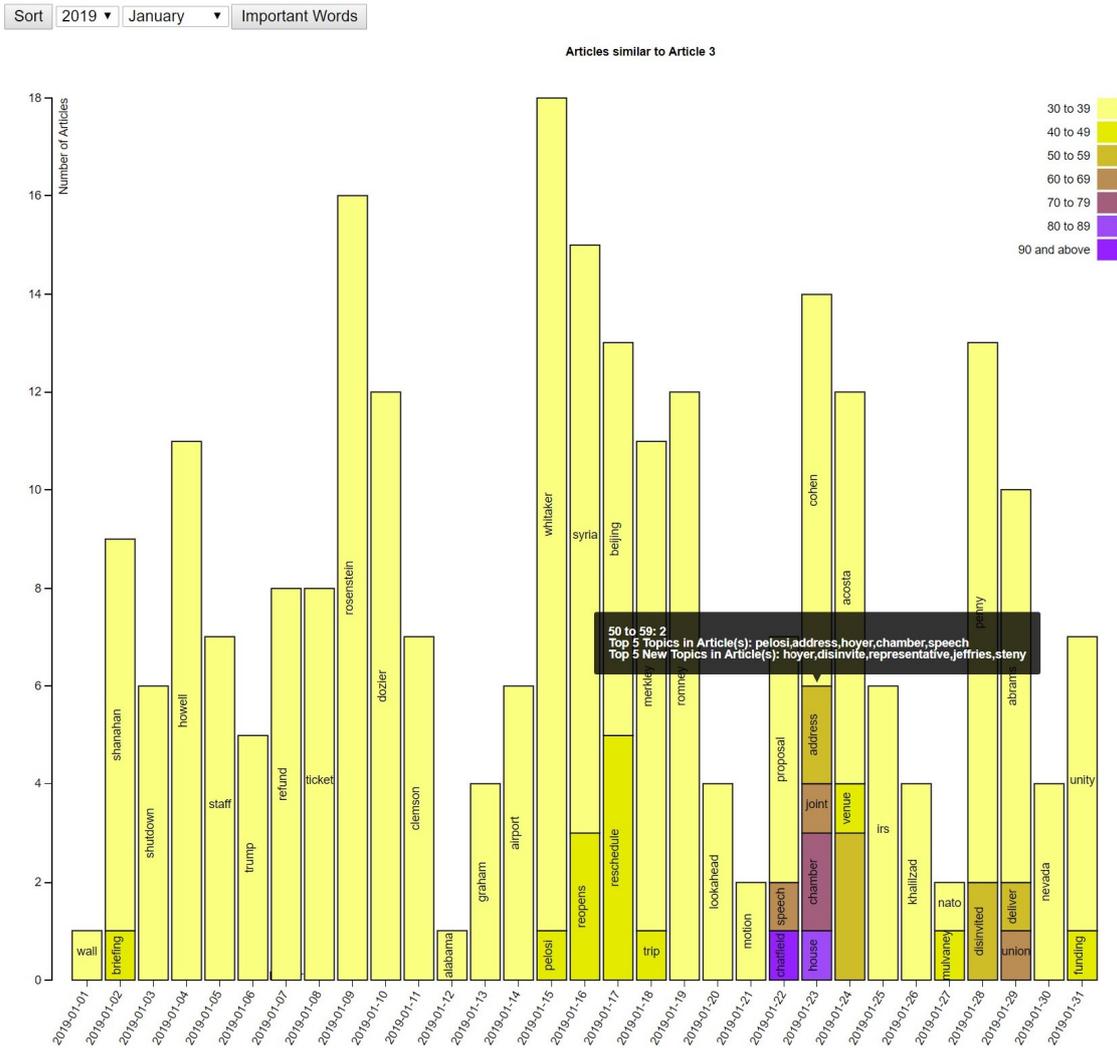


Figure 3.4: Example of stacked bar chart visualization used in the proposed system. Here, color is used to enhance the analyst’s ability to differentiate between different categories. The color ranges from light yellow to dark purple where the darker the purple, the higher the similarity. On hover the similarity category, amount of articles in the bar and top 5 topics in the bar as well as top 5 new topics that did not appear in previous bars appear.

variable that needs to be shown is the frequency of the words over time. This is important as the analyst can then determine what keywords are most popular when writing about a certain topic or how related keywords have changed as the topic evolves. To this effect, the node-link and heat map visualizations were the most suitable visualizations to display this information.

As mentioned before, one important variable that needs to be shown is if there is a relationship between words. A node-link diagram is a convenient way to visualize this information due to the use of edges. Words linked together have a relationship of some type. By linking together strings of words with directed edges, an analyst can determine a set of subtopics and related keywords for each word. For example, with a node-link branch consisting of *President* → *Trump* → *Mexico*, an analyst can observe that when talking about “Presidents”, “Trump” is an important sub-branch of the topic “President” and “Mexico” is an important word related to “President” and “Trump”. However, visual clutter due to intersecting and overlapping lines are a major drawback of node-link diagrams. To solve this, a ring-network diagram seen in Figure 3.5 was used. By using a ring layout, the visualization can more clearly show a hierarchical layout which improves the effectiveness of the visualization as the analyst can quickly determine that the central node is the main or queried topic, the nodes in the inner ring are the most related sub-topics to the central node and the nodes in the outer ring consists of the most important keywords for central and inner ring node combinations. Additionally, to minimized the amount of visual clutter and reduce the cognitive load on the analyst, only the top ten most related words to the central node and each of the central node and inner ring combinations are shown. To further reduce visual clutter, visual elements such as highlighting edges when hovering over nodes to show which nodes the focused node is related to. Last of all, to improve the expressiveness of the visualization, analysts can click on individual nodes and show that node as the new central node in context of the original queried node.

While using node-link visualizations are an expressive and effective way to visualizing the direct relationship between words, it is missing the temporal aspect and therefore cannot show the frequency of the words over time. However, heat maps are a good way of visualizing this type of information. The temporal aspect for each word can be shown in the axes of the visualization and the frequency of the word can be easily shown using shades of color. However, using shades of color by itself is an inaccurate way to quantifying data [Car99]. Therefore, hovering was included as a form of interaction where hovering over a cell reveals the date and word of the cell as well as the frequency of the word for that particular date.

Ranked Lists

The last role of visualization for the proposed system is where the visualization needs to show articles or words based on their relevance or importance. For this type of role, a univariate visualization is enough as it only needs to show the rank of a word or article. Due to the relative simplicity of the role, two visualizations were chosen: a combination of bubble map and word cloud to visualize the ranking of words; and a list in a table format to visualize the ranking of articles.

The bubble map visualization was chosen for a variety of reasons. First of all, in general, words in the English language are logically shorter than titles. This helps deter against the issue of labels going out of the boundaries of the bubble when it comes to large labels with small bubbles. Furthermore, by limiting the number of words in the ranking, the amount of visual clutter is reduced and there is no risk of overlapping as the size of the bubble and area between the bubble can fixed

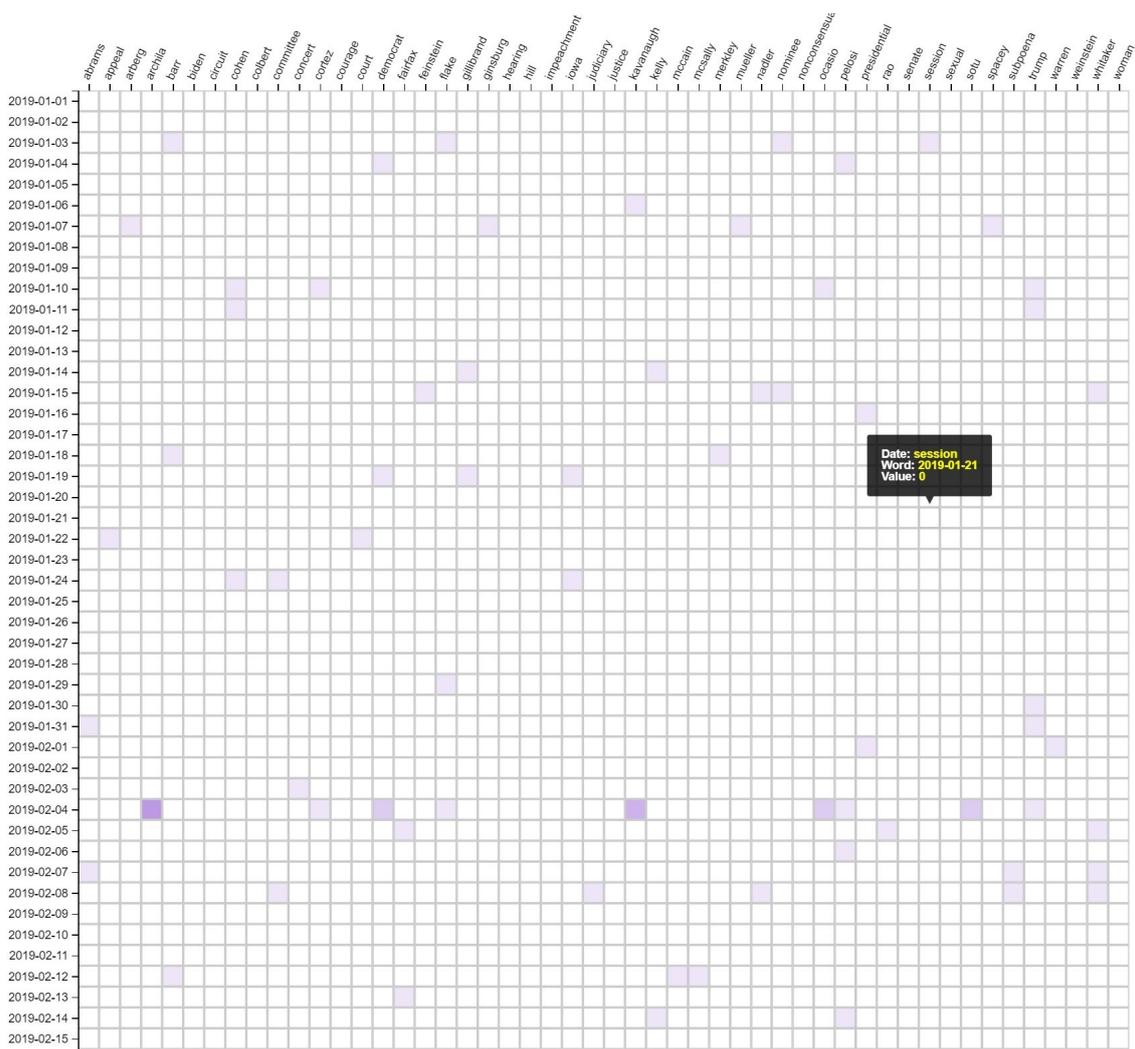


Figure 3.6: Example of heat map visualization used in the proposed system. Here, the colors range from white to dark purple where the darker the purple, the higher the frequency of the word.

libraries such as Whoosh!⁷ allows the system to be able to rank the articles in such a way that it does not require the analyst to do a large amount of exploring before finding interesting articles based on their query.

3.3 Design Implementation

In this section, the architecture of the proposed system will be explained. In this section, the data, preprocessing of the data and the general workings of the back- and front-end of the system will be described.

⁷<https://whoosh.readthedocs.io/>, last visited July 13, 2019.

Top 50 Most Important Keywords in Database



Figure 3.7: Example of bubble map visualization used in the proposed system which shows the top 50 most important words in the data set. In this visualization, the larger the bubble, the more important the word. The ranking is done using a TF-IDF and cosine similarity algorithm.

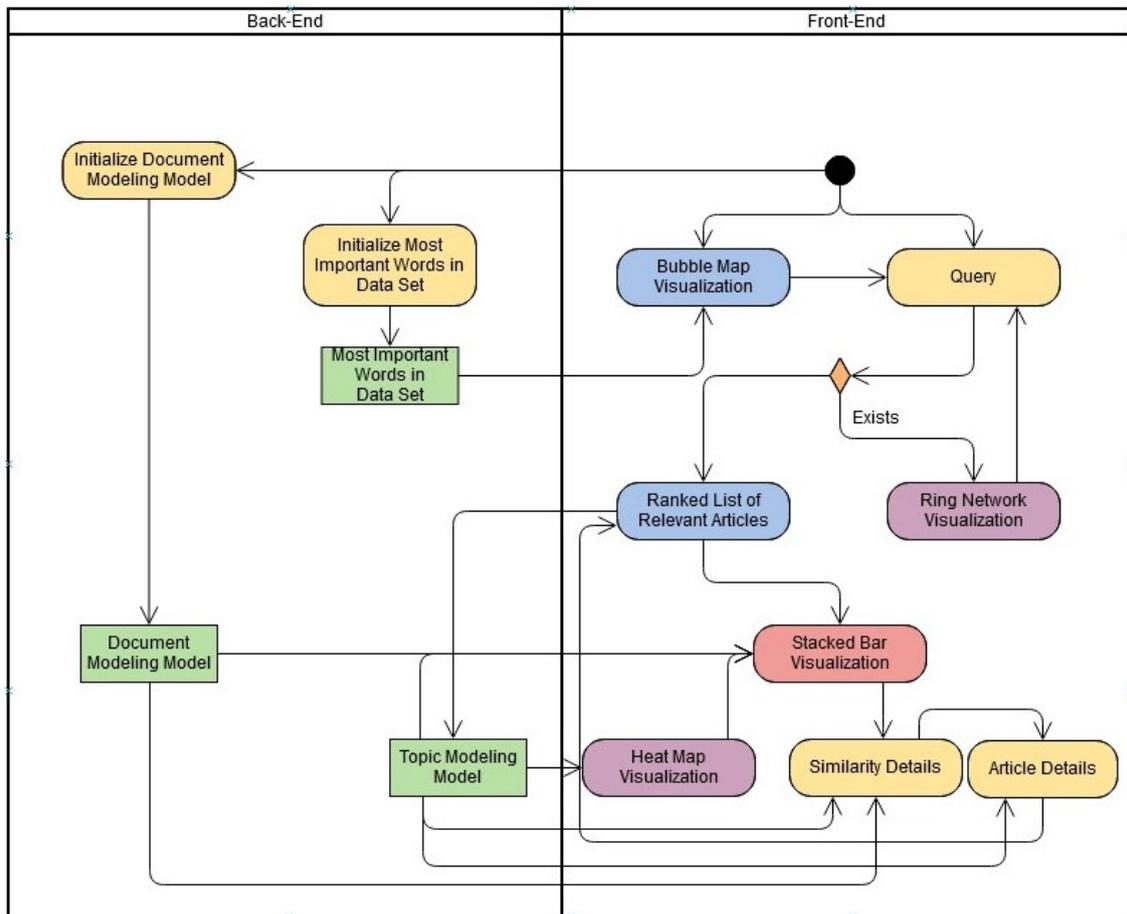


Figure 3.8: UML activity diagram for the proposed system which shows the relationship between each component. Blue activities represent ranked list visualizations, purple represents visualizations that show the relationship between words and red shows the relationship between articles.

3.3.1 Data set

The data set used for the study was provided by the University of Stuttgart⁸ and consisted of 63,450 news articles pulled from various digital news outlets such as The Guardian⁹ and AP News¹⁰ between January 1, 2019 and April 31, 2019. While the majority of the articles are affiliated with politics in some way due to the politically charged climate of that time period, news regarding areas outside politics such as entertainment, disasters and sports were also included in the data set. In the data set itself, the language, title, link, description, publishing date, time when the article was fetched, content of each article and the number of times it was reprinted was collected.

⁸<https://www.uni-stuttgart.de/>, last visited July 30, 2019.

⁹<https://www.theguardian.com/>, last visited July 15, 2019

¹⁰<https://www.apnews.com/>, last visited July 15, 2019

Pre-processing

Of the original data set, only the title, link, publishing date and content of the article were kept as only English articles were collected and the article fetch time was irrelevant in comparison to the publishing date of the article. The article content was then pre-processed in preparation for feature selection and machine-learning. Overall, seven types of pre-processing was used on the content of each article: 1) convert all words to lower case; 2) remove web links; 3) remove Arabic numerals; 4) remove punctuation; 5) remove leading and ending white-spaces; 6) remove stop-words with help from the Natural Language Toolkit¹¹ stop-words corpora; and 7) perform lemmatization on all words using the WordNet Lemmatizer from the Natural Language Toolkit. All pre-processing was done using Project Jupyter¹² using Python 3.7¹³

3.3.2 Back-end

This section will describe the back-end of the proposed system which is split into topic and document modeling components. The back-end processing used PyCharm 2018.3.5 Professional Edition¹⁴ with Flask¹⁵ using Python 3.7.

Topic Modeling

From the literature review, it is generally agreed on that TF-IDF with cosine similarity is generally superior LDA and to a lesser extent, superior or equal to Bag-of-Words and Word2Vec. Therefore, because of the results from the literature review and the ease of implementation due to the relative popularity of the TF-IDF library from scikit-learn¹⁶, the TF-IDF algorithm with cosine similarity was used when it comes to determining the similarity and ranking of independent words in individual documents. Furthermore, due to memory and performance constraints, determining the top 50 most important words across all articles in the pre-processed data set was done separately from the system itself as can be seen in Figure 3.8. However, establishing the top five most important words in related articles with a certain similarity value range or in a specific article are done in real-time by the system itself. It should also be noted that due to the memory and performance issues of the TF-IDF library with large feature and data sets, Word2Vec using the continuous Bag-of-Words algorithm was used to determine all words related to the words in the query shown in the ring network visualization. While the TF-IDF algorithm is considered more accurate than the Word2Vec model, the Word2Vec model from gensim¹⁷ has vastly superior memory management and performance because of the extensive use of sharding which is important for the system as the

¹¹<https://www.nltk.org/>, last visited July 25, 2019.

¹²<https://jupyter.org/>, last visited July 27, 2019.

¹³<https://www.python.org/downloads/release/python-370/>, last visited July 27, 2019.

¹⁴<https://www.jetbrains.com/pycharm/>, last visited July 27, 2019.

¹⁵<https://palletsprojects.com/p/flask/>, last visited July 27, 2019.

¹⁶https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, last visited July 12, 2019

¹⁷<https://radimrehurek.com/gensim/models/word2vec.html>, last visited July 27, 2019.

related words needed to be found in real-time. Another benefit of using Word2Vec in this instance is that it also automatically trained in parallel with the Doc2Vec and shares same model used for Doc2Vec. This reduces the need for any additional training or vector sets.

Document Modeling

From the literature review and the results of the test between Bag-of-Words, TF-IDF and Doc2Vec described in the previous section, it was discovered that using Doc2Vec in conjunction with cosine similarity for topic modeling would be the most advantageous to the goals of the thesis. From the analysis, Doc2Vec was determined to have the highest accuracy with the lowest number of false positives which allowed for the visualization of articles with lower similarity scores such as those with a cosine similarity score of between 0.3 and 0.39 without increasing the visual clutter by a large degree. Furthermore, while it can be seen from the training times that with a real-time document stream using TF-IDF may be superior to Doc2Vec, the performance of the system is currently not one of the main goals of the thesis. In fact, to fulfill the goals of the system, accuracy had a much higher priority in comparison to run-time performance. Additionally, many of the training performance issues can be solved by using better hardware, a cloud-based system or even through the use of multiprocessing.

Due to the relatively large size of the data set, the model was trained separately from the system itself using the full pre-processed data set mentioned above. For the document embedding of the article contents, the Doc2Vec library from Gensim¹⁸ using the discrete Bag-of-Words model was utilized. Furthermore, the built-in cosine similarity function was used to calculate the similarity of the documents. Due to the perceived success of the comparison between the Bag-of-Words, TF-IDF and Doc2Vec document embedding methods, the same Doc2Vec parameters were used: vector size of 400; maximum distance of 15, minimum frequency of 5, negative sampling of 5, and training epoch limited to 100 in order to maximize the accuracy of predicting the relationship between articles.

3.3.3 Front-end

In this section, the front-end application will be explained. Here, the final visualizations used as well as how the individual front-end components are related to each other are detailed. Throughout the thesis, the front-end application was written in Javascript and run on Google Chrome 75.0.3770.100¹⁹.

As can be seen in Figure 3.8, the front-end application can be separated into four main layers. The first layer contains the bubble map visualization and the query system as seen in Figure 3.9. The query system allows the user to search for one or more keywords by separating the words which spaces. The system will then automatically pre-process the query to remove capitalization, punctuation and numbers and check if the remaining pre-processed words exist in the trained Word2Vec model. On the other hand, the bubble map is intrinsically linked to the query system.

¹⁸<https://radimrehurek.com/gensim/models/doc2vec.html>, last visited July 26, 2019.

¹⁹<https://www.google.com/chrome/>, last visited July 27, 2019

3 Modeling



Figure 3.9: First layer of the proposed system. The left side contains the query system and the right contains the bubble map visualization containing the top 50 most important words in the entire data set.

The bubble map visualization simply offers popular and important keywords in the data set gathered from all documents in the data set. By clicking on one of the bubbles, the system then uses the word contained in the bubble as the query for the query system.

As seen in Figure 3.10, the second layer of the system consists of a ranked list of relevant articles which is placed directly below the query box and a ring network visualization that replaces the bubble map visualization. It should be noted that the ring network visualization is only displayed if all the pre-processed words in the query are part of the Word2Vec library. If not, an error is displayed as Word2Vec does not work with words that are not included in the trained vocabulary. The user is able to interact with this visualization by clicking on any of the nodes in the ring network diagram. If the user clicks on a node in the inner ring, then the word represented by the node is added to the query refreshing the list of articles. As mentioned in the previous section, the outer ring contains a word related to the query and one or more of the inner ring nodes. If the user clicks on one of these nodes and the one of the related inner ring words is already part of the query, then the word represented by the outer ring node is added to the query. However, if no related words from the inner ring are contained in the query, then the word contained in the selected outer ring node replaces the previous query of the user. It should be noted that due to a bug in the D3Plus²⁰ library used to construct the visualization, there can be duplicates in the legend as seen in Figure 3.10. However, this does not affect the functionality of the visualization in any way. Whether the

²⁰<https://d3plus.org/>, last visited July 27, 2019

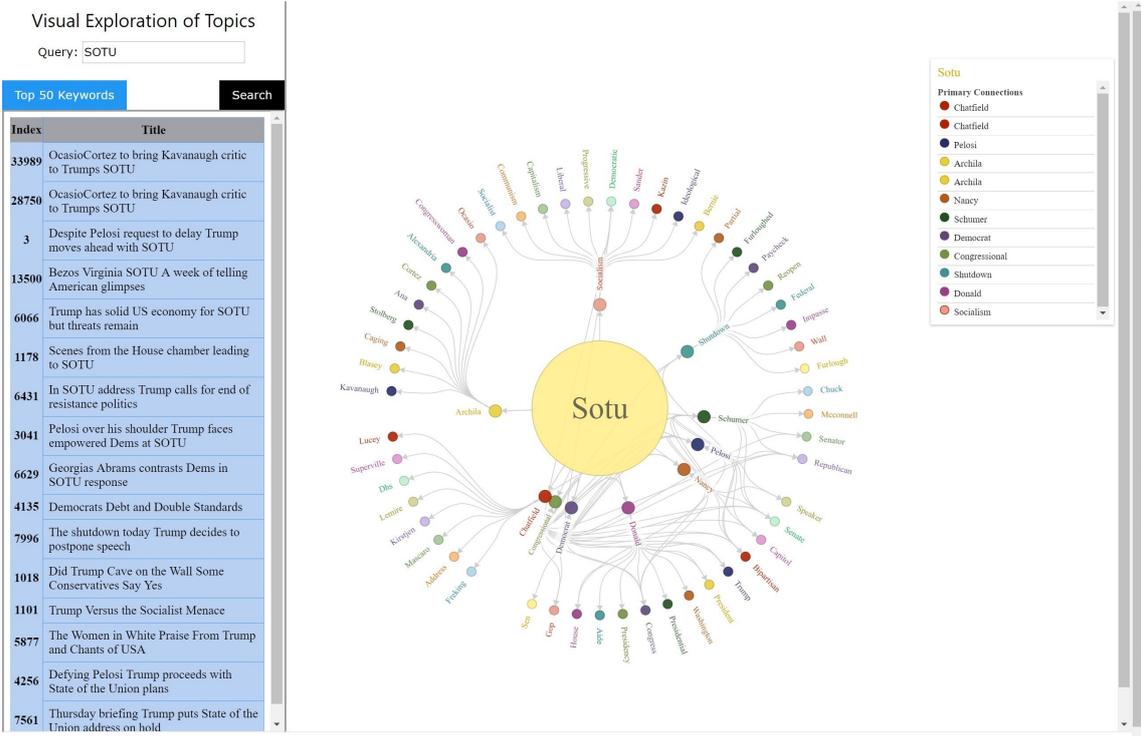


Figure 3.10: Second layer of the proposed system. The left side contains the query system and a list of articles related to the query sorted by relevance. The right side contains the ring network visualization which displays the top ten words related to the query as well as the top ten words related to the query and each of the words related to the query.

full query exists in the Word2Vec library or not, a list of article titles will still be displayed under the query system and are sorted by relevance using Whoosh!²¹. To determine the relevancy, Whoosh! uses a Bag-of-Words method and scores the number of times the words in the query appear in each article and title. Furthermore, an article is more likely to be relevant to a query if one or more words from the query are included in the title. This is because a title should include the most important keywords in the article to ensure relevancy and interest. Therefore, if a word in the query appeared in the title once, then it would be given a score of two instead of one if the word only appeared once in the article. The scores of each article are then tallied and the articles are shown from highest to lowest scores.

By clicking on a title in the list of relevant articles, the ring network visualization is then replaced with a stacked bar visualization which shows the articles related to the selected article based on their relevancy and date of publication. Only articles with a similarity value above 30 are shown in the visualization to remove completely unrelated articles. Using date filtering and data selection interactions, the user is able to reduce the amount of clutter and focus on the necessary information. More details about this visualization can be found in Chapter 3.2.2.

²¹<https://whoosh.readthedocs.io/en/latest/intro.html>, last visited July 27, 2019.

3 Modeling

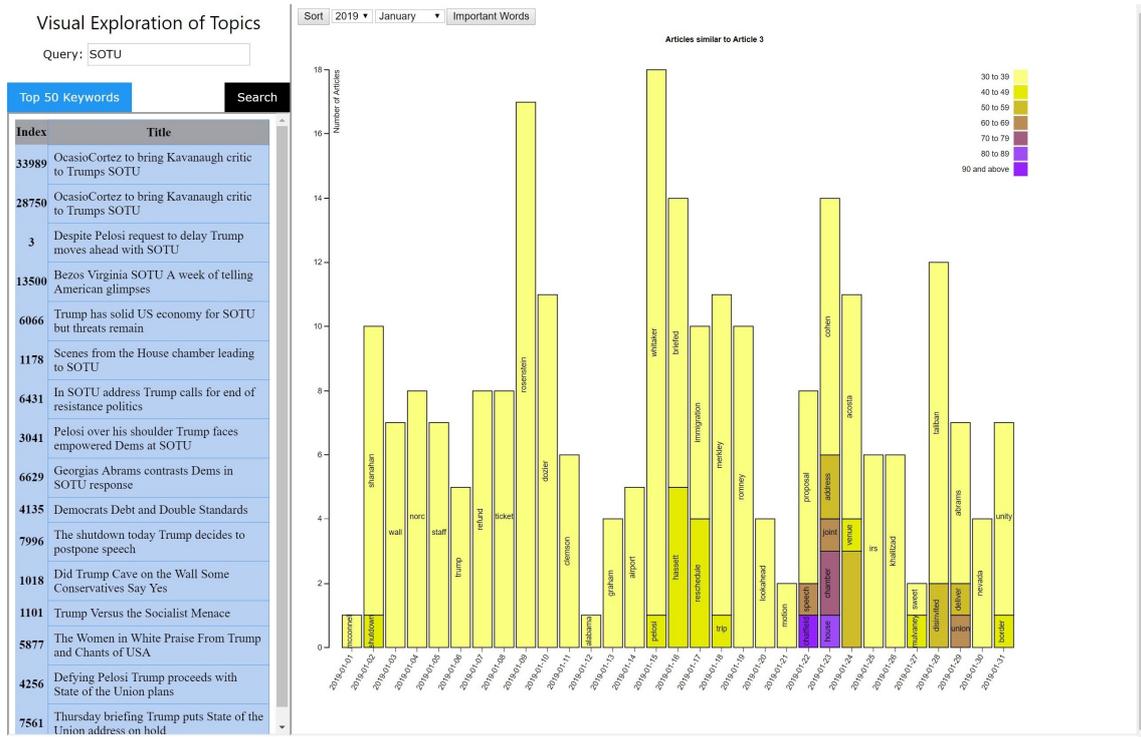


Figure 3.11: Third layer of the proposed system. The left side contains the query system and a list of articles related to the query sorted by relevance. The right side contains the stacked bar chart which visualizes the relationship between the selected article and the articles containing similar topics. In the image, the view is filtered by day for a specific year and month.

The last layer helps the analyst gather more specific details about the articles related to the selected article. By clicking the “Important Words” button available at the top of each stacked bar visualization, the user is able to replace the main view with the heat map visualization which displays the frequency of the most important words in the related articles to the selected article by date as seen in Figure 3.6. Another way of gathering details is using the details view which is an extension of the stacked bar chart as can be seen in Figure 3.12. This view can be accessed by selecting a bar in the stacked bar visualization. The view then displays the titles of articles contained in the selected bar as well as the overall most important words in the articles contained in the bar. By selecting one of the titles, the user is then able to see the details of the article such as the URL to the article, the date of publication, the most important words in the article itself as well as the text of the article.

3.3.4 Challenges

The main challenge of the design implementation was the overall performance of the pre-processing, back- and front-end. While Doc2Vec was found to be the most accurate and least prone to have false positives, the training time required was over 200 times longer than the comparable TF-IDF and Bag-of-Words algorithms. Furthermore, the performance issue can also be experienced while



Figure 3.12: One component of the fourth layer of the proposed system. The view is essentially the same as the third layer with the addition of a details view on the bottom of the main view which displays the articles contained in the selected bar from the stacked bar visualization.

using the system itself it required between 12 and 18 seconds to generate the data required for the stacked bar and heat map visualizations as well as the details view for the stacked bar chart. Furthermore, interacting or interrupting the system while it was generating the stacked bar and heat map visualization could lead to errors. However, as mentioned above, this challenge can be minimized through the use of better hardware or a cloud-based system.

3.4 Summary

In this chapter, many of the popular methods used for word and document embedding, topic modeling and visualization for visual analysis were analyzed. Through research and testing, TF-IDF with cosine similarity from scikit-learn²² was used for the topic modeling of words, and Doc2Vec²³ with cosine similarity for the modeling of documents. In addition, three main elements that needed to be visualized by the proposed system in order to achieve its goals were discovered: 1) a visualization that shows the relationship between articles; 2) a visualization that shows the relationship between

²²https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, last visited July 13, 2019.

²³<https://radimrehurek.com/gensim/models/doc2vec.html>, last visited July 13, 2019.

words; and 3) a ranked list that visualizes articles or words based on their importance or relevance. By comparing the different visualizations, the most expressive and effective visualizations for each element based on the criteria from Mackinlay [Mac86] were determined. From the results of the comparison, stacked bar charts with elaboration and filtering interactions to visualize the relationship between articles, both heat maps and a ring-network diagram for displaying different types of relationship between words and lastly both bubble maps and lists in table format for displaying ranked lists were used.

4 Study

The user study has the main goal of determining the effectiveness of the proposed software for the visual analysis of new trends as well as the usefulness of the system in providing easy access to important information for the users. Furthermore, another goal of the study is to determine the advantages and disadvantages of each component in the system to determine whether or not individual component serves their purpose for increasing the efficiency of the users' work flow. As efficiency is normally measured by the success as well as the amount of time it takes the user to complete a task, the time aspect is also important for the study. This can help us determine how quickly users can learn the features in the system which can help translate to the usability of the system as well as its intuitiveness. Naturally, the time aspect is also important to determine whether or not the system helps the users in the treatment group complete their tasks quicker than the system used in the control group as well.

The results of the study will support or discourage the hypothesis that the proposed system can be used as an effective and efficient way to visually analyze news articles as well as offer a way for analysts to track the evolution of news topics over time. The study will also be used to ascertain that every individual component serves a role in helping analysts gain an understanding of the articles contained in the data set without having to manually read through the article itself.

The next sections primarily deal with the implementation of the study. Here, the participants who took part in the study, the apparatus used in the system, the tasks and finally the procedures will be described in detail.

4.1 Participants

The study population consisted of twelve participants of which all were male. Of the twelve participants, four were students studying Media and Communication whereas the remaining eight were students studying Software Engineering. For the purposes of the experiment, only students working towards attaining their Master's degree were selected. A strictly localized population with similar backgrounds was used in order to reduce the probability of background or knowledge from influencing the use of the systems as well as the results of the study. In addition, before the experiment, none of the participant had any contact with the system and did not receive any information regarding the types of tasks that they would be evaluated on. All participants were between the ages of 23 and 28 with a mean age of 24.33 years ($SD = 1.60$ years). Furthermore, all participants were fluent in English to make sure that the participants fully understood the instructions, articles, tasks and survey questions. Additionally, all participants were experienced with using computerized devices and use them constantly throughout the day. Last of all, the participants were split equally between both the control and treatment groups. Both groups had exactly two students from studying Media and Communication and four from the Software Engineering department here

at the University of Stuttgart. The ages groups between the two test groups were also equalized to reduce the effect of age and experience on the results of the study with both test groups having an average age of 24.33 years (SD = 1.37 for the control group, SD = 1.80 for the treatment group).

4.2 Apparatus

For the study, two different setups were used. The control group used a 2015 13.3-inch MacBook Pro¹ with a i5-5257U processor and 8GB RAM. Furthermore, the device used a 2560x1600 resolution and was running Windows 10 Pro 1803. On the other hand, the study for the treatment group used a 12.3-inch Surface Pro 4 with a i5-6300U processor and 8GB RAM. The device was run using its native 2736x1824 resolution and like the MacBook was running Windows 10 Pro 1803. Both devices had PyCharm 2018.3.5 Professional Edition² with Flask³ running in the background for the back-end application and the front-end which the participants interacted with was using Google Chrome 75.0.3770.100⁴. Last of all, both devices used the Windows 10 Game Bar which was included with Windows 10 to record the screen of each participant.

4.2.1 Interaction

The interaction with each system is relatively straight forward. Each participant could only use each laptop's built-in keyboard as well as an external mouse. Furthermore, keyboard shortcuts were not implemented to reduce the effect of possible prior knowledge regarding computerized systems.

4.3 Tasks

In order to evaluate the effectiveness of the software for the visual analysis of new trends as well as determine the effectiveness of each component in the proposed system, a total of 11 tasks separated into two sets of tasks were devised.

4.3.1 First Task Set

The tasks in the first set of tasks were devised with a specific end-goal in mind with a correct answer in terms of the articles in the data set. Using these tasks, the participant was given some time to learn the possibilities of the software so that the study observer could determine how they used the proposed system for the second task set. The first task set consisted of the following questions:

1. Find the earliest known article in the database regarding Trump's border policies.

¹https://support.apple.com/kb/sp715?locale=en_US, last visited July 9, 2019.

²<https://www.jetbrains.com/pycharm/>, last visited July 15, 2019.

³<https://palletsprojects.com/p/flask/>, last visited July 15, 2019.

⁴<https://www.google.com/chrome/>, last visited July 15, 2019.

2. Search for “jon snow”. Is the first article the origin article for the topic “news network apologizing for Jon Snow, a news presenter (not the King in the North)”? If not, what is the origin article and when was it published?
3. Search for “SOTU”, click the third article (Index: 3, Title: “Despite Pelosi request to delay Trump moves ahead with SOTU”) and explain how the topic regarding Trump and the State of the Union has changed since the publication of the article.
4. Search for “game of thrones”. Are the first two articles highly related to each other?
5. What are the 10 most related keywords to “nfl”?
6. Search for “notre”. Name the titles of the 2 articles most closely related to “400 firefighters but Notre Dame may not be saved” (Index: 59590).
7. What is the most popular keyword for articles related to the article from Question 6?
8. Name 5 of the most popular topics associated with the keywords “hong kong” in the database?
9. In which month has the most articles about “Apex Legend” been published?
10. Pretend you are a political journalist. Name 10 of the most popular and important (descriptive, no verbs) keywords in the database.

The goal of the first task is to determine the effectiveness of the temporal aspect of the visualizations used in the proposed system. Through this task, the effectiveness of the system in determining and sorting the age of related articles over current systems such as Google News⁵ which currently does not support such features is demonstrated. The temporal aspect is important, especially in terms of topic modeling, to help determine the trends and evolution of topics over time and helps in the aspect of plagiarism or quotation detection. To solve the task, it was hypothesized that the participants would mainly use the stacked bar chart diagram as it provides an overview of related articles in temporal order.

The goal of the second task is similar to that of the first task. However, it differs from the first in that the task is much more specific and evaluates the participants understanding of the topics in the article as well. Here, the strengths of automatic topic similarity detection as well the topic comparisons between related articles in addition to the value of temporal-based visualizations are demonstrated. This type of task can be viewed as necessary as it is common nowadays to find articles or topics which are based on previous articles. Furthermore, it is a good way to establishing what information has already been written about and what is new. However, computerized algorithms to extract or process data are not infallible and can contain mistakes such as algorithmic miscategorization, typos or incomplete data. This task will contain an attention check for the participants using the proposed system as the article is miscategorized as an article published in January in the monthly view. If the participant double checks the answer either by changing the view to the daily view or even just clicking on the article, the user should be able to tell that it is a case of miscategorization and notify the study observer. As with the first task, heavy use of the stacked bar chart in conjunction with the detailed view for topic authentication purposes was hypothesized.

⁵<https://news.google.com/>, last visited July 10, 2019.

The third task focuses on discovering the effectiveness of system for topic modeling. Through this task, how the participant used the system to determine the difference in topics between related articles were evaluated. This type of task is particularly important as exploring topic trends is a very effective way of determining the popularity of future articles and topics which can directly contribute to revenue. For this task, it was predicted that the stacked bar chart as well the top 5 TF-IDF words and new TF-IDF words for each category of similarity would be of significant importance as it briefly summarizes the most important topics of clusters of relevant articles without requiring the analyst to actually read the article itself saving valuable time. Furthermore, the use of the heat map visualization could prove useful in determining the changes in important keywords over time as the frequency of keywords for all relevant articles is also sorted by time.

The fourth task focuses on the determining the relevance between two articles that seem very relevant on the surface. This can be seen as a sort of attention check for the participants as the titles and first few sentences are very similar. However, the main body of the articles are not as relevant as the titles and description might suggest. As mentioned before, time is a very valuable resource and as both articles are relatively long in length, it was hypothesized that there would be a significant decrease in time used to determine the relevancy between the two articles through the use of automatic topic modeling and similarity calculations. For this task heavy usage of the stack bar chart was expected as the visualization focused on article similarity.

The fifth task attempts to evaluate the system's ability to streamline the process of determining the most relevant or important words for a topic. This is especially important if the analyst does not have in-depth knowledge about a topic. This task was designed around the concept of using the ring network diagram described in Chapter 3. Through this visualization, the analyst should be able to complete this task quickly and efficiently without having to scroll through and read the titles and articles of all relevant articles in the database which can amount to thousands or more.

The sixth task is similar to the second and fourth in the aspect that it assesses topic similarity. Here, the participants are asked to find the two most similar articles to a specific article which can be useful when looking for sources and citations on a particular topic. Using the stacked bar visualization, the amount of reading required by the analyst is reduced as the information is visualized in such a way that they can immediately determine the most similar articles.

The seventh task is comparable to the fifth task in that the analyst has to find the most relevant keyword for the particular topic. This task was designed with the heat map visualization in mind as it indicates the frequency of each of the most important words in the cluster of related articles. By comparing the frequencies, the analyst is shown the keyword that is most often related to the topic. With this feature, it could potentially suggest to the analyst or journalist must-have words when writing articles or documents related to the topic.

The eighth task is similar to that from the fifth task. However, instead of solely looking for the most relevant topics, the participant also has to look for the most popular ones. For this task, the participants should use multiple views simultaneously, in particular the ring network visualization in combination with the list of relevant articles.

The ninth task is to simply determine the month where a topic is more popular. This task simulates the real world where an analyst, journalist or even businesses needs to determine the popularity of a topic over time. With this knowledge, journalists can determine whether or not it is worthwhile to write over a topic and analysts and businesses can use this knowledge to decide whether more

marketing needs to be done for a product and calculate the long-term demand and hype for their products. For this task, the stacked bar visualization would be the most apt way to determine the answer to this task as it visualizes the total amount of related articles per year, month and day.

The last task in the first task set is slightly more open-ended than the other tasks in the set. In this task, the participants are given free reign over what keywords to query. However, it does require some knowledge in politics and politics-related words. However, the difference in knowledge between the participants should be minimized due to their similar ages, field of study as well as interests. For this task, it was hypothesized that the participants would use the bubble map visualization containing the top 50 most important words in the database. Using this visualization, the participants would be able to simply select the largest bubbles with politically-affiliated words.

4.3.2 Second Task Set

In contrast to the tasks in the first task set, there is only one task in the second task set and the question is much more open-ended with no correct answer. Rather the goal of the task was to determine how the participants used the system after familiarizing themselves with the system from the tasks in the first task set. To do so, the task was formulated in such a way in order to simulate real-world usage. Here the participants should use the software under the pretense that they are playing a specific role. In particular, each participant was asked to complete the following task: “Pretend that you are a political blogger. Choose a main topic (unrelated to “Trump”) and find 3 related subtopics to write about (3 keywords per subtopic) and what sources you are basing your opinion on (at least 3 sources per subtopic).” With this task, the participant should simulate a political journalist writing three articles based on the chosen main and subtopics where the keywords and sources simulate the content and research for the body of the articles. For this task, rather than the results of the task, the process in which the participants gather the information required to complete the task is the most important as it helps us determine how the software is used with a real-life use-case.

4.4 Procedure

The procedure of the user study can be split into five main steps. First of all, the participants were told the goal of the study, what would be recorded and the general process of the study. After recording personal information such as age, gender and field of study, the participants were given a demonstration of the system where the study observer which doubled as the interviewer would explain the features of every visualization in the system. In the third step, participants were then given five minutes to try out the system as a form of training and familiarization and ask the study observer any questions they may have. After the five minute familiarization period, the main body of the study begins and the screen recording starts. It should be noted that during the study, the observer was not allowed to give the participants any hints or reminders about the available features in order to simulate real-world usage. The main body of the study consists of two parts. In each part of the main body of the study, the participants were asked to complete the first and second task sets respectively. However, it should be noted that the participants were given the tasks in the first part of the main study in random order. For all tasks, the time and screen of the participant will be recorded. Furthermore, due to the performance issue from the proposed system which could

be solved by more optimized code or better hardware, the loading times from the system will be deduced from the raw times. In addition, by recording the screen as well as from the notes taken by the observer, the actions of the user will be noted to determine how they are using the system. After completing each task, the participants will be asked by the observer to explain which component of the application made them decide that their answer was the “correct” one. After completing all tasks, the participants will be asked to complete two surveys, NASA-TLX [HS88] and a survey that documents their feedback on the software and visualizations. In the latter survey, the questions are based off the Likert scale [AS07] and also included questions which required the participant to comment on the various components in the proposed system.

4.5 Difference between Control and Treatment Groups

As mentioned in the previous sections, the study population was split into two groups: a control group and a treatment group. The control group consisted of six people who were to use a extremely simplified version of the proposed system in order to simulate the effect of a “Google News Search” used by many journalists and influencers without access to specialized software. In essence, the simplified system only contained three components: a query component; a list component which visualized the titles of the relevant articles as a list; and a details component which showed the title, link to the article, publishing date and body of the article as text. On the other hand, the six members of the treatment group were asked to complete the tasks using the proposed system described in Chapter 3.

4.6 Summary

In conclusion, the user study had a population of twelve participants. In the study, the participants were asked to complete a series of tasks which could be relevant in the real-world to simulate use-cases. Using the tasks, the visualizations evaluated on whether they were effective for their purpose of visual analysis and tracking of topic evolution over time and whether it is more effective than the use of publicly accessible topic and document retrieval tools such as “Google Search” used by many people today. The next step will be to determine whether or not the study results correlate with the theoretical results and hypotheses as well as determine from the results and feedback from the study to determine the usability and effectiveness of the proposed system in its current form.

5 Study Evaluation

Here, the results from both the control and treatment groups of the study are evaluated in order to determine if the hypothesis that the proposed system with its combination of visualizations and topic and document modeling can be used as a more effective way for analysts to perform visual tracking and analysis on web content. Along with the results of the tasks, this chapter will also evaluate the results of the two surveys done by the participants after the study as well as the live and spectating comments of the participants. This chapter is split into two main parts. The first section will present the raw results from the study. The second section will focus on the analysis of the results presented in the first section.

5.1 Results

In this section, the results from the first and second task set described in Chapter 4 are reported. Here, how the participant from both the control and treatment groups obtained their results are explained, compare their results with the expected results as well as chronicle the time needed for every task. Last of all, the results of the surveys and interviews done during and after the study will be presented.

5.1.1 Task Set 1

After processing the raw times and removing the times when the participants were explaining their answers, the control group took an average of 2,048 seconds ($SD = 546.52$) to complete the first task set as seen in Figure 5.1 and 415 seconds ($SD = 131.51$) to complete the second task set. To calculate the correctness of an answer, each answer from the participant was scored between 0 and 1. For tasks which only require one answer such as finding the earliest known article in the database regarding a certain topic, each participant could only get either 0 or 1 points for their answer. On the other hand, for questions that have multiple answers such as tasks 5, 8, and 10 which require the participants to list ten, five and ten items respectively, the correctness of the answers is based on the percentage of answers that were correct. Therefore, if a participant got 5 out of 10 items correct, then they would get a total of 0.5 points. Overall out of a maximum of ten points, the participants in the control group only managed to get a mean score of 2.44 ($SD = 2.57$).

On the other hand, the participants in the treatment group had an average total task time of 879.17 seconds ($SD = 264.23$) for the first set of tasks and 391.2 seconds ($SD = 220.60$) for the second. For all the times from the treatment group, the processing time to display the visualizations and amount of time needed to explain their answers and ask questions were removed from the raw times. Using

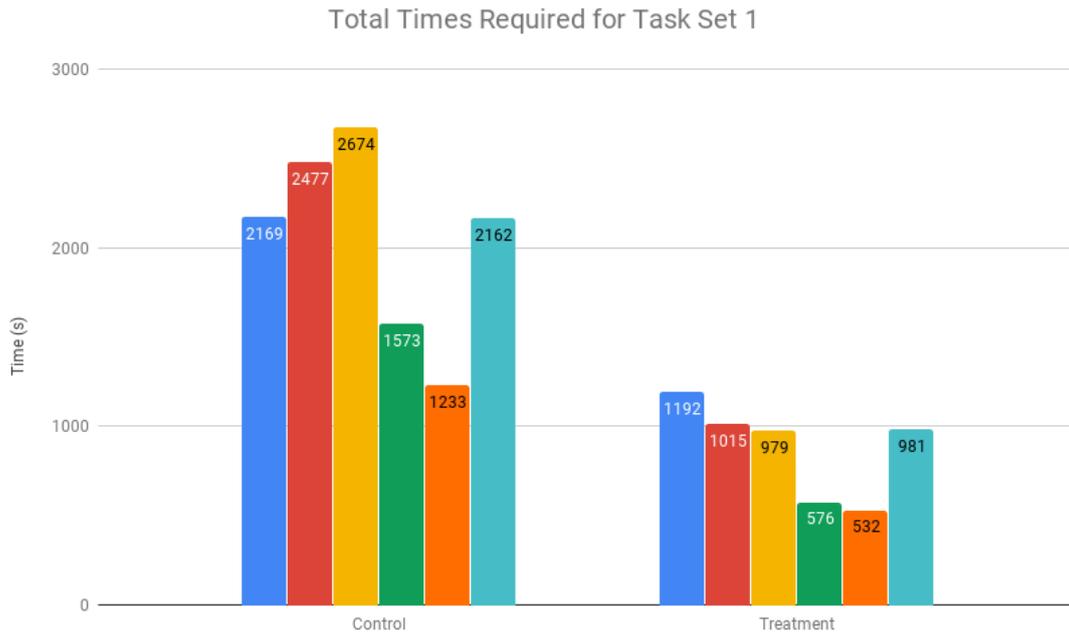


Figure 5.1: This graph shows the amount of time required by each participant in the control and treatment group to complete all tasks in the first task set.

the same point scale as the one for the control group, the participants in the treatment group using the proposed system had an average score of 4.4 (SD = 1.98). Out of a maximum of ten points, the participants in the control group managed to get a mean score of 4.4 points (SD = 1.98).

One common approach to evaluate the correlation between data sets is to calculate a t-test. As both the control and treatment group had equal populations, the one-tailed two sample variation of the t-test was used. Furthermore, as there was a large difference in variance between the two sets of data (9144.74 for the control group and 1762.19 for the treatment group), the t-tests were calculated assumed unequal variances. To calculate the t-tests, the average results from both the control and treatment groups were used. For the t-test, the following null and alternative hypotheses are assumed:

Null hypothesis 1: The treatment group does not complete the task(s) faster than the control group.

Alternative hypothesis 1: The treatment group completes the task(s) faster than the control group.

Null hypothesis 2: The treatment group does not perform the task(s) more correctly than the control group.

Alternative hypothesis 2: The treatment group performs the task more correctly than the control group.

Assuming an alpha value of 0.05, a t-test using the average times of all tasks in the first task set done by the control group and treatment group respectively was calculated. From the t-test, a p-value of 0.004 was obtained which is lower than the alpha value. Therefore, across all tasks in the first task

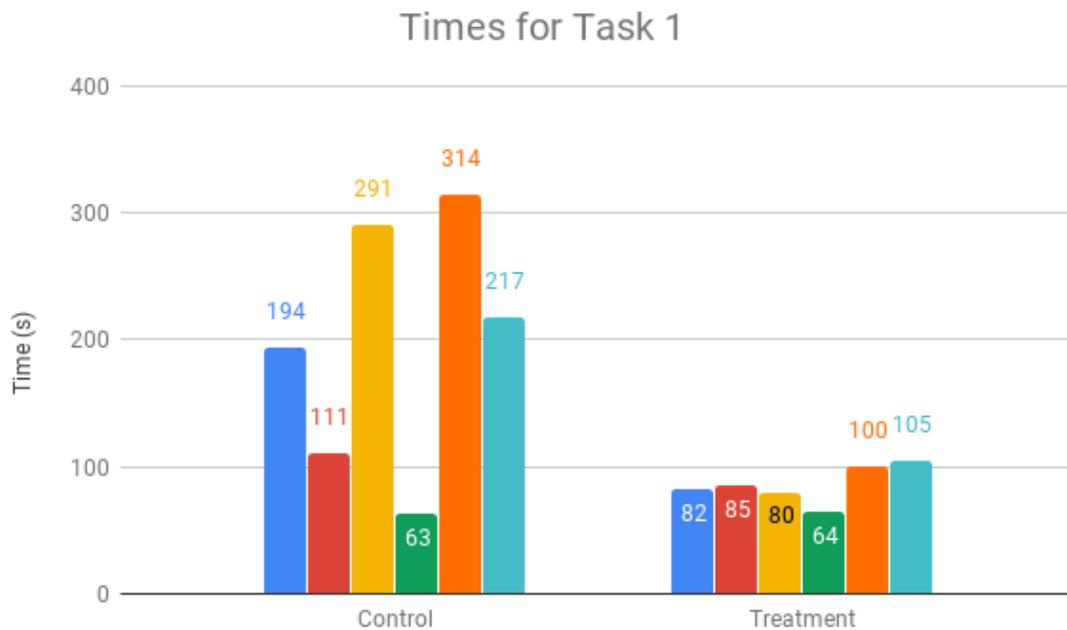


Figure 5.2: This graph shows the amount of time required by each participant in the control and treatment group to complete the first task in the first task set.

set, the resulting p-value indicates the treatment group completed the set of tasks faster than the control group. In terms of performance, using the combined scores for each task and group, the t-test returns a p-value of 0.036. This result indicates that overall, the treatment group performed the tasks in the first task set more correctly than the control group.

Task 1

As mentioned in Chapter 4, the first task was to find the earliest known article in the database regarding Trump's border policies.

In this task, it was expected that the control group participants would click through the list of the articles and find the article with the earliest date while making sure that the topic was relevant. Naturally, they were not expected to read every single article due to time and motivation restraints, but rather simply read through the titles to ensure that the articles are relevant to President Trump's border policies. The recordings from the study reflected this as all the participants simply clicked through the list of articles so that they could see the publishing date. While some answers were close, ultimately none of the participants managed to get the answer correctly, even though there were multiple answers available. The expected answer was one of the many articles published on January 1, 2019 regarding President Trump and the border between the United States and Mexico. However, the participants' answers ranged from January 2 to 31 with none managing to find a relevant article published on January 1. To complete this task, the participants in the control group needed an average of 198.33 seconds ($SD = 98.26$) with a minimum and maximum time of 63 and 314 seconds respectively as can be seen in Figure 5.2. Additionally, in context of the total

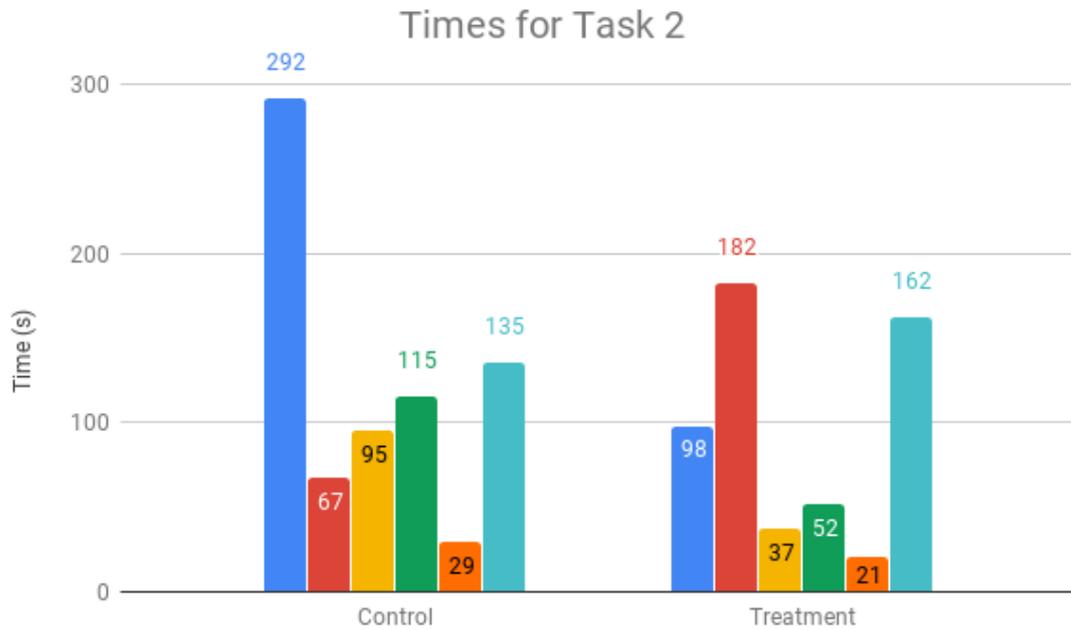


Figure 5.3: This graph shows the amount of time required by each participant in the control and treatment group to complete the second task in the first task set.

time needed for the participant to complete all tasks in the first task set, each participant needed an average of 10.64% (SD = 7.81) of the total time they spent to complete all ten tasks in the task set to complete the first task.

For the treatment group, the participants were expected to search for a term such as “trump border”, click the first article in the list as it is sorted from most to least relevant and use the stacked bar chart filtered by date to determine the earliest known article regarding President Trump’s border policies. From the observations of the study, this was true for half the participants with the other half searching for other queries such as literally “trump’s border policies”. For this task, three participants in the treatment group who used the expected query answered it correctly. As can be seen in Figure 5.2, the treatment group required an average time of 86 seconds (SD = 14.8), which is 10.67% (SD = 4.29) of the total task time to complete the first task.

Task 2

The second task required the participant to determine if the first article in the list of articles relating to the query “jon snow” was the origin article for the topic “news network apologizing for Jon Snow, a news presenter”. If the first article was not the origin article, then the participant should explain which article was the origin one.

As with the first task, heavy usage of the list of article titles was expected to determine relevant articles to click on so that the control group participants could determine the publishing date. Furthermore, the control group participants were expected to be able to answer this task relatively

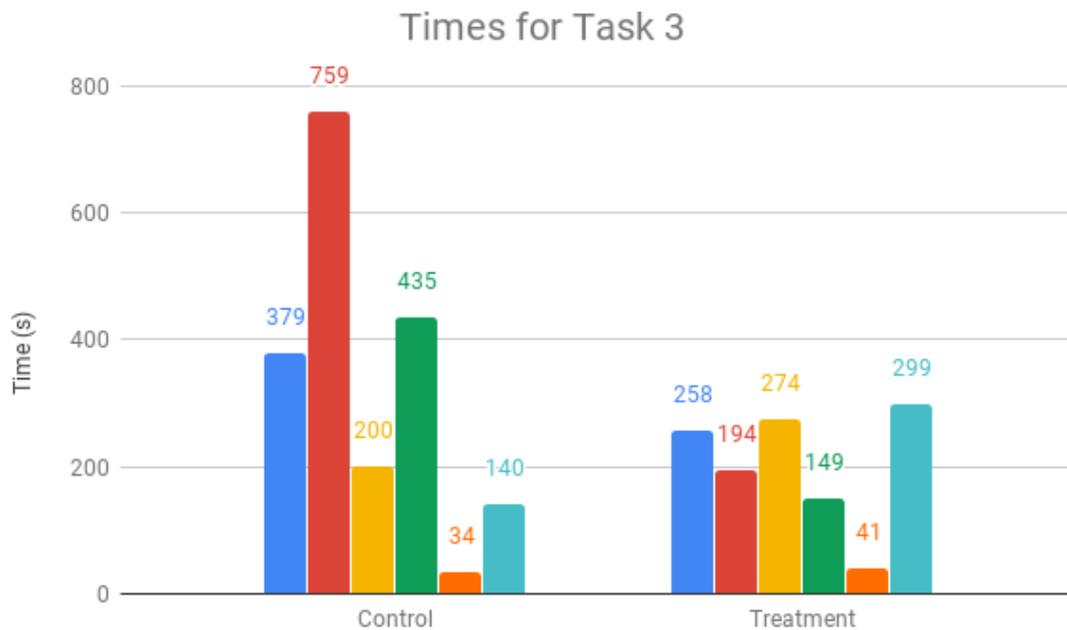


Figure 5.4: This graph shows the amount of time required by each participant in the control and treatment group to complete the third task in the first task set.

quickly as there were just four highly relevant articles ranked right at the beginning of the list. The results of the study validate this hypothesis as all six participants in the control group managed to get the correct answer. In addition, the participants on average only needed 122.17 seconds ($SD = 91.13$) with a minimum and maximum of 29 and 292 seconds to complete the task as seen in Figure 5.3. This is the second shortest average time per task overall for the control group and only required a mean of 5.94% ($SD = 4.19$) of the average total task set 1 time to complete.

The second task, as with the first task, required the use of the stacked bar visualization due to its temporal aspect. However, as mentioned in the previous chapter, an attention check was built into the task to ensure that the participants were paying attention to the details. Unfortunately, none of the participants noticed the inaccuracy in the data as the participants and they all provided the incorrect answer. As can be seen in Figure 5.3, the participants in the treatment group needed on average 30.17 seconds less than the control group ($SD = 67.38$) to complete the task and the time only consisted of 9.90% ($SD = 6.08$) of the total task time.

Task 3

The third task requires the participant to understand the topics of a specific article and compare the topics of that article with relevant articles with later publishing dates. Using this information, the participant should then infer using the system how the topics relating to “Trump” and the “State of the Union” has changed.

While some information can be gleaned from the titles of the articles, the participants in the control group were expected to actually read the articles in order to process the changes. Furthermore, as the articles are sorted based on relevancy instead of time, the participants in the control had to also be mindful of the publishing date of the relevant articles they clicked. Furthermore, as the task requires the participants in the control group to read the articles themselves and make inferences themselves based on the different articles they read, it was hypothesized that this task would occupy a much longer amount of time in comparison to the first and second tasks. The recordings and observations of the study largely reflects these theories. The participants in the control group all first read through the specified article. The majority of the participants then found relevant articles through the titles and read the articles in the correct time period. However, it should be noted that one participant simply based his answer through the difference in titles and two participants simply glanced through the articles and took note of words that stood out such as words with a high repetition rate or quoted phrases. From the results of the study themselves, the all the participants managed to answer the question. However it should be noted that the answers themselves, while acceptable, were very general and did not contain very specific information. For example, many of the answers simply stated that the focus of the articles went from being about the delay of the State of the Union address to about the contents of what will be addressed in the State of the Union address. While correct in itself, the answers based on quickly glancing through and comparing the texts of different articles were not specific enough to deduce any real information such as concrete topics that will be discussed in the State of the Union or other information outside of the State of the Union address itself but still relevant to the topic. Of the ten tasks, this task also took the participants in the control group the second longest mean time to complete with a average time of 324.5 seconds ($SD = 259.96$) and a minimum and maximum of 34 and 759 seconds as seen in Figure 5.4. In comparison with the total times overall, the participants in the control group needed on average 15.41% ($SD = 11.74$) of their total times to complete this task.

For the treatment group, the participants were expected to use the heat map and stacked bar chart visualizations to determine the evolution of topics after a certain date. However, from the observations, only two of the six participants in the treatment group used the heat map visualization to help answer and explain the task with the other four only using the information provided by the stacked bar visualization. As with the control group, all six participants answered the question correctly. Nonetheless, the answers obtained from the treatment group were more detailed than those from the control group with the participants giving explicit changes in topics such as the focus from “border” to “economics” in related articles or how “pelosi” became a less popular topic related to the State of the Union and President Trump in the publications that came after the specified articles. For this task, the treatment group required by far the longest average amount of time to complete at 202.5 seconds ($SD=96.46$) totaling an average of 22.13% ($SD = 8.19$) of the total task time.

Task 4

The goal of the fourth task is to determine if two specific articles are highly related to one another. For the purposes of the experiment, two articles were considered “highly related” if the two articles contained topics that were largely the same.

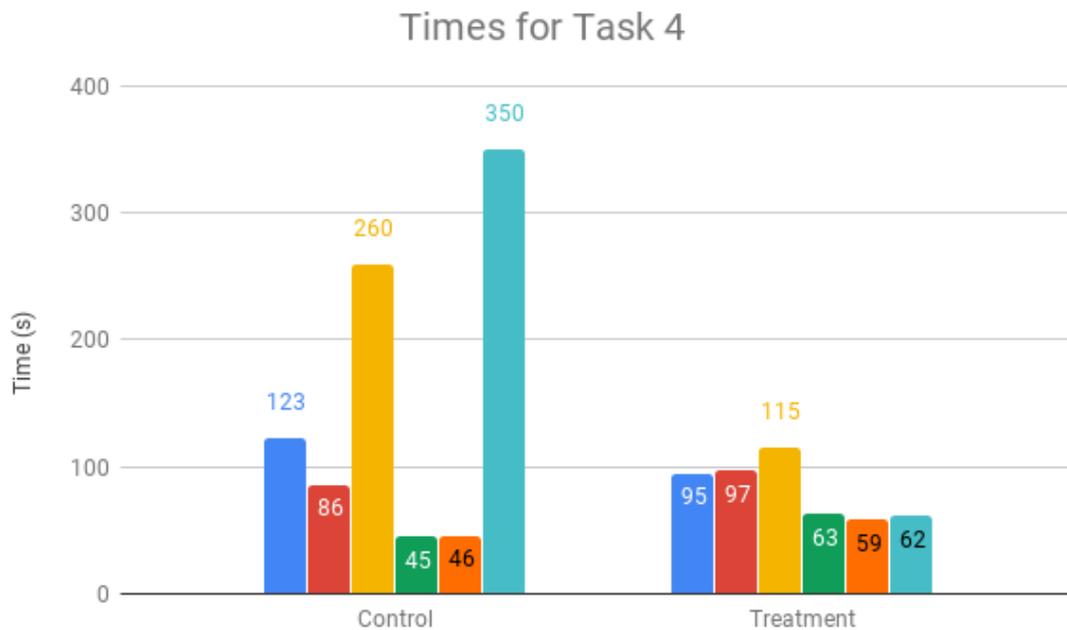


Figure 5.5: This graph shows the amount of time required by each participant in the control and treatment group to complete the first task in the fourth task set.

As mentioned in the previous chapter, this task served the dual goal of making sure the participants in the control group were paying attention to the content of the articles themselves rather than quickly reading through the titles and first few sentences of the articles. Due to the specific nature of the task, the participants of the control group were expected to read through both articles. However, the observations from the study reflect otherwise. Of the six participants, only half read more than the titles and introductions of both articles. Furthermore, of the three that spent more time reading the articles, none of them completely read through both articles as they were relatively long and rather mainly glanced through the articles looking for common keywords and phrases. From the results of the study only two members of the control group noticed that the articles were not highly relevant to one another despite their highly similar titles and introductions. As can be seen in Figure 5.5, the participants in the control group finished the task relatively quickly with an average time of 151.67 seconds ($SD = 125.50$) which consisted of an average 6.94% of their overall task time.

For this task, the participants in the treatment group were expected to click through the multiple bars in the stacked bar chart using the by year view to determine the similarity rating of the second article in comparison with the first. This is shown in the observations of the study. For this task, all six participants in the treatment group determined that the two articles are not highly related as the second article had a similarity value of between 40 and 49 when compared to the first article and vice versa. The treatment group needed an average of 81.83 seconds ($SD = 23.55$) to complete the task which consisted of 9.60% ($SD = 2.10$) of the total task time.

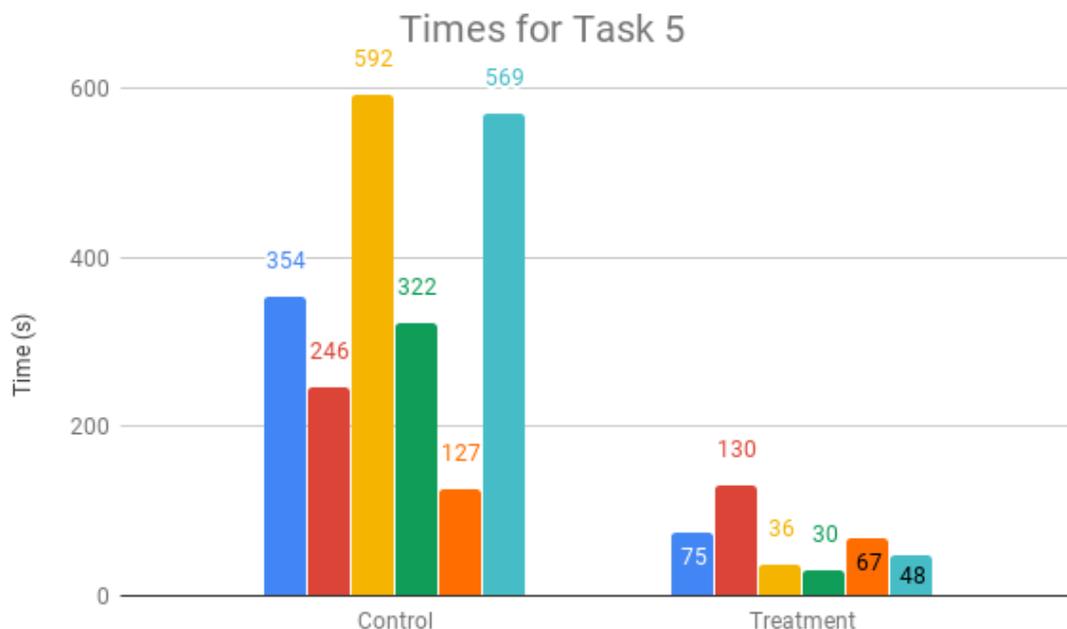


Figure 5.6: This graph shows the amount of time required by each participant in the control and treatment group to complete the fifth task in the first task set.

Task 5

The fifth task asked the participants to search for the 10 most related keywords in the database to the query “nfl”. For this task, “related” was defined as important and descriptive keywords that were frequently mentioned when discussing about topics related to the NFL¹.

For the participants in the control group, the participants were expected to obtain these keywords by searching through the titles as well as matching frequent keywords that appeared in the articles related to the NFL. Furthermore, as the content of the articles were not highly relevant to the task but rather just the keywords themselves, the control group was expected to require less time for the task than the expected times for task 3. The recordings from the study reflect the process in which the participants obtained their results but not the time required. In fact, this task required the longest time to complete in comparison with the other tasks with a mean time of 368.33 seconds (SD = 182.13) and necessitated an average of 17.58% (SD = 6.61) of the total task time. However, in the study, the length of time did not directly correlate with the correctness of the answers. On average, the control group participants only got on average 2.67 out of 10 keywords correct with a total of just 1.6 points out of the 6 possible points for this task.

For this task, the participants were expected to use the ring network visualization to determine the ten keywords most often related to NFL in the data set. All the participants used the proposed system as expected which provided the correct answers. Due to the simplistic nature of the task

¹<https://www.nfl.com/>, last visited July 15, 2019.

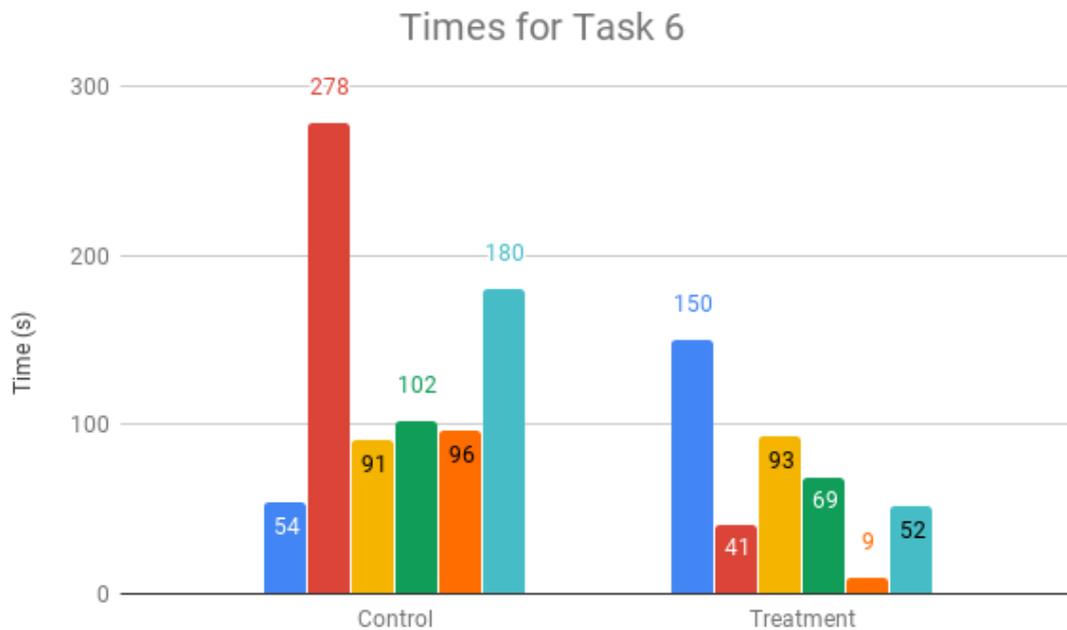


Figure 5.7: This graph shows the amount of time required by each participant in the control and treatment group to complete the sixth task in the first task set.

when using the proposed system, the participants needed a mere 64.33 seconds ($SD = 35.55$) to complete the task which is the second shortest amount of time for any task in the treatment group as seen in Figure 5.6. Due to the short amount of time needed on average, the time required for the task only required 7.58% ($SD = 4.05$) of the participant's total task time.

Task 6

The sixth task is similar to the fourth task. Only, the participants needed to explore the dataset and find the most related articles themselves instead of just confirming or denying if two articles were similar.

With the control group, due to the limited set of interactions and visualizations, they were expected to simply read the titles of the related articles to find the most relevant ones and confirm by reading the articles they thought were related. The observations confirm this to a certain extent but many of the participants in the control group simply compared titles to infer the topics and at most quickly glanced through the articles to see if they were relevant. To this effect, while the articles that they chose were relevant in some way to the specified article, none of the articles they chose were the most relevant in the data set. The duration needed to complete this task was surprisingly low at 133.5 seconds ($SD = 81.94$) and only took up 6.62% ($SD = 3.25$) of the total time needed for the tasks in the first task set as seen in Figure 5.7.

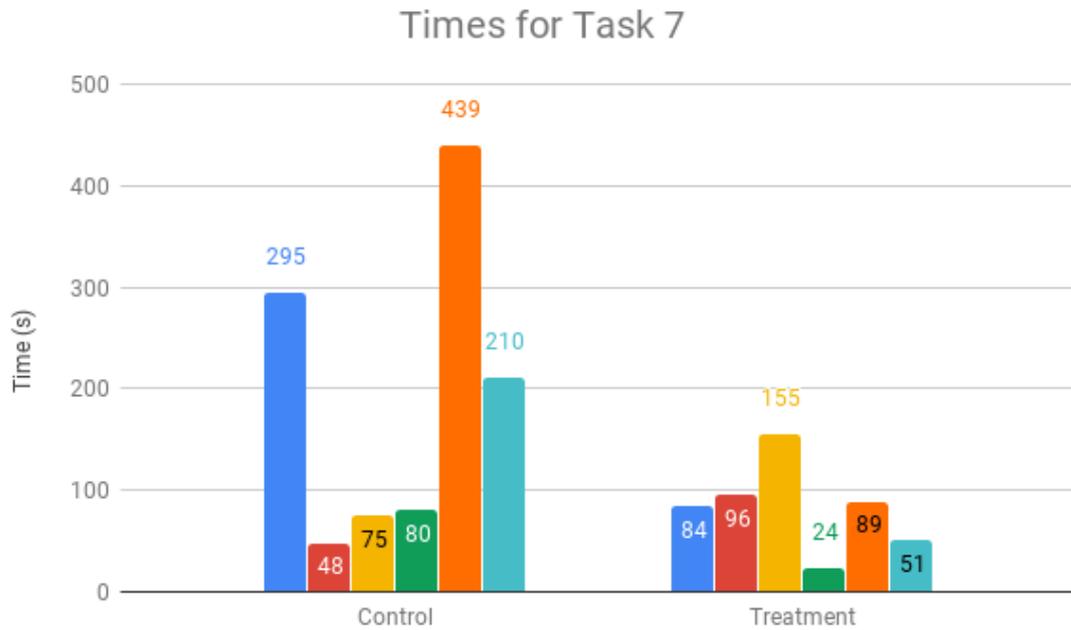


Figure 5.8: This graph shows the amount of time required by each participant in the control and treatment group to complete the seventh task in the first task set.

For the sixth task, it was assumed that there would be heavy usage of the stacked bar chart visualization by the treatment group as it is the only visualized way to determine the similarity between documents. The observations prove this hypothesis as five out of six participants simply clicked on the article, then clicked the bars that were most purple to find the articles with the highest similarity. On average, the participants in the treatment group needed 69 seconds ($SD = 48.6$) to complete the task. In addition, the task only used 7.52% ($SD = 4.48$) of the total task time required by each participant in the treatment group.

Task 7

The seventh task required the participants to find a single keyword that was mentioned the most often when discussing the Notre Dame.

For the control group, they were required to read through the articles and titles and manually determine the most frequent keyword related to the Notre Dame. Of the six participants in the control group, only 50% did more than a cursory glance through the titles and articles with an average task time of 191.17 seconds ($SD = 154.21$) which was 11.46% ($SD = 12.62$) of the total task time. However, only 2 out of 6 participants managed to find the correct answer. It should be noted that the participants that took the longest and third longest time to browse through the titles and articles before giving their answer (seen in Figure 5.8) were the ones that found the correct answer.

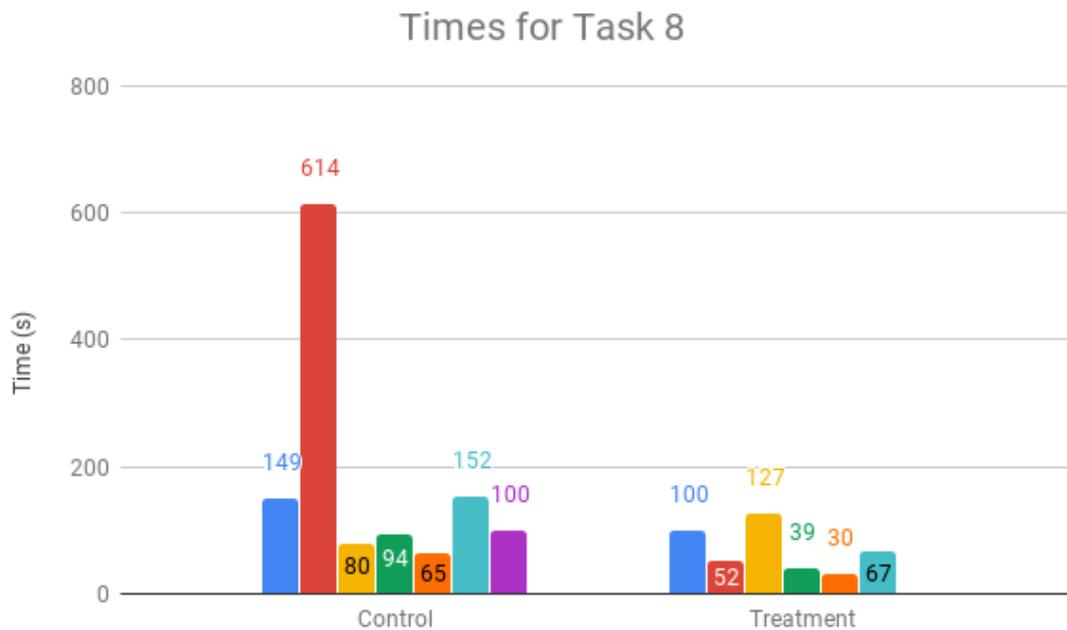


Figure 5.9: This graph shows the amount of time required by each participant in the control and treatment group to complete the eighth task in the first task set.

For the treatment group, the participants were expected to use the heat map visualization for the specified article to determine the most popular keyword related to it. All participants in the treatment group used the system as expected which meant that all participants answered the question correctly. The participants in the treatment group needed on average 83.17 seconds ($SD = 44.49$) to complete the task which consisted of 9.74% ($SD = 5.38$) of the average total time needed for all tasks.

Task 8

The eighth task is highly similar to the fifth task, but with a keyword that the participants had a low probability of knowing much about.

As with the observations from the fifth task, the control group participants mainly glanced through the titles and articles related to the specified keyword. However, as they only needed to determine five keywords instead of ten, the participants managed to complete the task with an average time of 192.33 seconds ($SD = 209.67$), 8.82% ($SD = 7.96$) of the mean total task time. However, despite the slightly time per keyword of 38.47 seconds per keyword in comparison to the 36.83 seconds for the results in task 5, none of the participants in the control group found any of the expected five most related keywords to the topic “Hong Kong”. This is despite the fact that there were ten different acceptable keywords due to the high relatedness and frequency of the keywords in the database overall. As with the fifth task, the participants in the treatment group were expected to use the ring network visualization to determine the most popular and highly related keywords of a specific term. As with the control group, there were ten acceptable keywords and the participant would receive a full score of one if they had five keywords from the list of ten acceptable keywords.

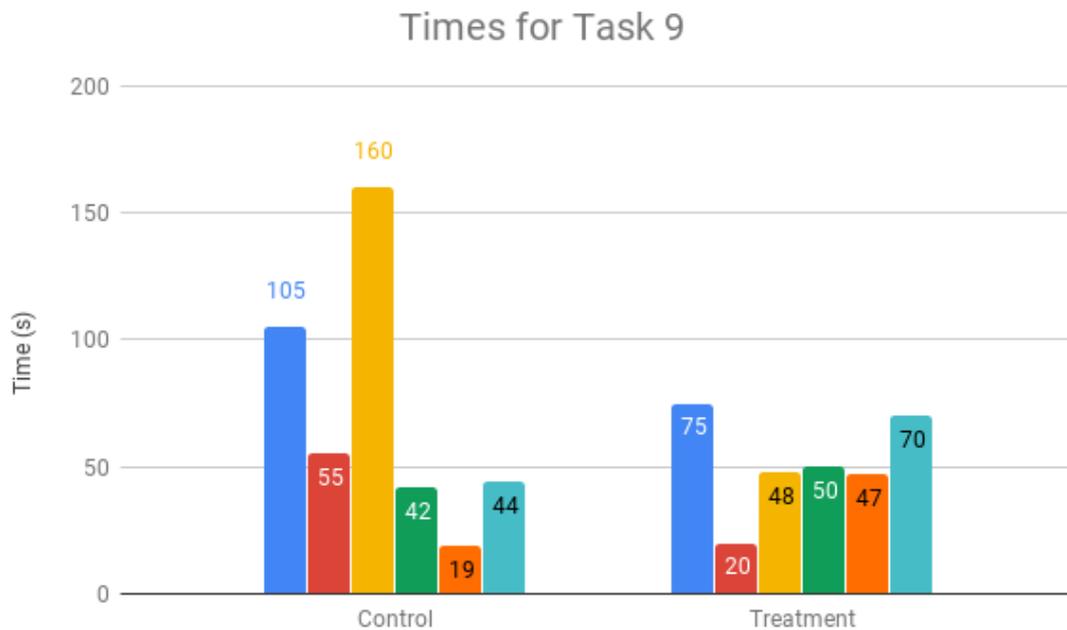


Figure 5.10: This graph shows the amount of time required by each participant in the control and treatment group to complete the ninth task in the first task set.

As predicted, the majority of the participants in the treatment group used the system in the way that it was designed and five out of six participants had answers that were completely correct. As this task is relatively simplistic and the system did not require the participant to read much if at all, task required an average time of 69.17 seconds ($SD = 37.52$) which was normally 7.62% ($SD = 2.85$) of the participants total task time.

Task 9

The ninth task simply needed the participants to determine in which month articles about a specific topic have been written about the most.

For the participants of the control group, they were expected to simply find the most related articles through the titles and memorize the months in which they have been written. The observations from the study did not disprove the hypothesis and all members of the control group answered the question correctly. This task required the shortest mean time to complete at 70.83 seconds ($SD = 52.17$) and took merely 3.22% of the average total task time.

For this task, the participants were expected to use the stacked bar chart to determine the frequency of highly relevant articles to the term “Apex Legend”. As the articles are sorted by relevancy to the query, the participants were expected to select the first article in the list and use the monthly view in the stacked bar chart to determine the answer. Furthermore, it can surmised that since the question explicitly asked for articles about “Apex Legends”, the participants would notice that only articles with a similarity value of 40 and above in comparison to the first article would pass that

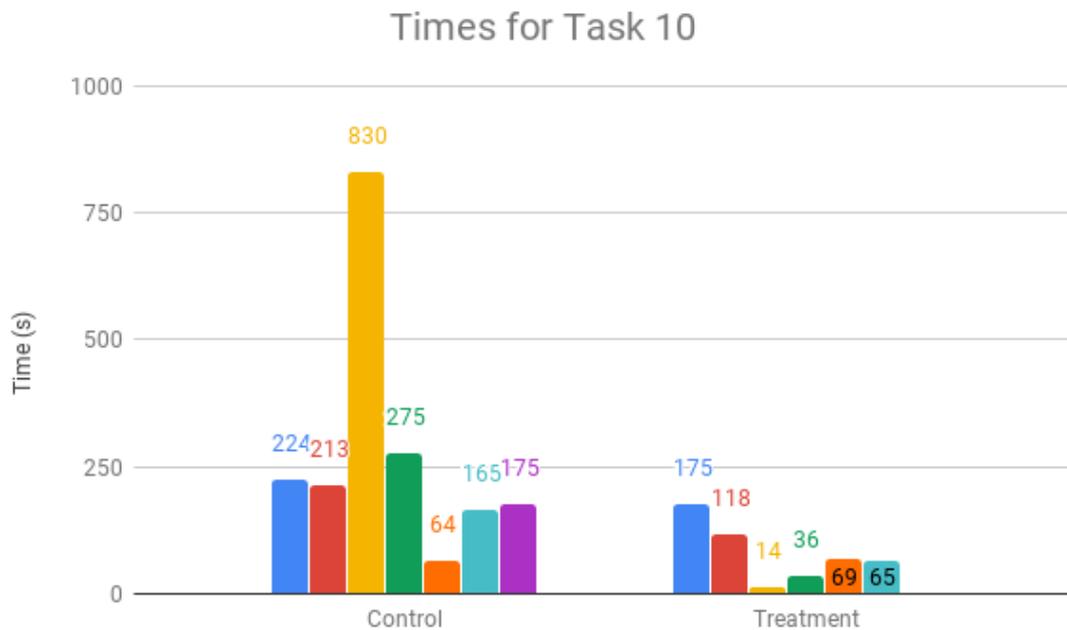


Figure 5.11: This graph shows the amount of time required by each participant in the control and treatment group to complete the tenth task in the first task set.

criteria based on the titles of the articles contained in the categories. However, only four out of six participants noticed this with the other two including the articles with similarity values between 30 and 39. The treatment group solved this task the quickest among all tasks in the set with an average time of 51.67 seconds ($SD = 19.60$) as seen in Figure 5.10. Furthermore, on average, the treatment group only spent 6.30% ($SD = 2.59$) of their total task time to solve this task.

Task 10

The last task in the first task set required the participant to find the ten most popular keywords in the database that is somewhat related to politics.

For this task, the control group was expected to use prior knowledge when seeking for the relevant keywords. In fact, the participants were expected to use their knowledge about politics to search for politically-related terms such as “government” or “trump” and roughly gauge the amount of articles for each term. However, the observations were different from the expected process. Here, four out of six participants simply searched for “politics” and based their answers on keywords they believed appeared frequently. From the 60 possible answers provided by 6 participants in the control group, only 8 were correct. On average, each participant only correctly determined 1.33 out of the 10 possible keywords. As can be seen in Figure 5.11, the participants needed an average of 295.17 seconds ($SD = 271.53$) and 13.38% ($SD = 9.60$) of the total task time to complete the task.

As with the control group, the participants were expected to use common knowledge of politics to help solve this task. The participants were expected to look through the bubble map visualization as it provided the user with the top 50 most important words in the database as a whole. It was predicted that the participant would then select the 10 largest bubbles containing commonly-known politically-affiliated words such as “government”, “trump” and “president”. However, two participants used solely the bubble map visualization, one combined the values from the bubble map and ring network visualization and the rest simply queried “politics” and used the answers contained in the ring network visualization. Therefore, from the six participants, only one found all the expected keywords, one found 60% of the expected keywords, another found half, two found 3 out of 10 and one only had a single correct keyword leading to an average of 4.67 correct terms per participant. For this task, the participants in the treatment group needed an average of 79.5 seconds (SD = 58.49) as seen in Figure 5.11 and 8.93% (SD = 5.01) of the total times required for all tasks in the first task set.

5.1.2 Task Set 2

For the second task set, as there were no set correct answers, this task was designed with the goal of exploring how the participants used their respective systems with a semi-real-world use-case as a simulation.

For the control group, the participants used an average of 415 seconds (SD = 131.51) to complete the task which on average was around 17.70% of the total time they needed for the complete study containing both task sets. To complete the task, it was observed that the participants simply split the task into two steps. In the first step, after selecting a main topic based on their interests, they simply queried their main topic and selected three keywords they were interested in from the list the titles of related articles and registered those as the three subtopics. In the second step, they would query the union of their main topic and each of the subtopics and note down the keywords and its relevant sources from the list of articles.

For the treatment group, it should be noted that only the results for five out of the six participants in the group were evaluated. This is due to the fact that the screen recording software crashed for one of the participants who was in the process of completing the second task set and was unnoticed by both the participant and observer. From the five results, the participants needed an average time of 391.2 seconds (SD = 220.60) which was 31.25% (SD = 14.07) of the average combined times needed for task set 1 and 2 for the treatment group. There were three ways in which the participants attempted to solve this task. One method the participants used was a combination of the query system and the ring network visualization. With this method, the participants would search for the main topic using the query system and then find the subtopics and keywords using the ring network visualization. Fundamentally, the central node in the ring network visualization represented the main topic, the surrounding nodes in the inner ring represented the popular subtopics related to the main topic and the outer ring connected to the specified subtopics were then taken as popular keywords. The second method used by the participants were similar to the one used by the participants in the control group as they simply queried their main topic and instead of using visualizations, simply took keywords from the titles they were interested in and used those as subtopics and keywords. The last method was only used by one of the five participants. Here, the participant queried their main topic like in the other two methods. From the titles of the relevant articles, the participant

Work Load	Control	Treatment
Mental Demand	14.5	9
Physical Demand	4.33	4.17
Temporal Demand	11	7.33
Performance	10.17	5.33
Effort	14.33	8.17
Frustration	13.67	5.83

Table 5.1: Average unweighted NASA-TLX scores by the participants in the control and treatment groups.

then selected their subtopic and chose the article most related to their main and subtopics. After selecting the article, the participant then used the heat map view and used the three most popular and important word in context of the article as the keywords.

5.1.3 Surveys

After the study, the participants were asked to fill out two surveys: a NASA-TLX survey and a Likert scale-based questionnaire regarding the usability of the system. From the results seen in Table 5.1, the participants using the proposed system rated the mental demand of the study significantly lower than the participants in the control group. Furthermore, more the survey, both study groups believed that the physical demand of the study was low resulting in a mean score of around 4 for both groups respectively. In terms of temporal demand, participants believed the pacing of the tasks were relatively rushed in comparison to the participants in the treatment group. In addition, in terms of the success in accomplishing the tasks in the study, the control group members rated near the middle between perfect and failure whereas the treatment group gave an average score of 5.33 which is significantly closer to “perfect”. The survey also indicated that the participants in the control group thought that the effort level required by the study was high whereas it was rated between low and normal for the treatment group. Last of all, the TLX results regarding the frustration of the study showed that the participants in the control group thought the study was highly frustrating to do whereas the treatment group believed that they were rarely frustrated in the study.

For the control group, they were interviewed on the degree to which they agreed with the following three statements to evaluate the usability of the system used by the control group: 1) I liked the system in general; 2) The system in general was difficult to understand; and 3) The system in general was informative. For the first statement, one participant strongly disagreed, two disagreed, one believed it was neither and two agreed with the statement. For the second, two strongly disagreed, two disagreed, one neither agreed nor disagreed and the last agreed with the statement. For the last statement, two participants in the control group strongly disagreed with the statement, one neither agreed nor disagreed and half the participants agreed that the system was informative.

For the treatment group, the Likert-scale survey had the goal of evaluating the usability of every visualization component of the proposed system. As can be seen in Figure 5.12, the results are varied for each visualization. In terms of the likeability of the bubble map visualization, the participants were split between neutral and liking the visualization. By contrast, all the participants

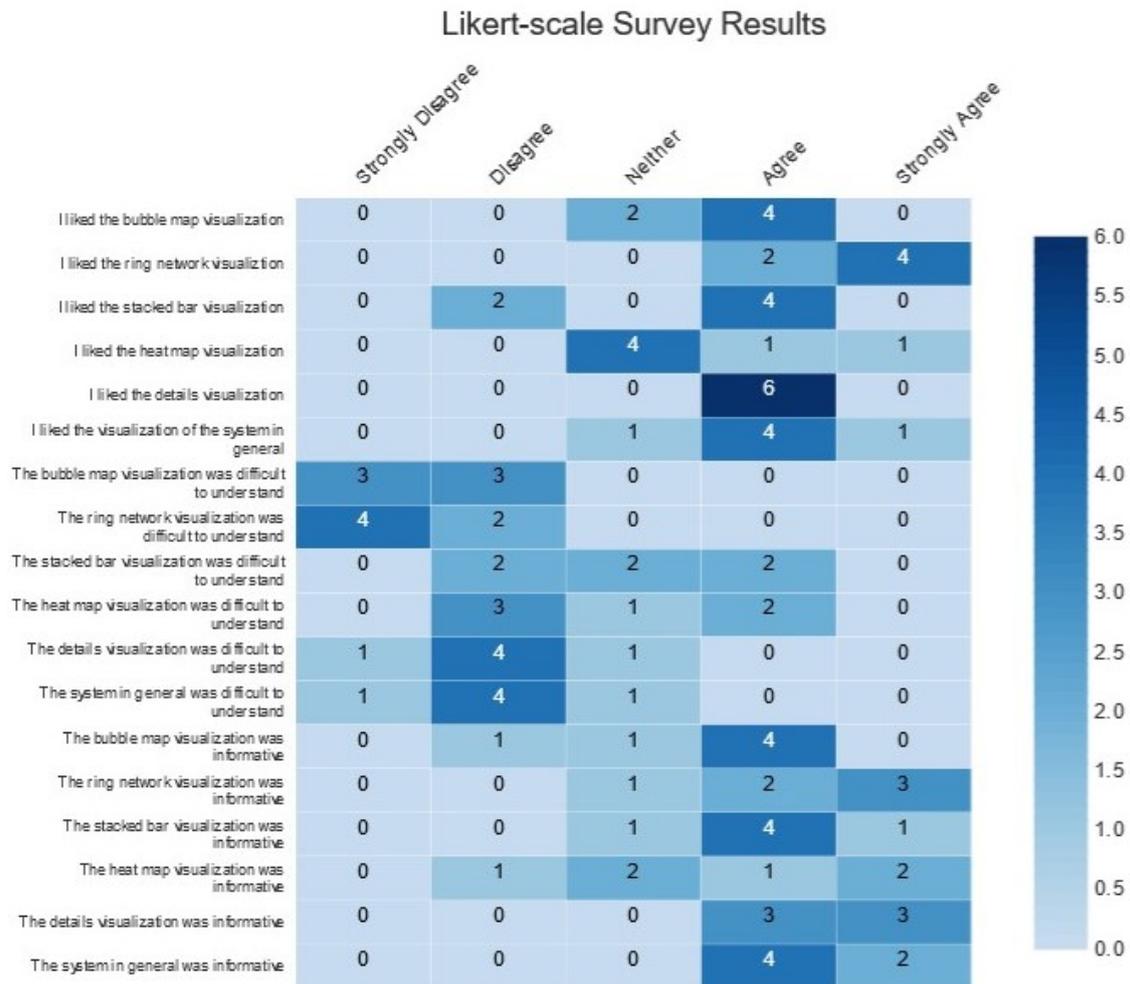


Figure 5.12: Results of the Likert-scale survey results done by the treatment group.

liked the ring network visualization. However, the participants in the treatment group were split when deciding whether or not they liked the stacked bar chart visualization. Most participants were neutral regarding the heat map visualization and everyone liked the details component which was a sub-component of the stacked bar chart visualization. Overall, the five out of six members of the treatment group liked the system. In terms of understandability, with the exception of the stacked bar chart and heat map, the participants had no problems understanding the content of the visualizations. Furthermore, with the exception of the heat map visualization, four or more out of six participants thought that every visualization was informative.

5.2 Discussion

The study had two primary goals: 1) to investigate the effectiveness of using software for the visual analysis of news trends; and 2) to determine the advantages and disadvantages of each component in the proposed system in the context of the visual analysis of news trends.

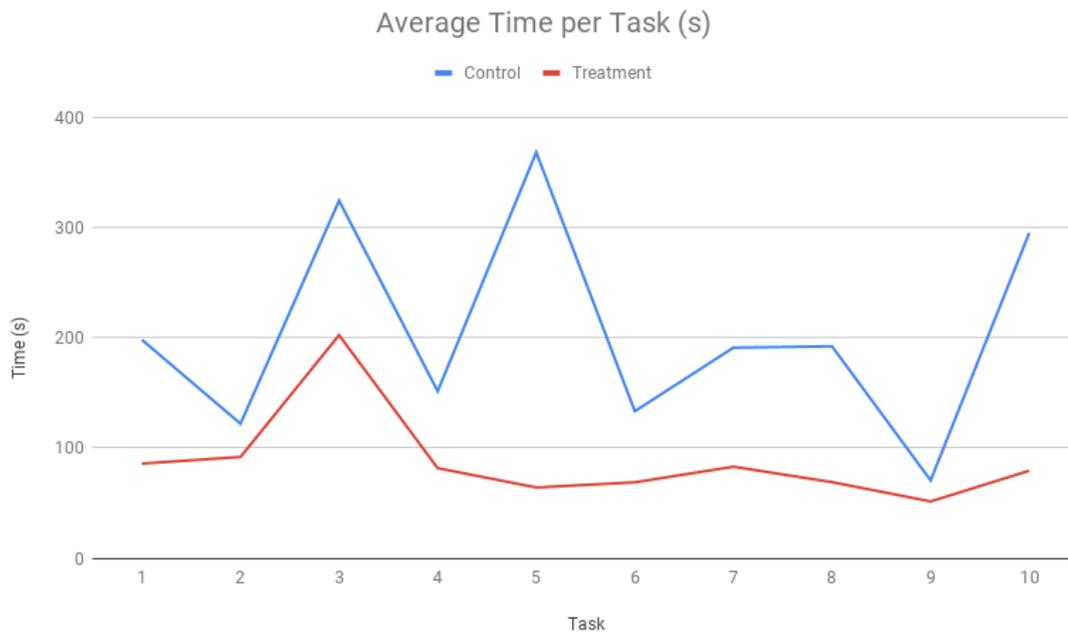


Figure 5.13: Average time each participant needed per task in seconds. In this visualization, the shades of blue represent data from every participant in the control group whereas the shades of red represent data from the members of the treatment group.

As can be seen in Figure 5.13, on average the participants using the proposed system completed every task in the first task set quicker than the participants in the control group. Despite the overall improvement in time, the trends of the time for a task in comparison to other tasks differ for both groups. For example, as seen in Figure 5.13 the control group needed on average more time for the first task than the second task. However, this does not hold true for the treatment group where the average time needed for the second task actually increased in comparison to the first task. This can be explained due to the relative lack of articles regarding “jon snow”. As mentioned before, there were only four articles regarding “jon snow” with all articles representing “jon snow” as the news presenter² and not the popular television character from Game of Thrones³. Due to the fact that the articles are sorted from the most to least relevant to the query and the titles clearly state the theme of the articles, the participants in the control group simply had to click the first four articles and compare the dates. On the other hand, for the treatment group, the process to complete the second task is nearly identical to the process needed to complete the first task which can explain why the times are so level. The same reasoning could be used to explain the mean time results for task nine. For the third task, both control and treatment groups saw a spike in average times needed to complete the task in comparison to the previous two tasks. For the control group, this can be explained as the task requires the participant to read the different articles and make their own inferences on the changes. The same can be said for the treatment group. For many of the other tasks, the proposed system simply outputs the expected answer. However for the third task, the

²<https://www.channel4.com/news/by/jon-snow>, last visited July 16, 2019.

³<https://www.hbo.com/game-of-thrones>, last visited July 16, 2019.

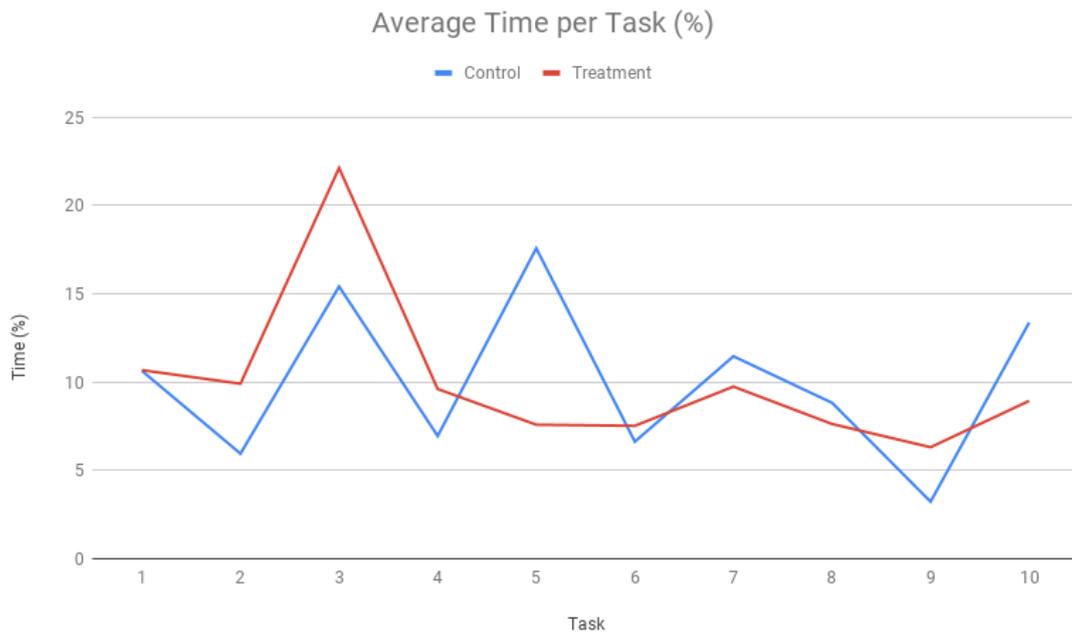


Figure 5.14: Percentage of time needed by each participant for every task in correlation to the times they needed to complete all tasks in the first task set combined. In this visualization, the shades of blue represent data from every participant in the control group whereas the shades of red represent data from the members of the treatment group.

participants in the treatment group still had to read the changes in the heat map visualization or changes in the top five most important words in each bar in the stacked bar chart and make base their opinion on that information. For the fifth task, the largest difference in average times between the control and treatment groups can be seen. Here, the control group needed the longest average time to complete over all tasks with a mean time of 368.33 seconds. In contrast, the time required by the treatment group to complete this task was the second shortest for them with an average time of 64.33 seconds. This could be explained by the vast difference in amount of work that needs to be done between the control and treatment group. For the control group, the participants actually needed to read the titles and articles to estimate the most related keywords to “nfl”. On the other hand, by searching for “nfl” in the proposed system, the system automatically generates a visualization containing the most related words to “nfl” allowing the treatment group participants to immediately get their answer. Additionally Figure 5.13 shows that for the treatment group, the average times for Task 7 were longer than those from Task 6. This could be due to the relatively low amount of tasks that required the use of the heat map and the high amount of tasks that use the stacked bar visualization. It was observed that many participants using the proposed system first tried to find the answer in the stacked bar chart before rediscovering the button that lead to the heat map containing the most important words for the articles related to the selected article.

In contrast, from the results of the percentage of time required for each task differ in comparison to the results of the mean time required as seen in Figure 5.14. From the rise and dips of each respective line, the rise and increase in the percentage of time to complete a task is relatively equal

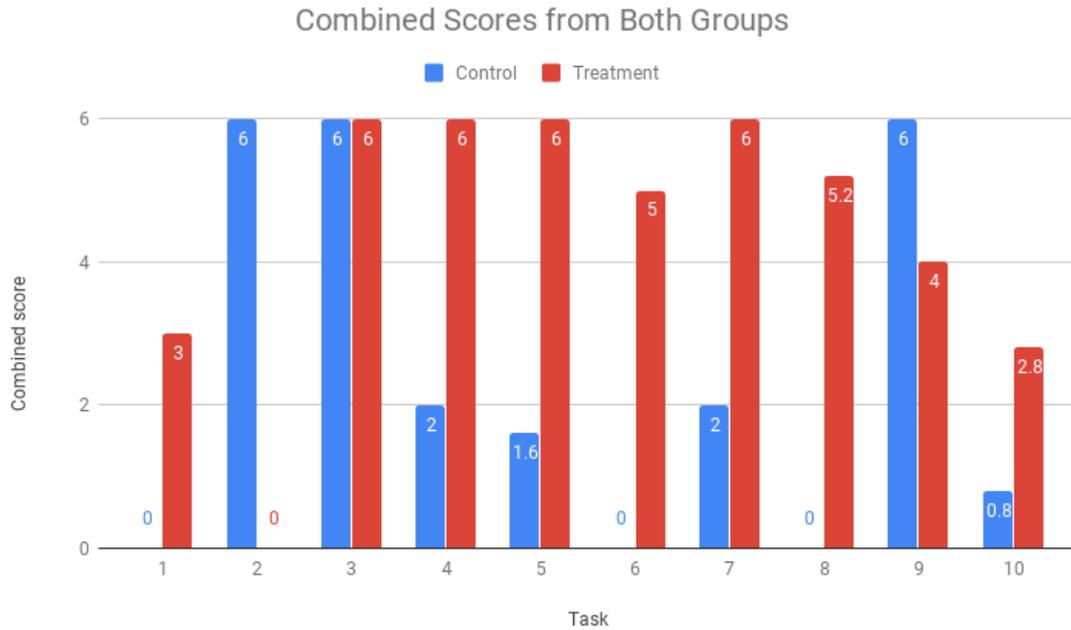


Figure 5.15: Combined scores for each task in the first task set each the control group (blue) and treatment group (red).

for both control and treatment groups can be seen. Furthermore, for the control group, a high percentage of time was needed for tasks where the participants needed to read the articles and make their own inferences as can be seen in the results for task 3, 5, 7, 8 and 10. Surprisingly, while Task 6 did require the control group participants to read, the percentage of time spent on the task was relatively low. This could be explained by the fact that the list of relevant articles were sorted by time. The majority of the participants simply read the titles of the first few articles related to the term “notre” and did not bother scrolling further down the list of articles to look for even more relevant ones or simply quickly browsing through the articles to look for frequent keywords. However, this is a problem of motivation and this is consistent with how the modern generation reads online documents today due to decreasing attention spans and online habits such as instant gratification when looking for specific information [Liu05]. To increase the incentive to do better and be more careful, competition and gamification could be a solution to this problem.

From Figure 5.15, it can be seen that the treatment group frequently found the correct or expected answers for the task more often than the control group. However, there were three exceptions where this did not hold true: Task 2; Task 3; and Task 9. As can be seen in Figure 5.15, in Task 2, the every participant in the control group found the correct answer whereas every participant using the proposed system found the incorrect answer. However, despite the incorrect answers, the treatment group used the proposed system as expected. Rather than using the system incorrectly, the results came from the fact that the participants were not careful and did not pay attention to the date format or used different filters. Here, the treatment group just used the monthly view of the stacked bar chart to determine their answer which by itself is correct. However, when clicking on the article itself, the participants simply assumed that “2019-04-01” meant January 4 instead of April 1. This was surprising as all the participants were German and were familiar with the Year-Month-Day

system used in Germany. Naturally they could have assumed that the date format used by the system was Year-Day-Month but the demonstration done by the observer at the beginning of study clearly stated that all dates used the Year-Month-Day system, even the temporal axes for both the stacked bar chart and heat map visualizations. For the third task, both the control and treatment group gave acceptable answers in the way that they were not incorrect. However, the control group answers took vastly longer to complete and at the same time were much less detailed than those from the control group. As for the ninth task, the participants in the treatment group again lost points because they were not paying attention. While the stacked bar chart contained all articles similar to the article that was most relevant to the query “Apex Legend”, not all articles were about “Apex Legends”. If the participants had checked the articles contained in each bar in the stacked bar chart, they would have noticed from the titles alone that only articles 40 and above contained articles that explicitly mentioned Apex Legends. However, two of the participants failed to verify the data and included relevant articles that did not discuss Apex Legends.

There were also instances where the treatment group did answer the tasks as correctly as expected as seen in Figure 5.15. In the first task, only 50% of the participants in the treatment group managed to find the answer correctly. However, from the observations, this was due to the fact that the participants who found in incorrect answers searched explicitly for “trump’s border policies” instead of searching for the most important keywords from the phrase such as “trump” and “border”. This is due to the fact that the words “policies” or “policy” rarely appear in articles discussing US-Mexico border due to President Trump’s policies. As for the participant who found the incorrect answer for the sixth task, it was again due to carelessness. Instead of selecting the bar containing the articles with a similarity value of 80 to 89, the participant mis-clicked and selected the bar containing the articles with a similarity value of 50 to 59. Furthermore, the participant did not notice that the similarity value was written at the top of the details view and highlighted in red. For Task 8, one of the participants only found 1 of the 10 possible correct keywords as they did not use the system as expected. Rather than generating the five most popular and relevant words to “hong kong”, the participant searched for “hong kong”, selected the most relevant article and used the heat map to predict the five most popular topics. However, this did not provide the correct answers as it only provided the most popular keywords for articles related to that article and not all articles relating to “hong kong” in general. The reasoning for Task 9 has already been explained above and for Task 10, it was again the case of the system not being used as expected. Only 2 out of 6 the participants used the bubble map visualization to answer the question. Of those two participants, one found all ten of the expected keywords, another found 6 out of 10 but also included keywords that were not highly relevant to politics such as “mr” and “new”. As for the other four participants, they attempted to find the answer through the ring network visualization by using the query “poltics”. However, the actual term “politics” is rarely mentioned when discussing political topics and rather use keywords of the political topics such as “government” or “president”.

While it was hypothesized that the treatment group would perform better than the control group, the control group also performed worse than expected. For example, in the first task, none of the participants managed to find the correct answer. This could be due to the fact that the earliest known article, while related to the keywords “trump” and “border”, did not contain as many instances of both “trump” and “border” as other articles. Due to this reason, the expected article was placed closer to the middle of the list rather than the top. Due to the fact that the majority of the participants in the control group only checked the most highly relevant articles in the list without scrolling further down, they did not reach any of the expected articles. The lack of incentive to scroll to less highly relevant articles also play a role in the reasons for the low scores for tasks 6, 7, 8 and

10. Furthermore, this is illustrated in the average temporal demand for the control group in the TLX survey. No time limit was set but total raw time of study for control group including answer explanations required an average time of 2,233.17 seconds ($SD = 608.58$) whereas the raw time for the treatment group required an average time of 1,379.5 seconds ($SD = 487.77$) despite the fact that the participants had to wait between 12 and 18 seconds every time they used the stack bar chart visualization due to the processing times. The due to the already high amount of times the participant in the control group needed to spend on completing the tasks, they saw no benefit in spending even longer to read through the hundreds or thousands of relevant articles contained in the list. Additionally, for Task 8, the results indicated that many participants in the control group only named specific familiar keywords that were in the titles and articles such as “trade” and “china”. However, half the expected results were the specific names of people holding important positions of power in the Hong Kong and Chinese government. The variation between the results obtained from the control group and the expected results could be due to the fact that names, and especially foreign and unfamiliar names, are less memorable than other types of personal information such as their position in the government as they are typically meaningless on their own [BB93].

5.3 Summary

From the results of the study, it was demonstrated that the participants using the proposed system performed the tasks faster and more accurately than the participants using a generic search engine-based system. The results of the processed times showed that the treatment group managed to complete the set of tasks on average more than twice as fast as the control group. Furthermore, the treatment group scored nearly double the amount of points as the control group. In addition, the treatment group lost nearly 12% of their total points mainly because they were careless when giving their answer and did not notice the attention check built into the system. On the other hand, the control group times were slower because the participants had to analyze the raw data themselves and it was determined that their performance was also lower due to the amount of reading that needed to be done and memorized which directly influenced motivation due to the temporal demand of manually analyzing hundreds or thousands of articles.

6 Conclusion

In this thesis a system using fundamental visualizations to help analysts analyze the evolution and track the changes of news trends visually was proposed. During the course of the thesis, different approaches for assessing and evaluating the similarity of words and documents as well as different visualizations that could prove to be expressive and effective when performing visual analysis on web content and online news articles in particular were researched and tested. With this information, a system was developed using TF-IDF and cosine similarity for the topic modeling and Doc2Vec with cosine similarity for document modeling. Furthermore, five visualizations were chosen to represent different aspects of the information from the data set of articles. A study was then conducted with the population being split between a control group which used a generic search engine-based system and a treatment group using the proposed system. For the user study, two sets of tasks were developed, the first set of tasks consisted of possible real-world tasks that a journalist or social media influencer would possibly need to do and contained tasks which had correct and incorrect answers. The second task set had the goal of simulating real-world usage where the participant uses their respective systems under the pretense that they are playing a specific role. As this task is designed to determine how the participants themselves use the system to find the answers they need, there were no correct or incorrect answers. The comparison of the results between the control and treatment group in the study indicate that using the proposed system would be beneficial for the visual analysis of new trends and that further development of this field would be promising. Additionally, the results from the study indicated that the participants using the proposed system worked more than twice as fast as the participants using the generic system. Furthermore, the treatment group had nearly double the points in comparison to the control group indicating that they could accomplish the task more correctly or accurately. While there are many areas of improvement such as better natural language processing to remove un-descriptive words such as “said” and “say”, positive feedback was received from the participants. The participants enjoyed the fundamental nature of the visualizations as it did not require any training to understand. Furthermore, they found visualizations like the ring network visualizations very aesthetically pleasing and informative. Additionally, the treatment group enjoyed being able to understand the most important topics of an article without having to read the article itself which is a definite time-saver in a world where people expect the information they want quickly due to the widespread use of the internet. While the current treatment group is still too small and localized to provide definite proof that the use of the proposed system is an effective and efficient tool for the visual analysis of news trends, this study could be used as a stepping stone towards finding said proof with larger and less localized treatment groups as well as a more refined system and process which could further enhance the information visualization and visual analysis aspect of the software.

Future Work

While the system has proven to be effective with the tasks and population used in the study and has been positively received, there are still many elements of the system that could be improved. One of the most requested improvements based on comments from the study was improved performance. Due to the large amount of information that needs to be pre-processed and visualized, the performance of the stacked bar visualization is not optimal. However, this could be partially solved using an online database which saves the preprocessing for every article so it only ever has to be done once. Furthermore, an upgrade to a streaming-based system receiving documents in real-time would be immensely helpful since the system would then be able to process live information. As mentioned in Chapter 3, this could be done via sacrificing a small amount of accuracy for a massive improvement in training time by using algorithms such as TF-IDF over Doc2Vec or use a cloud-based system to massively improve the machine-learning component. Additionally, testing other fundamental visualizations to visualize different aspects of the visual analysis of news articles could be helpful in determining if there are more optimal visualizations which stay intuitive for the analyst at the same time. Furthermore, the fundamental nature of the visualizations back-fired to a certain extent as the participants became negligent and did not double-check their answers leading to failures when it came to the attention checks. In addition, while the answers regarding the evolution of topics was more detailed for those in the treatment group, it is still not fully automated and did require a certain amount of reading and comparisons by the participants using the proposed system. However, these challenges can possibly be solved through the use of a stricter visual design guideline such as using more eye-catching visual elements or more consistent shapes, a system to automatically detect inconsistencies in the data or using different types of interaction design patterns. Last of all, long-term studies with a larger and more varied or even expert population could prove useful in further determining the effectiveness of visual analysis on web content dissemination.

Acknowledgements

I would like to offer special thanks to Dr. Dennis Thom and Johannes Knittel for their introduction to the topic as well as their assistance, advice and encouragement during the preparation and implementation of the system. Last but not least, I would like to express my great appreciation for the participants who took part in the user study.

Bibliography

- [] *word2vec*. URL: <https://code.google.com/archive/p/word2vec/> (cit. on pp. 13, 25).
- [Ali17] R. Ali Alshaymi. “Provenance Detection of Online News Article”. In: *Journal of Information Technology Software Engineering* 07 (Jan. 2017). doi: [10.4172/2165-7866.1000204](https://doi.org/10.4172/2165-7866.1000204) (cit. on pp. 13, 17).
- [AS07] I. E. Allen, C. A. Seaman. “Likert scales and data analyses”. In: *Quality progress* 40.7 (2007), pp. 64–65 (cit. on p. 60).
- [BB93] A. M. Burton, V. Bruce. “Naming faces and naming names: Exploring an interactive activation model of person recognition”. In: *Memory* 1.4 (1993), pp. 457–480. doi: [10.1080/09658219308258248](https://doi.org/10.1080/09658219308258248) (cit. on p. 81).
- [BJGK14] M. Bora, D. Jyoti, D. Gupta, A. Kumar. “Effect of different distance measures on the performance of K-means algorithm: an experimental study in Matlab”. In: *arXiv preprint arXiv:1405.7471* (2014) (cit. on p. 29).
- [BKP14] L. Barth, S. G. Kobourov, S. Pupyrev. “Experimental comparison of semantic word clouds”. In: *International Symposium on Experimental Algorithms*. Springer. 2014, pp. 247–258. doi: [10.1007/978-3-319-07959-2_21](https://doi.org/10.1007/978-3-319-07959-2_21) (cit. on p. 35).
- [BNJ02] D. M. Blei, A. Y. Ng, M. I. Jordan. “Latent dirichlet allocation”. In: *Advances in neural information processing systems*. 2002, pp. 601–608 (cit. on p. 29).
- [BNJ03] D. M. Blei, A. Y. Ng, M. I. Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022 (cit. on p. 29).
- [BVF18] D. den Brave, M. van de Velden, F. Frasinca. “Investigating the Performance of Different Feature Representations in Text Classification within Machine Learning”. In: (2018) (cit. on p. 37).
- [Car99] M. Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (cit. on pp. 40, 41, 43).
- [Cha06] W. W.-Y. Chan. “A survey on multivariate data visualization”. In: *Department of Computer Science and Engineering. Hong Kong University of Science and Technology* 8.6 (2006), pp. 1–29 (cit. on p. 33).
- [CWS+13] M. Chen, K. Q. Weinberger, F. Sha, et al. “An alternative text representation to TF-IDF and Bag-of-Words”. In: *arXiv preprint arXiv:1301.6770* (2013) (cit. on p. 37).
- [DOL15] A. M. Dai, C. Olah, Q. V. Le. “Document embedding with paragraph vectors”. In: *arXiv preprint arXiv:1507.07998* (2015) (cit. on pp. 23, 38).
- [Few06] S. Few. “Multivariate analysis using parallel coordinates”. In: *Perceptual edge* (2006), pp. 1–9 (cit. on p. 33).

- [GL14] Y. Goldberg, O. Levy. “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method”. In: *arXiv preprint arXiv:1402.3722* (2014) (cit. on p. 25).
- [HLL14] F. Heimerl, S. Lohmann, S. Lange, T. Ertl. “Word cloud explorer: Text analytics based on word clouds”. In: *2014 47th Hawaii International Conference on System Sciences*. IEEE. 2014, pp. 1833–1842. doi: [10.1109/hicss.2014.231](https://doi.org/10.1109/hicss.2014.231) (cit. on p. 34).
- [HS88] S. G. Hart, L. E. Staveland. “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research”. In: *Advances in psychology*. Vol. 52. Elsevier, 1988, pp. 139–183. doi: [10.1016/s0166-4115\(08\)62386-9](https://doi.org/10.1016/s0166-4115(08)62386-9) (cit. on p. 60).
- [Hua08] A. Huang. “Similarity measures for text document clustering”. In: *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. Vol. 4. 2008, pp. 9–56 (cit. on pp. 28, 36).
- [KEC06] R. Keller, C. M. Eckert, P. J. Clarkson. “Matrices or node-link diagrams: which visual representation is better for visualising connectivity models?” In: *Information Visualization 5.1* (2006), pp. 62–76. doi: [10.1057/palgrave.ivs.9500116](https://doi.org/10.1057/palgrave.ivs.9500116) (cit. on p. 32).
- [KK11] M. Khan, S. S. Khan. “Data and information visualization methods, and interactive mechanisms: A survey”. In: *International Journal of Computer Applications* 34.1 (2011), pp. 1–14 (cit. on pp. 30, 33, 34).
- [KNMK13] M. Krstajić, M. Najm-Araghi, F. Mansmann, D. A. Keim. “Story Tracker: Incremental visual text analytics of news story development”. In: *Information Visualization 12.3-4* (2013), pp. 308–323. doi: [10.1177/1473871613493996](https://doi.org/10.1177/1473871613493996) (cit. on p. 20).
- [KPA08] A. Kosmopoulos, G. Paliouras, I. Androutopoulos. “Adaptive spam filtering using only naive bayes text classifiers”. In: *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)*. 2008, pp. 1–2 (cit. on p. 37).
- [LBK09] J. Leskovec, L. Backstrom, J. Kleinberg. “Meme-tracking and the dynamics of the news cycle”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 497–506. doi: [10.1145/1557019.1557077](https://doi.org/10.1145/1557019.1557077) (cit. on pp. 16, 20).
- [LCWL14] S. Liu, W. Cui, Y. Wu, M. Liu. “A survey on information visualization: recent advances and challenges”. In: *The Visual Computer* 30.12 (2014), pp. 1373–1393. doi: [10.1007/s00371-013-0892-3](https://doi.org/10.1007/s00371-013-0892-3) (cit. on p. 30).
- [Liu05] Z. Liu. “Reading behavior in the digital environment: Changes in reading behavior over the past ten years”. In: *Journal of documentation* 61.6 (2005), pp. 700–712. doi: [10.1108/00220410510632040](https://doi.org/10.1108/00220410510632040) (cit. on p. 79).
- [LM14] Q. Le, T. Mikolov. “Distributed representations of sentences and documents”. In: *International conference on machine learning*. 2014, pp. 1188–1196 (cit. on pp. 13, 21, 26, 27, 29, 37, 38).
- [LWW+13] S. Liu, Y. Wu, E. Wei, M. Liu, Y. Liu. “Storyflow: Tracking the evolution of stories”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2436–2445. doi: [10.1109/tvcg.2013.196](https://doi.org/10.1109/tvcg.2013.196) (cit. on pp. 18–20).

- [LYW+15] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, J. Pei. “Online visual analytics of text streams”. In: *IEEE transactions on visualization and computer graphics* 22.11 (2015), pp. 2451–2466. doi: [10.1109/tvcg.2015.2509990](https://doi.org/10.1109/tvcg.2015.2509990) (cit. on pp. 13, 16, 18, 20).
- [LZZ15] J. Lilleberg, Y. Zhu, Y. Zhang. “Support vector machines and word2vec for text classification with semantic features”. In: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE. 2015, pp. 136–140. doi: [10.1109/icci-cc.2015.7259377](https://doi.org/10.1109/icci-cc.2015.7259377) (cit. on p. 36).
- [Mac86] J. Mackinlay. “Automating the design of graphical presentations of relational information”. In: *Acm Transactions On Graphics (Tog)* 5.2 (1986), pp. 110–141. doi: [10.1145/22949.22950](https://doi.org/10.1145/22949.22950) (cit. on pp. 39, 41, 54).
- [MDP+11] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts. “Learning word vectors for sentiment analysis”. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics. 2011, pp. 142–150 (cit. on p. 36).
- [MLS13] T. Mikolov, Q. V. Le, I. Sutskever. “Exploiting similarities among languages for machine translation”. In: *arXiv preprint arXiv:1309.4168* (2013) (cit. on pp. 25, 26).
- [MRS10] C. Manning, P. Raghavan, H. Schütze. “Introduction to information retrieval”. In: *Natural Language Engineering* 16.1 (2010), pp. 100–103 (cit. on p. 37).
- [PL04] B. Pang, L. Lee. “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts”. In: *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2004, p. 271. doi: [10.3115/1218955.1218990](https://doi.org/10.3115/1218955.1218990) (cit. on p. 37).
- [Ram+03] J. Ramos et al. “Using tf-idf to determine word relevance in document queries”. In: *Proceedings of the first instructional conference on machine learning*. Vol. 242. Piscataway, NJ. 2003, pp. 133–142 (cit. on p. 24).
- [RKA12] F. Rahutomo, T. Kitasuka, M. Aritsugi. “Semantic Cosine Similarity”. In: *The 7th International Student Conference on Advanced Science and Technology ICAST*. 2012 (cit. on p. 28).
- [RMC91] G. G. Robertson, J. D. Mackinlay, S. K. Card. “Cone trees: Animated 3d visualizations of hierarchical information.” In: *CHI*. Vol. 91. 1991, pp. 189–194. doi: [10.1145/108844.108883](https://doi.org/10.1145/108844.108883) (cit. on p. 32).
- [RTB96] B. E. Rogowitz, L. A. Treinish, S. Bryson. “How not to lie with visualization”. In: *Computers in Physics* 10.3 (1996), pp. 268–273. doi: [10.1063/1.4822401](https://doi.org/10.1063/1.4822401) (cit. on p. 34).
- [SK96] P. Sollich, A. Krogh. “Learning with ensembles: How overfitting can be useful”. In: *Advances in neural information processing systems*. 1996, pp. 190–196 (cit. on p. 38).
- [SR17] J. H. Shen, F. Rudzicz. “Detecting anxiety through reddit”. In: *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality*. 2017, pp. 58–65. doi: [10.18653/v1/w17-3107](https://doi.org/10.18653/v1/w17-3107) (cit. on pp. 35, 38).

- [TWY+14] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin. “Learning sentiment-specific word embedding for twitter sentiment classification”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 1555–1565. DOI: [10.3115/v1/p14-1146](https://doi.org/10.3115/v1/p14-1146) (cit. on p. 16).
- [War12] C. Ware. *Information visualization: perception for design*. Elsevier, 2012 (cit. on p. 13).
- [WCS+18] Y. Wu, Z. Chen, G. Sun, X. Xie, N. Cao, S. Liu, W. Cui. “StreamExplorer: A multi-stage system for visually exploring events in social streams”. In: *IEEE transactions on visualization and computer graphics* 24.10 (2018), pp. 2758–2772. DOI: [10.1109/tvcg.2017.2764459](https://doi.org/10.1109/tvcg.2017.2764459) (cit. on pp. 19, 20).
- [WF09] L. Wilkinson, M. Friendly. “The history of the cluster heat map”. In: *The American Statistician* 63.2 (2009), pp. 179–184. DOI: [10.1198/tas.2009.0033](https://doi.org/10.1198/tas.2009.0033) (cit. on p. 31).
- [WLC+16] X. Wang, S. Liu, Y. Chen, T.-Q. Peng, J. Su, J. Yang, B. Guo. “How ideas flow across multiple social groups”. In: *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. 2016, pp. 51–60. DOI: [10.1109/vast.2016.7883511](https://doi.org/10.1109/vast.2016.7883511) (cit. on pp. 15, 20).
- [YKS+07] J. S. Yi, Y. ah Kang, J. T. Stasko, J. A. Jacko, et al. “Toward a deeper understanding of the role of interaction in information visualization”. In: *IEEE Transactions on Visualization & Computer Graphics* 6 (2007). DOI: [10.1109/tvcg.2007.70515](https://doi.org/10.1109/tvcg.2007.70515) (cit. on p. 30).
- [ZJZ10] Y. Zhang, R. Jin, Z.-H. Zhou. “Understanding bag-of-words model: a statistical framework”. In: *International Journal of Machine Learning and Cybernetics* 1.1-4 (2010), pp. 43–52. DOI: [10.1007/s13042-010-0001-0](https://doi.org/10.1007/s13042-010-0001-0) (cit. on p. 24).
- [ZLW+16] Y. Zhong, S. Liu, X. Wang, J. Xiao, Y. Song. “Tracking idea flows between social groups”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016 (cit. on p. 16).
- [ZZM16] R. Zhao, A. Zhou, K. Mao. “Automatic detection of cyberbullying on social networks based on bullying features”. In: *Proceedings of the 17th international conference on distributed computing and networking*. ACM. 2016, p. 43. DOI: [10.1145/2833312.2849567](https://doi.org/10.1145/2833312.2849567) (cit. on p. 35).

All links were last followed on July 16, 2019.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature