ABSTRACT

COMPUTER & INFORMATION SCIENCES

ELTAYEBY, OMAR B.SC. ALEXANDRIA UNIVERSITY, 2011

MEASURING THE INFLUENCE OF MAINSTREAM MEDIA ON TWITTER USERS

Committee Chair: Péter Molnár, Ph.D.

Thesis dated: May 2014

This research study is based on analyzing text on Twitter to quantify the correlation between the segregated opinions posted by the users on multiple issues and the news mention in between those opinions. The results show various correlation percentages within the range of sentiment used by the clusters of segregated opinions towards different topics. The study participates in fitting the model of mining the opinions into using multiple algorithms, such as Apriori for finding the trending topics, Hierarchical clustering to describe the semantic relatedness between adjectives and the Expectation-Maximization algorithm to mine the hidden variables which result into different clusters. The framework shows how those algorithms are applied on the dataset collected from Twitter. Multiple experiments are conducted with different filtering categories to extract tweets with certain properties suitable for the analysis.

MEASURING THE INFLUENCE OF MAINSTREAM MEDIA ON TWITTER USERS

A DISSERTATION

SUBMITTED TO THE FACULTY OF CLARK ATLANTA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE MASTERS DEGREE

BY OMAR ELTAYEBY

DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

ATLANTA, GEORGIA

MAY 2014

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided me the possibility and help to complete this thesis. A special gratitude I give to Professors Péter Molnár and Roy George, who provided guidance and directions through the course of the project and thesis writing. I thank them for their efforts and time, which they dedicated for the success of this project. Additionally, I would like to express my appreciation to all Professors who taught me during my Masters here at Clark Atlanta University, Atlanta, GA and my undergraduate studies at Alexandria University, Egypt.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CI	Competitive Intelligence
DJIA	Dow Jones Industrial Average
dnc	Democratic National Committee
EM	Expectation-Maximization
GPL	General Public License
GPOMS	Google-Profile of Mood States
GUI	Graphical User Interface
IC	Information Content
KDD	Knowledge Discovery & Data Mining
LBP	Loopy Believe Propagation
LCS	Least Common Sub-summer
LDA	Latent Semantic Allocation
MAPE	Mean Average Percentage Error
MRF	Markov Random Field
NLP	Natural Language Processing
nyc	New York City
OWS	Occupy Wall Street
p2	Progressive Propaganda
POS	Part of Speech
SWOT	Strengths, Weaknesses, Opportunities and Threats
tcot	Top Conservatives On Twitter

tfb	Team Follow Back
TF-IDF	Term Frequency-Inverse Document Frequency
tiot	Top Independent On Twitter
WOE	Wikipedia-based Open Extractor
WWW	World Wide Web

CHAPTER 1

INTRODUCTION

This thesis is based on a research project that has been conducted at Clark Atlanta University (CAU) under the supervision of Professors Roy George and Peter Molnar. The aim of the project is to construct a framework for measuring the influence of mass media on Twitter users. Media influence or media effects are used in media studies, psychology, communication theory and sociology to refer to the theories about the ways in which mass media and media culture affect how their audiences think and behave. Arguably, the agenda-setting process is an unavoidable part of news gathering by the large organizations which make up much of the mass media. For example, four main news agencies — AP, UPI, Reuters and Agence-France-Presse — together provide 90% of the total news output of the world's press, radio and television. According to Stuart Hall, because some of the media produce material which often is impartial and serious, they are accorded a high degree of respect and authority. Stuart says, "independence is not a mere cover, it is central to the way power and ideology are mediated in societies like ours" (Stuart Hall, 1973). In 1972, McCombs and Shaw demonstrate the agenda-setting effect at work in a study conducted in Chapel Hill, North Carolina, USA during the 1968 presidential elections. A representative sample of un-decided voters was asked to outline the key issues of the election as it perceived them. Concurrently, the mass media serving

these subjects were collected and their content was analyzed. The results showed a definite correlation between the two accounts of predominant issues. The purpose, of this current study on the application level, shows the same correlation, but between the mass media and the people's opinion through twitter.

On the development level, the basic concept of finding this correlation derives the methodology for our analyzing the sentiment used on Twitter. A comparison between the sentiment used when mentioning and not mentioning news sources on Twitter towards trending topics is shown to infer the how much the mass media is influential. In Computer Science, sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. A basic task in sentiment analysis (Michelle de Haaf, 2010) is classifying the polarity of a given text at the document, sentence, or feature/aspect level, whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry," "sad," and "happy" (Linhao Zhang, 2013). Many research works were done in the field of aspect-based opinion mining on scientific documents, web content generally and social media for multiple purposes such as stock market sentiment analysis, opinion mining about product features, spam review detection etc.... The aspect-based opinion mining task in this

project is accomplished first by extracting the topics which is mostly concerned by the twitter users then finding the semantic relatedness between the corresponding words used to describe those topics. Concretely, semantic relatedness can be estimated for instance by defining a topological similarity, by using ontologies to define a distance between terms. The ontology of terms could be defined by several text corpuses, which we used in our project by importing them using the Natural Language Processing Toolkit (NLTK). As an example, a naive metric for the comparison of concepts ordered in a partially ordered set and represented as nodes of a directed acyclic graph (taxonomy), would be the minimal distance in terms of edges composing the shortest-path linking the two concept nodes. Based on text analyses, semantic distance between units of language can also be estimated using statistical means such as a vector space model to correlate words and textual contexts from a suitable text corpus (co-occurrence).

The remainder of this thesis is organized in the following manner. Chapter 2 is the literature review to show previous related work from other papers and projects in the field of text mining, association rules mining, sentiment analysis and opinion clustering. This chapter will not handle the steps or the work done in the project, it will just cover a broad perspective of different applications and work done in those areas. Such exposure to other work enhances the readers awareness about the contribution of this thesis to the various fields mentioned. Chapter 3 shows the framework in details and the previous analysis done before constructing the framework. This analysis discusses some of the primary results that are the outcome of the initial framework. This chapter includes the

methodology of collecting the data from twitter on the fedora cluster of CAU, some statistical analysis that shows the general trend of tweets' types that users post and the framework of the aspect-based opinion mining process. The chapter handles the details of the steps specified in the framework with an explanation of how we fit the algorithms used into our model. Through chapter 4 inter-step results are shown with the visuals that show the meaning and inferences about the results by discussing those visuals and how they fulfill the aim of this project on both the application and computer science levels. Finally, chapter 5 concludes the research thesis through explaining the best practices for developing this framework and the disadvantages that were encountered out of these results.

CHAPTER 2

LITERATURE REVIEW

Introduction to Text Mining:

Nowadays evolving technologies provide enormous amount of textual data and it is growing at staggering rate. The best example of this growth is the World Wide Web (WWW), which is estimated to provide access to Exabytes of text from blogs and social media to regular websites and electronic markets. Data scientists took the challenge to efficiently mine interesting patterns, trends and potential information that are of interest to the user and that could derive insights for solving real-life problems (Ricardo Baeza-Yates et al., 2002). Text mining, also known as Knowledge-Discovery in Text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text, which could be located in databases or files. In general, data mining deals with structured data (for example relational databases), where data is organized in a way that multiple tables are connected to each other through common fields, while text presents special characteristics and is unstructured. As languages are used for many types of information exchange, it creates a dilemma for data scientists to organize them as structured relational databases. The unstructured data is totally different from databases, where mining techniques are usually applied and structured data is managed, even when using them in the same context (Vishal Gupta and Gurpreet S. Lehal, 2009). Text mining could be used for unstructured or semi-structured

data sets such as emails, full-text documents and HTML files and more (Delgado et al., 2002). An example of semi-structured data is an email with an appointment details, holding information about the location, time and date. This type of information formats are easier to analyze for whatever purposes due to the organized and predefined data types used in such cases. Text mining shares many characteristics with classical data mining, but differs in many ways (Ah-hwee Tan, 1999). However, many algorithms used for discovering knowledge in relational databases are ill-suited for the textual applications. Thus, text mining methodologies start with the usage of Natural Language Processing (NLP) techniques to organize the text before being further processed.

Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining (Nasukawa and Nagano, 2001). Starting with a collection of documents, a text mining process would retrieve a particular document and preprocess it by checking format and character sets. Then it would go through a text analysis phase, sometimes repeating techniques until information is extracted (Vishal Gupta and Gurpreet S. Lehal, 2009).

Text preprocessing classically means tokenization and then Part of Speech Tagging, as we will see in the methodology chapter how we used the NTLK, or in a bag of words approach word stemming and the application of a stop word list. Tokenization is the process of splitting the text into words or terms. Part of Speech (PoS) Tagging tags words according to the grammatical context of the word in the sentence, hence dividing

up the words into nouns, verbs and more. This is important for the exact analysis of the sentence structure, as it is needed in the extraction of relations between the texts. Most text mining objectives fall under the following categories of operations: Search and Retrieval, categorization (supervised classification), summarization, Trends Analysis, Associations Analysis, Visualization and more, where each objective suite the special task that is to be applied.

The following subsections explain different techniques for information extraction from text documents generally, while focusing on previous work related to our research project in the area of Association Rule Mining, Temporal Association Rules Mining and Prototypical Documents Mining.

Information Extraction:

Information extraction systems can be used to directly extricate abstract knowledge from a text corpus, or to extract structured data from a set of documents which can then be further analyzed with traditional data mining techniques to discover more general patterns and insights that suite the application (Raymond Mooney and Razvan Bunescu, 2005). Information extraction is the task of locating desired pieces of data. Many text mining methods have been developed in order to achieve the goal of retrieving useful information for application users or web surfers, for example the paper published by Edda and Jorg and Sebastiani in 2002. Based on this hypothesis, Lewis in 1992, conducted several experiments using phrasal indexing language on a text categorization task. The results showed that the phrase-based indexing language was not

superior to the word-based one. Although phrases carry less ambiguous and more brief meanings than individual words, the likely reasons for the discouraging performance from the use of phrases have inferior statistical properties to words and low frequency of occurrence.

Related work in Association Rule mining:

Association rule mining is a powerful data analysis technique which brings basic attributes and summaries of documents into consideration and appears frequently in data mining literature (Pack Chung et al., 1999). Association rules aim to extract interesting correlations, frequent patterns, associations or causal structures among sets of items in the transaction databases or other repositories. First, all frequent itemsets of different sizes are found by alternating between two main steps; minimum support count filter is applied to find the frequent itemsets and the next larger size of candidate itemsets is filtered by the pruning process to find the next larger frequent itemsets. Then all frequent itemsets and the minimum confidence constraint is used to form rules. The main advantages of association rules are simplicity, intuitiveness and freedom from model-based assumptions. They are widely used in many other areas such as telecommunication networks, market and risk management, inventory control and more (Qiankun Zhao et al., 2013).

In recent times, extracting semantic relationships among entities in specific collections of text documents has gained enormous popularity, which leads to the motivation of our research, that is to apply association rule mining to text databases to

capture the relationships among words (terms). Association rules have been researched and applied extensively, in diverse domains and applications (Bench-Capon et al., 2000). However, it should be mentioned that the association rule extraction is of exponential growth and a very large number of rules can be produced. The extracted association rules identify the relations between features in the documents collection. The scattering of features in text contribute to the complexity of define features to be extracted from text. These kinds of features relationships can be better described with the association rule mining of text. Several researchers have presented algorithms and approaches for mining associations from text document collections, for example Hany Mahgoub in 2006.

Part-of-Speech tagging:

The process of Part-of-Speech (POS) tagging is preceded by splitting the sentence into separate words, which is "Tokenization." Afterwards each word is tagged with Morpho-syntactic category (noun, verb, adjective etc...) that fits the word in the sentence. The process allows filtering out non-significant words, or enables the application to capture the concerned category. In our experiments, we used a rule-based tagger described by Eric Brill (Brill, 1992) in his PhD thesis that is implemented in the Natural Language Processing Toolkit functions.

Social Media Background:

Everywhere on the internet it is observed that there has been an extensive use of social media over the last few years. Online communities and blogs were developed to focus and assist users on both personal and professional life issues. Nowadays the

internet provides access to more than 900 social media sites and blogs. On these social media sites there are groups which focus on every potential area of interest on some of the most popular social sites like Facebook and Google Plus for social connections, personal networking and sharing posts, Twitter for following up news, opinions and short messages, LinkedIn for searching jobs and potential candidates to hire, YouTube for video sharing and uploading, Pinterest for sharing online products and wish lists, and Sound Cloud for sharing music. To help in understanding and realizing the extensive use of those social media sites, the following statistics were compiled in November 2013 by Jonathan Bernstein:[1] There are 751 million users on Facebook from mobile with 7,000 different devices, which gives a hint about how those sites give the makes the platforms available on different kinds of devices; there are over 288 million monthly active users on Twitter; the total number of LinkedIn groups is 1.5 million; there are 751 million users on Facebook from mobile with 7,000 different devices; 77% of internet users read blogs.

The majority of the population is using social media in some form or another. Given the substantial increase in the use of social media, there is a significant amount of information that is being generated. As seen from the same cited sources, the volume of this content is staggering: 350 million Photos are uploaded every day; There are over 1 billion unique monthly visitors on YouTube; On an average, over 400 million tweets are being sent per day; Over 3 million LinkedIn company pages.

---

[1] Jonathan Bernstein. Social Media Today. http://socialmediatoday.com (accessed February 15, 2014). A lot has happened in the fast-moving world of social media already this year.

Additionally there are many companies that are spending their time and money to engage on social media and create a significant amount of content for propaganda and advertisement purposes. However, they also aim for users' feedback. The outcome of such time spent on social media and the information being generated, businesses have taken notice and are attempting to leverage the power of social media to help them succeed.The following statistics compiles and proves such statement: Two-thirds of comScore's U.S. Top 100 websites and half of comScore's Global Top 100 websites have integrated with Facebook; Many businesses now have established Twitter accounts in an attempt to connect with current and potential customers; Eighty-eight percent of companies use LinkedIn as a recruitment tool; Corporate blogging accounts for 14% of blogs.

The commitment that businesses are making to increase their presence in social media is also being shown in the resources they are committing to this effort. According to eMarketer, U.S. advertisers increased the digital ad spending. As digital matures, and continues siphoning dollars from traditional media, the options within digital advertising are also proliferating. Breaking down where advertisers expected to make the biggest web increases, social media advertising ranked first, with 47% of respondents expecting to up investments in the next year.[2] According to Banking2020.com, 50% of Chief Marketing Officers at Fortune 1000 companies say they have launched a corporate blog because it is a cost of doing business today. So not only is the corporate investment being

[2] eMarkter. http://www.emarketer.com/Article/Social-Video-Sites-Will-See-Big-Boosts-US-Advertiser- Spending/1010300#z5CFEMvLICRf2uvU.99 (accessed February 15, 2014). Social, Video Sites Will See Big Boosts in US Advertiser Spending.

evidenced by dollars spent but also in the time it takes to create and maintain social media efforts.

Twitter Background and related work:

Twitter is a social networking site that allows users to send and read short messages of a maximum of 140 characters. Twitter was created in March 2006 and was officially launched in July 2006. The growth of Twitter has been phenomenal, as was shown by the facts mentioned in the previous section. Users sign up for an account on Twitter, and once they have an account they can begin to "tweet," which is the terminology for sending a message that is popular for Twitter users. Users can subscribe to other users, a process known as "following." These subscribers are known as "followers," which means the followers' home page will be showing the followed users' tweets. By default, tweets that a user sends are public to everyone; however, users can also choose to send tweets specifically to their followers that will not be visible to the public.

Users on Twitter are identified by a user name, and this user name is proceeded by the "@" symbol. When a user identifies another user in their tweet by their user name, it will be visible to the public, and the user that is referenced will be notified by Twitter that they have been "mentioned." This option is either used to mention some user in a new tweet to start a conversation or reply on an existing tweet that has been posted. If a user sees a tweet that is interesting and wants to repost it, they can "retweet" the post, which is similar to forwarding an email message to a new set of users, in this case their

followers. Retweets will generally be identified with an "RT" that is embedded in the heading of the tweet. Messages can be grouped by topic or type by the use of hashtags "#."A hashtag preceding the topic will allow Twitter users to find tweets related to a particular topic when performing a search. This option gives a powerful credibility in analyzing the hashtags as will be mentioned. Twitter also has a location function, which enables the users to choose to turn on their location, and their latitude and longitude will be captured with the tweet.

Tweets can be related to anything, but much of the content on Twitter is related to several key categories. These categories were outlined in research done by Pear Analytics.[3] This study found that tweets were primarily related to six categories: Pointless babble; Conversational; Pass along value; Self-promotion; Spam; News.

Twitter is a conduit for many different types of information, including breaking news (Kwak et al. 2010), political discourse (Conover et al. 2010), community events (Washington Post 2011a), and call for protest (Los Angeles Times 2011). Twitter's reach and diversity of uses makes it a powerful tool for shaping public opinion: indeed Twitter is already being used to defame political candidates and discredit their views (Ratkiewicz et al. 2010; Metaxas and Mustafaraj 2010). Countries such as China are using censors to track internet discussions and shape opinions. Brigham Young University,[4] most people who closely follow both political blogs and traditional news media tend to believe that the content in the blogosphere is more trustworthy.

---

[3] Ryan Kelly. Pear Analytics. http://www.pearanalytics.com/blog/ (accessed February 15, 2014).

[4] Richard Davis. Brigham Young University. http://news.byu.edu/archive09-may-blogs.aspx (accessed February 15, 2014).

There have been many research applications and challenges proposed in the knowledge discovery conferences for facilitating social media, Twitter particularly, to mine, detect, identify, cluster and classify useful information about Twitter users. Such information could be used by marketing companies, news agencies, governments etc ... for different interests and uses. The following is a summary of papers in the field of mining social media data to exploit the general direction of such field:

Roosevelt C. Mosley Jr in 2012 discussed various applications of correlation, clustering and association analyses to social media for insurance companies. The paper demonstrates the analysis of insurance Twitter posts to help identify keywords and concepts which can facilitate the application of this information by insurers. The main theme of the paper is about providing a platform, through social media to proactively address potential market and customer issues when analyzing daily information. The paper also proposes the challenges faced in the process of analyzing social media data such as accessing, collecting and cleaning the data, which is a big dilemma in most social media projects.

Xintian Yang et al. in 2012 presented a dynamic pattern driven approach to summarize data produced by Twitter feeds. The developed novel approach maintains an in-memory summary while retaining sufficient information to facilitate a range of user-specific and topic-specific temporal analytics. Also, in this paper they compare their approach with several state-of-the-art pattern summarization approaches along the axes of storage cost, query accuracy, query flexibility, and efficiency using real data from

Twitter. Their approach is found not only scalable but also outperforms existing approaches by a large margin.

Hila Becker et al. in 2011 explored approaches for analyzing the stream of Twitter messages to distinguish between messages about real-world and non-event messages. The approach relies on a rich family of aggregate statistics of topically similar message clusters based on temporal, social, topical and Twitter-centric features. The authors develop query formulation strategies using those features to retrieve relevant content from various social media sites. Their experiments were applied on datasets from Twitter, YouTube and Flicker to test the effectiveness of the strategies in retrieving the relevant event content.

Geli Fei et al. in 2013 approached the problem of automatic spam detection of reviews by exploiting the burstiness of nature of reviews to identify the review spammers. The reviewers and their occurrence in bursts are modeled as a Markov Random Field (MRF), and employ the Loopy Belief Propagation (LBP) method to infer whether a reviewer is a spammer or not in the graph. The paper proposes several features and employ feature induced message passing in the LBP framework for network inference. Additionally, the paper proposes a novel evaluation method to evaluate the detected spammers automatically using supervised classification of their reviews. The authors employ domain experts to perform a human evaluation of the identified spammers and non-spammers. Both the classification result and human evaluation result show that the proposed method outperforms strong baselines, which demonstrate the effectiveness of the method.

Zhiyuan Chen et al. in 2013 proposed the problem of identifying intention posts in online discussion forums. The author exploits several special characteristics of the problem using a new transfer learning method unlike the general ones used in other research problems. The paper starts with discussing the Expectation Maximization algorithm and its Feature Selection version, and finally the Co-Class algorithm which is inspired by Co-training in (Blum & Mitchell, 1998).

Trending Topics:

In the previous related work discussed we exposed the characteristics and the research done on Twitter and social media generally, this subsection discusses the special work done on discovering the trending topics. Our research will later address the same topic of trending topics on Twitter but using a different technique.

Glivia Barbosa et al. in 2012 aimed to assess the hashtags as a resource for sentiment analysis on Twitter. Their primary results support the hypothesis that hashtags facilitate and provide automatic tracking of users' sentiment on different topics, which in our case consider as the collection of hashtags. This hypothesis shapes our research as will be shown in the methodology chapter towards using hashtags the basic input for finding trending topics on Twitter.

Yiye Ruan et al. in 2012 discussed an approach for predicting microscopic (individual) and macroscopic (collective) user behavioral patterns with respect to specific trending topics on Twitter. The paper seeks to predict the strength of content generation which allows more accurate understanding of Twitter users' behavior and more effective

utilization of the online social network for diffusing information. While previous efforts have been focused on analyzing driving factors in whether and when a user will publish topic-relevant tweets. The paper considers multiple dimensions into one regression-based prediction framework covering network structure, user interaction, content characteristics and past activity. Experimental results on three large Twitter datasets demonstrate the efficacy of the proposed method. They find in particular that combining features from multiple aspects (especially past activity information and network features) yields the best performance. Furthermore, they observe that leveraging more past information leads to better prediction performance, although the marginal benefit is diminishing.

Chi Wang et al. 2013 presented an algorithm for recursively constructing multi-typed topical hierarchies for constructing high quality concept hierarchies that can represent topics at multiple granularities benefits tasks such as search, information browsing, and pattern mining. The idea is based on modelling heterogeneous digital data collections as a heterogeneous information network, linking text with multiple types of entities. The proposed approach handles textual phrases and multiple types of entities by a newly designed clustering and ranking algorithm for heterogeneous network data, as well as mining and ranking topical patterns of different types. Their experiments on datasets from two different domains demonstrate that the algorithm yields high quality, multi-typed topical hierarchies.

Mor Naaman et al. in 2011 contributed in two interesting aspects for interpreting emerging temporal trends in these information systems; they developed a taxonomy of the trends present in the data and identified important dimensions according to which

trends can be categorized. They examined the computed features for different categories of trends quantitatively, and detected significant differences across those categories.

Sentiment Analysis:

Zhiyuan Chen et al. in 2013 proposed a framework to leverage the general knowledge in topic models. Such knowledge is domain independent. Specifically, they use one form of general knowledge, i.e., lexical semantic relations of words such as synonyms, antonyms and adjective attributes, to help produce more coherent topics. However, there is a major obstacle, i.e., a word can have multiple meanings/senses and each meaning often has a different set of synonyms and antonyms. Not every meaning is suitable or correct for a domain. Wrong knowledge can result in poor quality topics. To deal with wrong knowledge, they proposed a new model, called GK- LDA, which is able to effectively exploit the knowledge of lexical relations in dictionaries. There experiments using online product reviews show that GK- LDA performs significantly better than existing state-of-the-art models. We expose such research since we are going to show how we used lexical semantic relations from synonym lists for sentiment analysis, which is a bottleneck in our project.

Carmela Cappelli in 2003 focused on synonym relations between words. A cluster analysis approach is presented, aiming at detecting groups of synonyms of a given term which are characterized by a high degree of homogeneity and therefore are interchangeable. Some applications to the case of Italian words are shown and discussed. The results show that the proposed approach is promising in identifying different senses

of a word. In relation to our work this paper exposes the use of hierarchical clustering for appealing the Dendogram of relations between words driven by synonym list.

Seungyeon Kim et al. in 2012 considered higher dimensional extension of the sentiment concept which represent a richer set of human emotions. The approach's model contains a continuous manifold rather than a finite set of human emotions. The paper investigated the resulting model, compared it to psychological observations, and explored its predictive capabilities. Besides obtaining significant improvement over a baseline without manifold, the paper showed a visualization of different notions of positive sentiment in different domains.

Elif Aktolga et al. in 2013 focused on diversifying the sentiment according to explicit bias to allow users to switch the result perspective to better grasp the polarity of opinionated content, such as during a literature review. For this, the paper first inferred the prior sentiment bias inherent in a controversial topic - the `Topic Sentiment'. Then, utilized this information in 3 different ways to diversify results according to various sentiment biases: Equal diversification to achieve a balanced and unbiased representation of all sentiments on the topic; Diversification towards the Topic Sentiment, in which the actual sentiment bias in the topic is mirrored to emphasize the general perception of the topic; Diversification against the Topic Sentiment, in which documents about the 'minority' or outlying sentiment(s) are boosted and those with the popular sentiment are demoted. In the same sense our research direction, towards sentiment value assignment stage, changed to use scoring and lexical semantic relations instead of positive and negative word lists.

Johan Bollen et al. in 2011 investigated the correlation between the collective mood states derived from large-scale Twitter feeds and the value of the Dow Jones Industrial Average (DJIA) over time. They analyzed the text content of daily Twitter feeds by two mood tracking tools, namely OpinionFinder that measures positive vs. negative mood and Google-Profile of Mood States (GPOMS) that measures mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). They cross-validated the resulting mood time series by comparing their ability to detect the public's response to the presidential election and Thanksgiving Day in 2008. A Granger causality analysis and a Self-Organizing Fuzzy Neural Network were then used to investigate the hypothesis that public mood states, as measured by the OpinionFinder and GPOMS mood time series, were predictive of changes in DJIA closing values. The results indicated that the accuracy of DJIA predictions can be significantly improved by the inclusion of specific public mood dimensions but not others. They found an accuracy of 86.7% in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error (MAPE) by more than 6%.

Cristian Lumezanu et al. in 2012 studied the tweeting behavior of Twitter propagandists, users who consistently express the same opinion or ideology, focusing on two online communities: the 2010 Nevada senate race and the 2011 debt- ceiling debate. They identified several extreme tweeting patterns that could characterize users who spread propaganda: sending high volumes of tweets over short periods of time, retweeting while publishing little original content, quickly retweeting, and colluding with other, seemingly unrelated, users to send duplicate or near-duplicate messages on the

same topic simultaneously. These four features appear to distinguish tweeters who spread propaganda from other more neutral users and could serve as starting point for developing behavioral-based propaganda detection techniques for Twitter.

Opinion Clustering:

Jing Wang et al. in 2012 proposed the problem of identifying the diversionary comments under political blog posts. The paper showed the categorization of diversionary comments under 5 types and proposed an effective technique to rank comments in descending order of being diversionary. The evaluation on 2,109 comments under 20 different blog posts from Digg.com shows that the proposed method achieves the high mean average precision of 92.6%. Sensitivity analysis indicated that the effectiveness of the method is stable under different parameter settings.

Lei Zhang and Bing Liu in 2014 introduced the aspect-based opinion mining method, and discussed the model used for aspect extraction approaches. The paper showed multiple approaches used for topic models like Latent Semantic Allocation (LDA) and Multi-grain LDA. For evaluation they used measures for information extraction such as precision, recall and F-1 scores which are also often used in aspect and entity extraction.

Janyce Wiebe et al. in 2003 proposed the question of ability to building frameworks of mining perspectives of agents. The paper started by discussing the tasks addressed by the MPQA project. Then the paper described the framework for annotating, learning and using information about perspective. Finally, the paper reported the results

of the preliminary annotation study, machine learning experiments, and clustering experiments. In the annotation study, they found that annotators agreed on about 85% of direct expressions of opinion, about 50% of indirect expressions of opinion, and achieved up to 80% kappa agreement on the rhetorical use of perspective. While they did not present the annotation scheme or agreement study in detail, the results demonstrate the feasibility of annotating information about perspective.

CHAPTER 3

METHODOLOGY

Hypothesis & Influence Quantification:

In this chapter we introduce the challenge of measuring the influence or the effect of main stream media on its audience. This readership could be described in different ways using Twitter, which are addressed in this chapter. However, we focus on representing the opinions of Twitter users generally using vector of sentiment that express the bias or neutrality towards multiple different topics. We first start with describing our research questions and hypothesis in comparison with other research works' hypotheses and approaches in quantifying the influence of media, and then we describe the opinion model that we based our analysis and inferences upon.

Research questions:

1) Mass media shaping the audiences' opinions in multiple topics
2) Audience interaction towards information transmitted with the personal influence arising from social NWs

Often, media users may find themselves in disagreement with certain perspectives uncovered in media content. When that occurs, those with oppositional readings to media turn to other sources to find perspectives that align better with their own (Festi

1957). Individuals with particularly high levels of disassociation with the media will frequently experience feelings of dissonance (D'Alessio & Allen 2002). These people then make individual media selections that align with their own views and support their own perspectives. Therefore, on the individual level, acceptance of media messages can often be refuted or assimilated within previously held beliefs and not immediately accepted as part of one's own reality. This does not refute the systemic ideological biases embedded within all media (Herman & Chomsky 1988). At the macro-level, one can see ideological consistency throughout society and across media outlets. Quantifying the influence of mass media through Twitter could help us find the factor at which the society relies on news outlets without evaluating the content before agreeing with it. Thus, our research question concludes into whether if the mass media shape individuals opinions? And how does the audience interact towards the information transmitted with the personal influence arising from social media?

Other Approaches:

In this subsection we mention three other approaches for solving the research question proposed earlier:

Zhongyu Wei et al. in 2013 analyzed the behavior of mainstream media on Twitter and studied how they exert their influence to shape the public opinion. The hypothesis of this question is that Twitter gives the brief picture about the basic ecology

habit of mass media in influencing public opinion. The paper considered three questions to answer, which are how to quantify bias on Twitter? How information originated from media propagates on Twitter? And how mass media compares with the most influential individuals in terms of social influence? The method was applied on a Twitter dataset collected about the UK general elections, were three major parties played a role. To answer those questions the paper proposed an empirical measure to quantify media bias based on sentiment analysis. First, they try traditional lexicon-based sentiment analysis methods, which failed, since more than 61% of the tweets contain sentiment about more than one party. Thus, they used OpenAmplify for entity-level sentiment extraction from tweets. The results showed 54% accuracy when using the traditional lexicon-based sentiment analysis, while 74% when using OpenAmplify. The quantified media bias measure in this paper is represented by the following equation:

$$Media\ Bias_{ij} = \frac{C_{ij}^{pos} + 1}{C_{ij}^{neg} + 1} - 1$$

Where C_{ij}^{pos} and C_{ij}^{neg} denotes the total number of positive and negative tweets from a media outlet i towards a party j. Media Bias takes value 0 if there is no bias. And it is positive for positive bias and negative vice versa.

Then the paper transitioned to the analysis of media intermediates by studying the information propagation. The information propagation is addressed as the retweets which are used to replicate information from news Twitter pages. The intermediates are defined

as the direct re-tweeters, and their contribution is measured by several categories, for example, the retweet rate, the average retweet times per tweet and the life span. Those measures are applied to compare between multiple categories of intermediates like celebrities, bloggers, mainstream media and journalists. Similarly, [60] presented a measure for the tweeting behavior of propagandists on Twitter, and showed the effects through retweets.

Lastly, the paper compared the information diffusion patterns from different categories of sources. Supposing a single information cascade is generated by seed tweet followed by all of its retweets, they calculated the distribution of information cascades by source category, and the observation is that most information cascades are originated from media (including mainstream media and social media) and party.

The second approach introduced by Seth Myers et al. in 2012 focused on both internal and external influence on social networks. In their model they distinguished between exposures and infections. An exposure event occurs when a node gets exposed to information, and an infection event occurs when a node reposts a tweet with the same information. Exposures to information lead to an infection. They developed an estimation technique from a given network and a set of node infection times. The event profile is defined as the user that absorbs external information to the rest of the nodes. The event profiles quantify the number of exposures generated by the external source over time. Additionally, they infer the exposure curve that models the probability of infection as a function of the number of exposures of a node. They experimented with their model on

Twitter and found that the occurrence external out-of-network events are detected accurately, and the exposure curve inferred from the model is often 50% more accurate than baseline methods. However the model was fitted to thousands of different URL's that have appeared across Twitter users, and used the inferred parameters of the model to provide insights into the mechanics of the emergence of these URLs.

The third approach is introduced by DeMarzo et al. in 2003, which proposed a boundedly rational model of opinion formation in which individuals are subject to persuasion bias; that is, they fail to account for possible repetition in the information they receive. They showed that persuasion bias implies the phenomenon of social influence, whereby one's influence on group opinions depends not only on accuracy, but also on how well-connected one is in the social network that determines communication. Persuasion bias also implies the phenomenon of *unidimensional* opinions; that is, individuals' opinions over a multidimensional set of issues converge to a single "left-right" spectrum. They explored the implications of their model in several natural settings, including political science and marketing, and obtained a number of novel empirical implications.

Targeted audience:

Similarly as Seth Myers et al. we distinguish between exposures and infections. Unlike Seth Myers et al. and DeMarzo et al. we disregard internal infections, which mean that our main focus is on analyzing external influences only. When a node U gets

exposed to or becomes aware of information *I* whenever one of its neighbors in the social network posts a tweet containing *I* (we call this an internal exposure). However, we consider internal exposures, since the task of distinguishing between internal exposures and infections is a very challenging problem. From our results, we observed another category of users which depends on each news outlet separately. This category concerns news channel referrers and non-referrers. For example, news referrers of Fox news are the users who mentioned Fox news whether using hashtags or without.

Model:

In our approach we model the opinion of Twitter users subjected to persuasion bias from mass media, unlike DeMarzo et al., their model tests the persuasion bias internally. Thus, we are concerned about the phenomena of *unidimensional* opinions in afore mentioned paragraph to be the basic measure of influence. Our hypothesis is that blind (loyal) followers to a particular news channel fall into the same herd of opinions and express their *unidimensional* opinion. One of the main features that differentiate *unidimensional* opinions from other diverse perspectives is the isolation property. According to such assumption we defined the main task is to detect isolated opinions on multiple issues (topics). Then we quantify the assurance factor of influence of a particular channel as the percentage of tweets which referred that news channels out of the total number of isolated tweets. We assume that news channels referred in a tweet is the source of information that resulted in biasing the opinion of that tweet.

Our aspect-based opinion mining framework is based on modelling opinions into vectors of sentiment towards different topics T_j. An opinion O_i expressed in a $tweet_i$ using the sentiment based assignment values S_T for each of the topics from T_1 to T_n follow the vector representation:

$$O_i = \{S_1, S_2, \ldots, S_n\}$$

Sentiment values depend on the method on which we categorize the sentiment, which will be mentioned in more details in the next section (i.e. scoring, groups, trivial polarity). However, each method uses one of the categories at a time. An opinion group O_g is a set of combinations of sentiment vectors that are very similar to each other. Those groups of opinions are clustered using Expectation Maximization (EM) algorithm.

The problem of recognizing blind followers relies on detecting which group of clustered opinions is isolated from the rest of the clusters. Thus, we are looking at the distribution of clusters among the sentiment towards each topic, while considering the number of referrers that are in the isolated cluster. One of the main advantages of using EM algorithm is that the results indicate the mean and the standard deviation of the clusters towards each attribute, which is the topic in our context. An isolated cluster is defined as the cluster that has no other overlapping clusters in terms of the sentiment values that it spans to a certain topic. The isolated clusters are defined as the ones that do not overlap with other clusters. By this definition we can calculate the minimum and maximum of each cluster using the mean and standard deviation resulting from EM

algorithm, then find overlapping and non-overlapping clusters. Consequently, the non-overlapping clusters are the isolated ones. To understand the importance of detecting isolated opinion groups, we show the resulting visuals of the EM algorithm. The visuals contribute to show the other point of view to the isolation property of opinion groups, which is diversity. Diversity is claimed for a certain range of sentiment values towards a topic, where this range should contain more than one opinion group if it is diverse. However, the diversity cannot be quantified, only through the negation effect of isolation.

Framework:

In this section we reveal the framework of algorithms and techniques used to mine the opinions of Twitter users towards multiple the trending topics, inside the collected dataset. We describe in details the languages and tools used and all technical difficulties faced through the project. The framework is composed of three steps:

- Trending Topics extraction
- Sentiment Analysis
- Opinion clustering

As shown in figure 1, we first start with mining the trending topics using Apriori algorithm from two different inputs, the hashtags and the most frequent words. The elements in white circles are the optional inputs which could be provided to the step, where it refers to, which means that either of the inputs is experimented one at a time. The difference between using both inputs is explained in the results chapter. The output

of the Apriori algorithm is the frequent itemsets, where each word is an item representing a topic that concerns the users. The second step is calculating and assigning the sentiment to construct the sentiment matrix, which the clustering process is based on. The sentiment values used are categorized into three; trivial polarity, adjective hierarchy and scoring, where each category resulted into different number, distribution and output formats of clusters. The sentiment categories are explained in details in the next section, and the resulted clusters from each category are discussed in the results chapter.

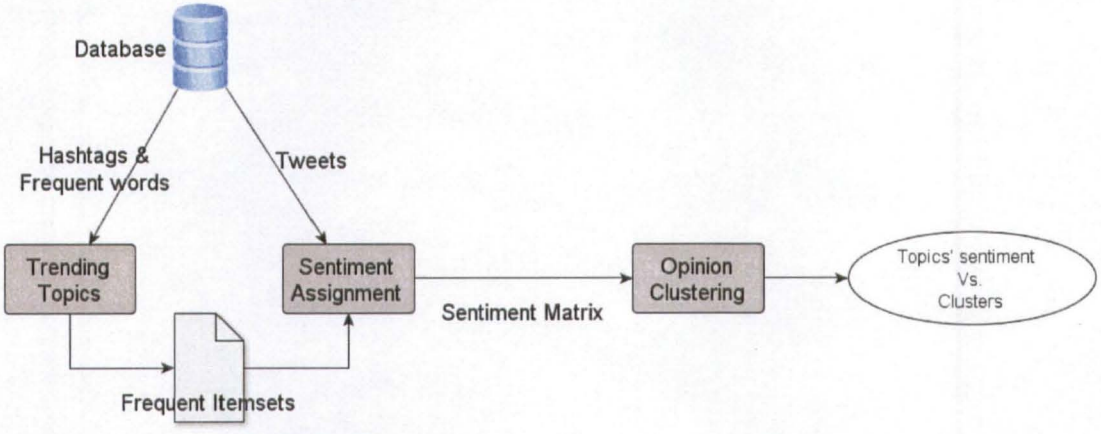

Figure 1. The framework of the aspect-based opinion mining process utilizes those three main steps: Trendngin Topics, Sentiment Assignment and Opinion Clustering.

To our knowledge this framework has not been investigated by any research work before, and the validation proves the compliance to the hypothesis mentioned with those steps. The data collection, the different analysis methods and their results are discussed in the next chapter (Results & Discussion).

Trending Topics extraction (Apriori):

In this section we cast the challenge of finding the frequent itemset problem as the trending topics by the dataset collected through keywords. Although, one can think that by default that the harvesting keywords used in streaming the tweets will be mostly the dominant factor and similar to the output of frequent itemsets, the results show that it is not totally true. Here we describe the Apriori algorithm, which was used to find the trending topics in the collected tweets. We conducted two main experiments to mine the trending topics. In the first we used the most frequent words as the input but filtering out stop words, while in the second we used all hashtags instead. The results are explained in the next chapter to fill out the reasoning of which method is better (Latiri et al. 2001).

Apriori:

The indexing structure for a collection of indexed tweets T containing different combinations of a keyword set A can be used as a basis for information extraction and the goal would be extracting significant keyword associations. Consider a set of key- words $A = \{w_1, w_2, \ldots, w_m\}$ and a collection of indexed tweets $T = \{t_1, t_2, \ldots, t_n\}$ (i.e. each t_i is associated with a subset of A denoted $t_i(A)$). Let $W \subseteq A$ be a set of key-words, the set of all tweets t_i in T such that $W \subseteq t(A)$ will be called the covering set for W and denoted $[W]$. Any pair (W, w), where $W \subseteq A$ is a set of keywords and $w \in A/W$, will be called an association rule (or simply an association), and denoted $W \Rightarrow w$.

Given an association rule R: $(W \Rightarrow w)$;

- $S(R,T) = |[W \cup \{w\}]|$ is called the support count for rule R with respect to the collection of tweets T ($|X|$ *denotes the size of the set* X)
- $C(R,T) = \frac{|[W \cup \{w\}]|}{|[W]|}$ is called the confidence of rule R with respect to the collection of tweets T

Notice that $C(R,T)$ is an approximation (maximum likelihood estimate) of the conditional probability for a text of being indexed by the keyword w if it is already indexed by the key-word set W. An association rule R generated from a collection of texts T is said to satisfy support and confidence constraints σ and γ if

$S(R,T) \geq \sigma$ And $C(R,T) \geq \gamma$

To simplify notations, $[W \cup \{w\}]$ will be often written $[W\ w]$ and a rule R: $(W \Rightarrow w)$ satisfying given support and confidence constraints will be simply written as:

$W \Rightarrow w$, where $S(R,T)/C(R,T)$

Informally, for an association rule $(W \Rightarrow w)$, such σ/γ constraints can be interpreted as: there exist a significant number of tweets (at least σ), for which being related to the topic characterized by the keyword set W implies (with a conditional probability estimated by γ) to be also related to the topic characterized by the keyword w.

As far as the actual association extraction is concerned, the common procedures are usually two steps algorithms; First generate all the keywords sets with support at least equal to σ (i.e. all the keywords sets W such that $|[W]| \geq \sigma$). The generated keywords sets are called the frequent sets (or σcovers). Second generate all the association rules that can be derived from the produced frequent sets and that satisfy the confidence constraint γ.

The frequent sets are obtained by incremental algorithms that explore the possible subsets, starting from the frequent singletons (1-frequent itemsets) (i.e. the $\{w\}$ such that $|[\{w\}]| \geq \sigma_{min}$) and iteratively adding only those keywords that produce new frequent sets, which become the candidate itemsets. This step is the most computationally expensive (exponential in the worst case with the 2-candidate itemsets). Then the pruning step takes care of the eligible itemsets to be tested for the support count filter.

The associations derived from a frequent set W are then obtained by generating all the implications of the form $W/\{w\} \Rightarrow w$, $(w \in W)$, and keeping only the ones satisfying the confidence constraint γ. Some additional treatment (structural pruning) is usually the following step after support counts filter that reduces the number of candidates. Nevertheless, we did not consider the second step in finding the association rules, since we are looking for frequent sets only in the trending topics case (Chengqi Zhang, Shichao Zhang et al. 2002).

On the implementation side, figure 2 shows the generic view of the incremental procedure for finding the candidate itemsets and the frequent itemsets.

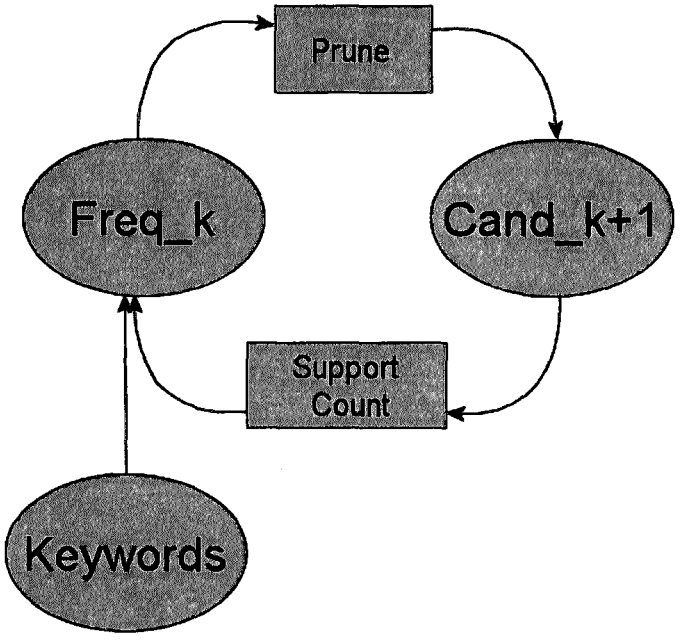

Figure 2. The two alternating steps of the Apriori algorithm between pruning and support count filtering.

The following steps intend to show the pseudo-code which was implemented in C++ on fedora for conducting our experiments described in the next chapter:

C_k: Candidate itemset of size k

L_k: Frequent itemset k

$L_1 = \{frequent\ items\};$

for ($k = 1; L_k! = \emptyset; k++$) do begin

C_{k+1} =candidates generated from L_k

for each tweet t_i in the database do

increment the count of all candidates in C_{k+1}that are contained in t_i

L_{k+1} =candiates in C_{k+1} with support count $\geq \sigma_{min}$

end

return $\cup_k L_k$;

end

It is very important to demonstrate the pruning step, since it reduces the memory space consumed between each incremental step and heavy computation due to large C_{k+1} generated. A next candidate C_{k+1} is said to satisfy the pruning condition, when all its subsets are present in the frequent itemset L_k. For example, a candidate $\{A, B, C\}$ passes the pruning step if and only if $\{A, B\}$, $\{A, C\}$ and $\{B, C\}$ are present in the frequent itemset. The following example is intended to show the whole process of pruning the frequent itemsets using their subsets and filtering the candidate itemsets using σ_{min}.

Consider the database consisting of 9 tweets in the table 1.1, and suppose the $\sigma_{min} = 22\%$, which means 2 out of the 9 tweets. The items are numbered with a prefix I.

Table 1.1. Transactions of tweets example

Tweet ID	List of keywords
1	w_1, w_2, w_5
2	w_2, w_4
3	w_2, w_3
4	w_1, w_2, w_4
5	w_1, w_3
6	w_2, w_3
7	w_1, w_3
8	w_1, w_2, w_3, w_5
9	w_1, w_2, w_3

Table 1.2. The support counts of 1-itemsets according to table 1.1

Itemsets L_1	Support Counts
w_1	6
w_2	7
w_3	6
w_4	2
w_5	2

The first step is to generate the 1-itemset frequent pattern, which can be found by counting the frequency of each item individually. It appears that all candidates satisfy the σ_{min} of 22% specified previously. Now it is time to generate the 2-itemset candidate pattern, which is the following table, with their support counts:

Table 1.3. The support counts of 2-temsets according to table 1.1

Itemsets L_2	Support Counts
w_1, w_2	4
w_1, w_3	4
w_1, w_4	1
w_1, w_5	2
w_2, w_3	4
w_2, w_4	2
w_2, w_5	2
w_3, w_4	0
w_3, w_5	1
w_4, w_5	0

Although, this is the second least candidate pattern in number of items, it contains the largest number of itemsets possibilities in comparison with other n-itemsets candidate patterns. Thus, the advantage of allocating memory incrementally is appreciated when

pruning is applied. And by applying the filter of minimum support count the following is the 2-itemset frequent pattern:

Table 1.4. The support counts of 2-frequent itemsets after filtering according to the minimum support counts.

Itemsets L_2	Support Counts
w_1, w_2	4
w_1, w_3	4
w_1, w_5	2
w_2, w_3	4
w_2, w_4	2
w_2, w_5	2

Till now we have not used the Apriori property yet, since the pruning effect has not been applied. It will be more obvious now when generating the 3-itemsets candidate pattern. Transitioning to C_3 requires the initial suggested candidates which requires joining the items as following:

$$C_3 = \{\{w_1, w_2, w_3\}, \{w_1, w_2, w_5\}, \{w_1, w_3, w_5\}, \{w_2, w_3, w_4\}, \{w_2, w_3, w_5\}, \{w_2, w_4, w_5\}\}$$

For example, $\{w_1, w_2, w_3\}$, the 2-item subsets of it are $\{w_1, w_2\}$, $\{w_2, w_3\}$ and $\{w_1, w_3\}$. Since all 2-item subsets of $\{w_1, w_2, w_3\}$are members of W_2, we will keep $\{w_1, w_2, w_3\}$ in C_3. Another contrary example, $\{w_2, w_3, w_5\}$ which shows how the pruning is performed. The 2-item subsets are $\{w_2, w_3\}$, $\{w_2, w_5\}$ and $\{w_3, w_5\}$, but

$\{w_3, w_5\}$ is not a member in W_2 and hence it is not frequent, violating the Apriori property. Thus, we will remove the $\{w_1, w_2, w_3\}$ from C_3.

Therefore, $C_3 = \{\{w_1, w_2, w_3\}, \{w_1, w_2, w_5\}\}$, which satisfy the minimum support count to be the W_3. Finally, when transitioning to the 4-itemset candidate pattern the join operation on L_3fails to generate any itemset for $C_4 = \emptyset$. The algorithm terminates, having found all of the frequent itemsets.

The last step is generating the association rules from the frequent itemsets resulted. However, we did not use the association rules to represent the trending topics; we only used those important words that were inside different sizes of the frequent itemsets. For each frequent itemset W, all nonempty subsets s of W are generated. Then for every nonempty subset $s$of W, an output rule is "$s \Rightarrow W - s$" if $\frac{supportCount(L)}{supportCount(s)} \geq \gamma_{min}$. Using the same example if we took $\{w_1, w_2, w_5\}$, all its nonempty subsets are $\{\{w_1, w_2\}, \{w_1, w_5\}, \{w_2, w_5\}, \{w_1\}, \{w_2\}, \{w_5\}\}$ and $\gamma_{min} = 0.7$. Thus, the selected resulting rules from the table 1.5 are the ones above 70%:

Table 1.5. The confidence level of potential rules, the red marked ones are above the minimum confidence used in this example. The notation of the $sc()$ function means the support count of the itemset between the parentheses.

Rules	Confidence
$\{w_1, w_2\} \Rightarrow w_5$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_1, w_2\})} = \frac{2}{4} = 50\%$
$\{w_1, w_5\} \Rightarrow w_2$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_1, w_5\})} = \frac{2}{2} = 100\%$
$\{w_2, w_5\} \Rightarrow w_1$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_2, w_5\})} = \frac{2}{2} = 100\%$
$w_1 \Rightarrow \{w_2, w_5\}$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_1\})} = \frac{2}{6} = 33\%$
$w_2 \Rightarrow \{w_1, w_5\}$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_2\})} = \frac{2}{7} = 29\%$
$w_5 \Rightarrow \{w_1, w_2\}$	$\frac{sc(\{w_1, w_2, w_5\})}{sc(\{w_5\})} = \frac{2}{2} = 100\%$

There is one last implementation issue that is worth mentioning for memory reduction during the generation of candidate itemsets. The code is shown in appendix A, where the transition between the alternating steps (filtering and pruning) is highlighted in yellow. Since the generation of suggestions for candidate itemsets before pruning exponentially consumes the memory, we efficiently implement this step by integrating it with the support count filtering step to test each individual itemset separately then include it in the frequent itemset if satisfies σ_{min}. That means if we have a candidate itemset generated from L_k we pass it individually, without storing it in an actual C_{k+1} of itemsets, to be tested for pruning. Then if it passed the pruning it is tested for the support count.

The cycle is repeated when the itemsets in L_k are all tested and stored in L_{k+1}. The only exceptional step is C_2, since we need to generate all possible combinations between the 1-itemsets frequent patterns. Thus, as clarified by the code comments we separate the steps of generating the 1 & 2-itemsets frequent patterns and the generic number-itemsets frequent patterns.

Sentiment Analysis:

In this step we propose three different approaches in defining the sentiment used then assign for each tweet the appropriate sentiment according to the category defined. The sentiment assignment totally depends on the adjective used in the tweets towards different topics. Nevertheless, we consider only tweets which have one adjective. According to the value and category the adjective falls into, the sentiment assigned only to the topics mentioned in the same tweet, while the rest of the topics are assigned to be neutral (zero). For example, if the adjective was recognized to a corresponding value of x and the only mentioned topics are of index 1, 3 and 4 out of K topics, then the sentiment vector representing this $tweet_i$ will be as following:

$$O_i = \{x, 0, x, x, 0, \dots\}_K$$

In the three methods we proposed, the NLTK[1] platform implemented in python to detect the adjectives in the tweets. NLTK is a leading platform for building Python programs for NLP. It provides easy-to-use interfaces to over 50 corpora and lexical

[1] Natural Language Processing Toolkit. http://nltk.googlecode.com/svn/trunk/doc/howto/wordnet.html (accessed February 15, 2014).

resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, and tagging, parsing, and semantic reasoning.

The following are the three different methods for assigning the sentiment of the tweets:

1. Trivial polarity
2. Scoring
3. Adjective Hierarchy (semantic relatedness)

Trivial polarity: In this method we downloaded two lists of positive and negative adjectives. We developed programs in python to extract the adjectives by tokenizing and then tagging the sentences in the tweets. The words which match the tag "JJ" are the adjectives, thus we compare those words with both the positive and negative lists downloaded. If the adjective matches a word in the positive list the nominal value "P" is assigned, while if it matches a word in the negative list the nominal value "N" is assigned, if it did not match any of the lists a nominal value of "N" is assigned. However, some tweets contain more than one adjective, and if both contradict by matching both the positive and negative lists, the nominal value "M" is assigned.

Scoring: In this method we also downloaded a list containing 2,477 adjectives and their scores rated from -5 to +5 by Finn Nielsen in 2009-2011. The list is called "AFINN" and can be downloaded from.[2] This list was used by Lars Kai Hansen et al. in 2011 for sentiment analysis on Twitter. The same process of tokenizing and tagging the sentences

[2] Finn Nielsen. DTU Compute. http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010 (accessed February 15, 2014).

takes place in this method too but the adjectives are compared with the scoring list. The score of the adjective is assigned to the topics mentioned in the vector, and if there is more than one adjective in the tweet, the average replaces both scores.

Adjective Hierarchy: In this method we list all adjectives used in the analyzed tweets, also using tokenization and tagging. The goal of listing all the adjectives is to find how they are related semantically using the lexicon imported from WordNet, and then group those words which are the closest to each other as groups of sentiment. Those groups are the basics of sentiment values in this method. The semantic relations give the distance between each adjective and the other through the synonym list. We first look up the synonym list of each adjective in the list through the synstes() function. The other parts of speech are NOUN, ADJ and ADV. A synset is identified with a 3-part name of the form: word.pos.nn. NLTK also facilitates functions to obtain the definition, examples, lemmas and the lemmas' sysnets. Synets by the NLTK definition is a set of synonyms that share a common meaning. Each synset contains one or more lemmas, which represent a specific sense of a specific word.

Thus, we give the following definitions from[3] as a reference for the reader to interpret the linguistic meaning of:

- Synonyms: are words with the same or similar meanings.
- Antonyms: a word opposite in meaning to another. Fast is an antonym of slow.

[3] About.com. Grammar & Composition. http://grammar.about.com/ (accessed February 15, 2014).

- Hypernym: A linguistic term for a word whose meaning includes the meanings of other words. For instance, flower is a hypernym of daisy and rose.
- Hyponym: In linguistics, a specific term used to designate a member of a class. For instance, daisy and rose are hyponyms of flower.
- Holonyms: A term that denotes a whole whose part is denoted by another term, such as 'face' in relation to 'eye'.[4]
- Pertainyms: (computational linguistics) a word, usually an adjective, which can be defined as "of or pertaining to" another word.

However, some relations have to be defined by WordNet only over Lemmas (i.e. antonyms, derivationally related forms and pertainyms), where Lemmas can also have relations between them, which can only apply on Lemmas not on synsets. At the end we only used the sysnset() function of the adjectives without restricting a pos argument to them in order to calculate the score of the similarity between their each other's senses. There are multiple ways to calculate this score that denotes how two similar word senses are.

First the synonym lists are retrieved for each adjective using the synset() function. Using NLTK we have three options for denoting the similarity between both words: Path Similarity, Leacock-Chodorow Similarity and Wu-Palmer Similarity. The Wu-Palmer Similarity function returns a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Sub-summer

[4] Wiktionary. Holonyms. http://en.wiktionary.org/wiki/ (accessed February 15, 2014).

(LCS) (most specific ancestor node). Note that at this time the scores given do not always agree with those given by Pedersen's Perl implementation of WordNet Similarity. The LCS does not necessarily feature in the shortest path connecting the two senses, as it is by definition the common ancestor deepest in the taxonomy, not closest to the two senses. Typically, however, it will so feature. Where multiple candidates for the LCS exist that whose shortest path to the root node is the longest will be selected. Where the LCS has multiple paths to the root, the longer path is used for the purposes of the calculation.

Additionally, the same three functions can be used with different information content dictionary imported from the optional corpora. Information Content (IC): loads an information content file from the wordnet_ic corpus, where we can also specify the information content of certain lists to be held in variables. Moreover, there is an option to create an information content dictionary from a corpus (or any corpus that has a words() method). We used the Wu-Palmer similarity since it features the common ancestor deepest in the taxonomy not closest to the two senses. We collect all the adjectives in the tweets and calculate the distance matrix in terms of Wu-Palmer similarity. Finding the similarity is based on the SemCor corpus which is a subset of the Brown corpus. SemCor corpus is a sense-tagged corpora created at Princeton University by the WordNet Project research team,[5] which defines the relational taxonomy between words. The reason for using the SemCor corpus is that it has the highest percentage of adjective connections. The distance matrix then is used to construct the hierarchy of the adjectives within the

[5] Gabormelli. http://www.gabormelli.com/RKB/SemCor_Corpus (accessed February 15, 2014). SemCor Corpus.

list, as an input for the hierarchical clustering algorithm. By this hierarchy we grouped the adjectives as the sentiment values, so the sentiment values of the tweet will depend on adjective choice that was used from those groups. We used the R programming language to apply the hierarchical clustering algorithm, and the input and output formats and the functions are explained in this section too.

Hierarchical Clustering Algorithm:

Hierarchical clustering algorithm is used in many data mining applications to build a binary tree of data that successively merges similar groups of points. Visualizing such information provides useful summary of the data, but we used this type of tree, which is called "Dendogram," in our analysis to define a threshold separating the adjectives into groups of sentiment values. This separation could be defined number of groups or level based. The algorithm only requires a measure of similarity or dissimilarity between groups of data points. At first each point could be viewed as an entity group by itself, then the algorithm decides to merge pairs of these groups incrementally until all of the data points are one single group. This type of hierarchical clustering is called "Agglomerative." While if all data points at first are considered as a single group then algorithm works the opposite way by splitting up this group into pairs incrementally, then it is said to be "Divisive."

There are several types of metrics that can be used, which are basically the formula on which the distance matrix was built upon. For example, the Euclidean

distance squared Euclidean distance, Manhattan distance, maximum distance, Mahalanobis distance, cosine similarity, Hamming distance and Levenshtein distance, where their equations are shown below. Although, all of these metrics are the standards used in most of the applications, the most appropriate metric is based on the scoring that denotes the similarity between word senses. We convert this similarity into dissimilarity matrix by the similarity score from one, since the maximum score is one. The reason for using dissimilarity matrix is that most of the free software (i.e. R and Weka) available now has the standard of using it instead of the similarity matrix, except if it is an option to change. The following are the formulas for the standard metric criteria that can be used:

$$Eucledian\ distance: \left||a-b|\right|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

$$\text{squared Euclidean distance: } \left||a-b|\right|_2^2 = \sum_i (a_i - b_i)^2$$

$$Manhattan\ distance: \left||a-b|\right|_1 = \sum_i |a_i - b_i|$$

$$Maximum\ distance: \left||a-b|\right|_\infty = max_i|a_i - b_i|$$

$$Mahalanobis\ distance: \sqrt{(a-b)^T S^{-1}(a-b)}, where\ S\ is\ the\ covariance\ matrix$$

$$Cosine\ similarity: \frac{ab}{||a||||b||}$$

Another feature in the hierarchical clustering algorithm that should be specified when using is the linkage criteria. The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations. Some commonly used linkage criteria between two sets of observations A and B, where d is the chosen metric, are (SAS/STAT 9.2 Users Guide):

$$Maximum\ or\ complete\ linkage\ clustering: \max\{d(a,b): a \in A, b \in B\}$$

$$Minimum\ or\ single-linkage\ clustering: \min\{d(a,b): a \in A, b \in B\}$$

$$Mean\ or\ average\ linkage\ clustering, or\ UPGMA: \frac{1}{|A|\,|B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$$

$$Minimum\ energy\ clustering: \frac{2}{nm}\sum_{i,j=1}^{n,m}||a_i - b_i||_2 - \frac{1}{n^2}\sum_{i,j=1}^{n}||a_i - a_j||_2 - \frac{1}{m^2}\sum_{i,j=1}^{m}||b_i - b_j||_2$$

Apparently, the distance matrix is replicated on both sides of the diagonal, which is an advantage in our case that we utilized to reduce complexity by half when calculating the dissimilarity matrix between adjectives. We calculate only the lower part of the distance matrix to input it into the hierarchical clustering algorithm. The algorithm starts with finding the closest pair of words to merge them into a single cluster. Then the distance from this new compound object to all other objects is computed. In our case we used the single linkage criteria. In single link clustering the rule is that the distance from

the compound object to another object is equal to the shortest distance from any member of the cluster to the outside object. The process is repeated several times until finally the last two clusters are merged at a certain level, and the process is summarized by the a hierarchical tree (Dendogram), where we would see how the adjectives merge at different heights. Thus, the adjectives can be grouped using a certain value of level or by specifying the number of groups that needs to be formed from the concluded structure.

Hierarchical clustering using R programming:

The R programming software is available online for free, which is used by many analysts in the industry, due to its ease-of-use and portability on various types of machines (i.e. OSX, Windows, Linux). It is installed on our fedora machine at CAU. Our concern is to use the hierarchical clustering algorithm to find the semantic relation between the adjectives used in the tweets collected and build a hierarchical structure and Dendogram to observe how those adjectives could be grouped. The algorithm is implemented using the method:[6]

$$hclust()$$

This function performs a hierarchical cluster analysis using a set of dissimilarities for the n objects being clustered. Initially, each object is assigned to its own cluster and then the algorithm proceeds iteratively, at each stage joining the two closest clusters, continuing until there is just a single cluster. At each stage distances between clusters are

[6] Swiss Federal Institute of Technology Zurich. http://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html (accessed February 15, 2014). Hierarchical Clustering (R-manual).

recomputed by the Lance–Williams dissimilarity update formula according to the particular clustering method being used. However, there is a number of different clustering methods are provided. Ward's minimum variance method aims at finding compact, spherical clusters. The complete linkage method finds similar clusters. The single linkage method (which is closely related to the minimal spanning tree) adopts a 'friends of friends' clustering strategy. The other methods can be regarded as aiming for clusters with characteristics somewhere between the single and complete link methods. Note however, that methods "median" and "centroid" are not leading to a monotone distance measure, or equivalently the resulting Dendrograms can have so called inversions (which are hard to interpret).

We used R programming for clustering the adjectives into groups by applying the hierarchical clustering algorithm implemented in R. The script first collects the tweets then extracts all the adjectives using NLTK to put them in a list. This list is used to find the distance matrix between each adjective and the other. Lastly, the R script scans this file of distance matrix to convert it into a distance object for the *hclust* function as shown in the steps below. Thus, we follow the steps of scanning the distance matrix and converting it to a distance object representing all adjectives as separate objects to build the Dendogram upon. As shown we follow these steps to divide the adjectives into groups through the hierarchical structure created from their semantic relatedness:

1. Scan the lower the file of the distance matrix
2. Calculate the number of columns of the matrix

3. Create an empty matrix with the number of rows and columns as the number calculated
4. Scan the file into the matrix created
5. Transpose the matrix
6. Row and column bind the matrix
7. Convert the distance matrix into a distance object
8. Execute the agglomerative hierarchical clustering method from the distance object using single linkage method
9. Cut the tree to create five separate groups of sentiment
10. Write a file containing each adjective and its corresponding group
11. Plot the tree (Dendogram)

An object of class hclust is a list with several output components that describes the tree produced by the clustering process. These components describe the merging of the clusters, the clustering height, the original observations suitable for plotting, labels for each of the clustered objects and the distance and cluster method that has been used.

Opinion clustering (Expectation-Maximization Algorithm):

The last step of our framework is the goal step of fitting each tweet into a cluster of possible opinions (vector of sentiment). EM assigns a probability distribution to each tweet which indicates the probability of it belonging to each of the clusters. EM can

decide how many clusters to create by cross validation, or by previously specifying how many clusters to generate.

Generally, EM algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM algorithm is the most suitable clustering algorithm since it enables parameter estimation of the distributions in probabilistic models with incomplete data, which we call the "Incomplete Case." In our model the latent (hidden) variable here is the source of the opinion, where it could be a news channel or other external influences (i.e. other tweets, classmates, co-workers, friends, family, etc.). In order to simplify the explanation we start with giving an example of a simple opinion tracking experiment. Lastly, we show how we used Weka to find the clusters' mean and standard deviation.

Consider a simple opinion tracking experiment in which we track the sentiment of two Twitter pages managed by two news channels with unknown biases θ_A and θ_B respectively (channel A has a positive sentiment towards a topic with probability θ_A and negative sentiment with probability $1 - \theta_A$ and similarly for channel B). Our goal is to estimate $\theta = (\theta_A, \theta_B)$ by repeating the following procedure five times: randomly choose one of the two channels, and perform ten independent sentiment assignments posted by the selected channel about a single topic. Thus, the entire procedure involves a total of 50 tweets analyzed (table 2.1).

Table 2.1. The complete case of the opinion tracking experiment.

Channel ID	Sentiment of 10 tweets	Channel A's sentiment counts	Channel B's sentiment counts
B	+, -, -, -, +, +, -, +, -, +		5 +, 5 -
A	+, +, +, +, -, +, +, +, +, +	9 +, 1 -	
A	+, -, +, +, +, +, +, -, +, +	8 +, 2 -	
B	+, -, +, -, -, -, +, +, -, -		4 +, 6 -
A	-, +, +, +, -, +, +, +,-, +	7 +, 3 -	
Total sentiment counts		24 +, 6 -	9 +, 11 -

During the experiment, suppose that we keep track of two vectors $x = (x_1, x_2, \dots, x_5)$ and $z = (z_1, z_2, \dots, z_5)$ where $x_i \in \{0,1, \dots, 10\}$ are the number of positive sentiment observed during the i_{th} set of tweets, and $z_i \in \{A, B\}$ is the identity of the channel used during the i_{th} set of tweets analyzed. Parameter estimation in this setting is known as the complete data case in that the values of all relevant variables in this model (the sentiment towards the topic and the news channel posted the set of tweets) are known. Here, a simple way to estimate θ_A and θ_B is to return the observed proportions of positive sentiment for each channel:

$$\hat{\theta}_A = \frac{\#\ of\ poistive\ sentiment\ posted\ by\ channel\ A}{total\ \#\ of\ tweets\ posted\ by\ channel\ A\ about\ the\ topic}$$

$$\hat{\theta}_B = \frac{\#\ of\ poistive\ sentiment\ posted\ by\ channel\ B}{total\ \#\ of\ tweets\ posted\ by\ channel\ B\ about\ the\ topic}$$

This intuitive guess is, in fact, known in the statistical literature as maximum likelihood estimation (the maximum likelihood method assesses the quality of a statistical model based on the probability it assigns to the observed data). If $logP(x, z; \theta)$ is the logarithm of the joint probability (or log-likelihood) of obtaining any particular vector of observed positive sentiment counts x and channel identities z, then the formulas above solve for the parameters $\hat{\theta} = (\hat{\theta}_A, \hat{\theta}_B)$ that maximize $logP(x, z; \theta)$.

Now consider a more challenging variant of the parameter estimation problem in which we are given the recorded positive sentiment counts x but not the identities z of the channels that posted each set of the tweets. We refer to z as hidden variables or latent factors, which in our model represent the source of opinion which we want to reveal. Parameter estimation in this new setting is known as the incomplete data case. This time, computing proportions of positive sentiment for each channel is no longer possible, because in this setting we assume do not know the source of the tweet. However, if we had some way of completing the data (guessing correctly which channel posted in each set of the tweets), then we could reduce parameter estimation for this problem with incomplete data to maximum likelihood estimation with complete data.

One iterative scheme for obtaining completions could work as follows: starting from some initial parameters, $\hat{\theta}^{(t)} = (\hat{\theta}_A^{(t)}, \hat{\theta}_B^{(t)})$ determine for each of the five sets whether channel A or channel B was more likely to have posted the observed tweets (using the current parameter estimates). Then, assume these completions (guessed channel assignments) to be correct, and apply the regular maximum likelihood estimation

procedure to get $\hat{\theta}^{(t+1)}$. Finally, repeat these two steps until convergence. As the estimated model improves, so too will the quality of the resulting completions.

The expectation maximization algorithm is a refinement on this basic idea. Rather than picking the single most likely completion of the missing channel assignments on each iteration, the expectation maximization algorithm computes probabilities for each possible completion of the missing data, using the current parameters $\hat{\theta}^{(t)}$. These probabilities are used to create a weighted training set consisting of all possible completions of the data. A modified version of maximum likelihood estimation that deals with weighted training examples provides new parameter estimates, $\hat{\theta}^{(t+1)}$. By using weighted training examples rather than choosing the single best completion, the expectation maximization algorithm accounts for the confidence of the model in each completion of the data (fig 4).

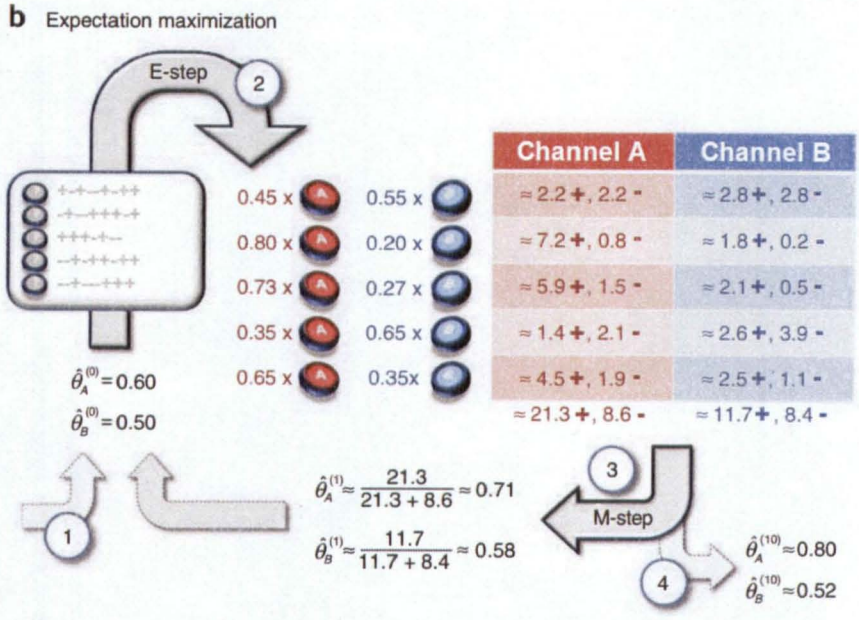

Figure 3. The incomplete case of the opinion tracking experiment.

However, the implementation of this model is not the exact incomplete case we are aiming for, thus we modify conditions of the previous incomplete case experiment as following. First, the sets of ten sentiment values are driven from the same tweet about multiple different 10 topics, which are defined apriori and constant among all tweets. Second, the probabilities computed in the expectation step according to the distributions are for all sets vertically in figure 3, since each set is now considered as one tweet. Third, the tweets analyzed are from anonymous users affected by multiple opinion sources. It is important to point that we are not concerned about the identity of the user; we are concerned about the source of the opinion which the sentiment is based upon. Lastly, the sentiment values used do not necessarily have to be trivial polarity (positive and negative); they could be sentiment groups or scores. Thus, in our model, the aim of the

expectation step is not to find a single value for $\hat{\theta}$, but is to fit a normal distribution onto the sentiment observed among the tweets. This means that the EM algorithm initially assumes all of the analyzed tweets are in one cluster with a normal distribution. Then the algorithm applies the maximum likelihood procedure to improve the assumed parameters, which could result into splitting the guessed cluster into two, and so on.

The expectation maximization algorithm alternates between the steps of guessing a probability distribution over completions of missing data given the current model (known as the E-step) and then re-estimating the model parameters using these completions (known as the M-step). The name 'E-step' comes from the fact that one does not usually need to form the probability distribution over completions explicitly, but rather need only compute 'expected' sufficient statistics over these completions. Similarly, the name 'M-step' comes from the fact that model re-estimation can be thought of as 'maximization' of the expected log-likelihood of the data. Introduced as early as 1955 by Ceppellini et al. in the context of gene frequency estimation, the expectation maximization algorithm was analyzed more generally by Hartley and by Baum et al. in the context of hidden Markov models, where it is commonly known as the Baum-Welch algorithm. The standard reference on the expectation maximization algorithm and its convergence is Dempster et al in 1977.

EM using Weka:

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from a Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. The software is available for free online, and installed on our fedora server at CAU.

The Weka explorer window is easy to use for importing data in ARFF format and applies several machine learning algorithms. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types followed by the data. The "Cluster" tab gives several options of clustering algorithms (i.e. Cobweb, DBScan, FarthestFirst, FilteredClusterer, etc.). However, we are concerned with using the EM algorithm.

Along with assigning the sentiment of each tweet we search for keywords relative to the news channels analyzed in the tweet. Thus, with each tweet we have information about which news channel is the tweet referring to. As discussed in the hypothesis section, it is very important to calculate the percentage of referrers in herds of opinions. This type of information is assigned as two nominal values $\{news, Nonews\}$. For example, if the tweet contains $news$ at the fox news column, but has $Nonews$ at the CNN news column, then this tweet has referred its opinion from fox news but not from

CNN. In this step we ignore the news refers, using the ignore attributes button, values for all news channels addressed, because we do not want these attributes play a role in the clustering algorithm. We only need these attributes to show us on the visual an approximate analysis of the percentage of referrals in diverse and herds of opinions.

In Weka, the clustering scheme generates probabilistic descriptions of the clusters in terms of mean and standard deviation for the numeric attributes and value counts (incremented by 1 and modified with a small value to avoid zero probabilities) - for the nominal ones. We investigate the mean and the standard deviation of each cluster in order to find the overlapping and the isolated clusters. In "Classes to clusters" evaluation mode this algorithm also outputs the log-likelihood, assigns classes to the clusters and prints the confusion matrix and the error rate. EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or you may specify apriori how many clusters to generate.

The cross validation performed to determine the number of clusters is done in the following steps:

1. The number of clusters is set to 1
2. The training set is split randomly into 10 folds
3. EM is performed 10 times using the 10 folds
4. The log likelihood is averaged over all 10 results

5. If log likelihood has increased the number of clusters is increased by 1 and the program continues at step 2

The number of folds is fixed to 10, as long as the number of instances in the training set is not smaller than 10. If this is the case the number of folds is set equal to the number of instances.[7]

[7] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/ (accessed February 15, 2014).

CHAPTER 4

RESULTS & DISCUSSION

Data Collection:

Under the administration of Professor Peter Molnar over 170 million tweets were harvested using a stream that was active since September 2012 to monitor the current political situation around the world. The Twitter project was established on the fedora server to grant the access to this database to the faculty and the students of CAU, and researchers affiliated with the institution. The website hosts all detailed information at the fedora website at.[1]

We chose the 140dev streaming API to store the tweets into our fedora using MySQL database. The 140dev API framework is a free source code library written by Adam Green[2] and released under the General Public License (GPL). The goal of this API is to provide a simple interface to the Twitter Streaming API. The current version provides a tweet aggregation database, and a plugin for tweet display on any Web page. However, Mr. Green is planning to provide plugins for data mining, automated tweeting and account management in the future. 140dev is written in PHP and JavaScript, and uses

[1]Peter Molnar. The Twitter Project. http://fedora.cis.cau.edu/~pmolnar/TWITTER/ (accessed February 15, 2014).

[2] Adam Green. 140Dev. http://140dev.com/ (accessed February 15, 2014).

the MySQL database for storage. Thus, all our extraction queries that we present in the thesis are in MySQL. All of the interactions between the modules in this framework are through the database, which means that additional modules can be written in any language that has a MySQL interface of 140dev. Additionally, for developers' interests, the API provides flexibility in expanding, which is one of the reasons for calling it a framework. The framework is composed of the database server and other plugins. The database server is the core module of the 140dev API. It uses the Twitter API to gather tweets for selected keywords and stores them in a MySQL database. In our relational database we have 10 tables connected together, which contains information about the users, tweets, tweet URLs, tags and mentions, mentions' counts, JSON cache, the degrees and their in and out. The rest of the libraries are built as plugins that share information with this database server. One of the important plugins that most advertising websites used to add Twitter widgets is the display plugin. The plugin calls the copy of the Twitter database server, retrieves the most recent tweets, and returns them as formatted HTML. All tweet entities are rendered as links. In order to monitor the political situation with respect to coverage of mainstream media, we chose particular terms to be used in streaming the tweets.

Statistical analysis:

Our statistical analysis on the percentage of mainstream media mentions among the total number of tweets was conducted for 10 million tweets. The following table shows news channels' names, keywords used for their search and their frequency:

Table 3.1. The number of tweets which mentioned the following news channels and the used keywords to search for them.

Channel's name	Search keywords	Counts
CNN	#cnn	24,354
ABC news	#abc/@abc/abc news/abcnews	23,100
Reuters	Reuters	22,896
NBC news	#nbc	18,426
Fox News	Foxnews/fox news	16,798
BBC	bbc	11,198
Associated Press	@ap/#apassociated press/associatedpress	10,963
NY Times	Nytimes/nytimes/newyorktimes/ny times/new york times	8,351
Washington Post	washington post/washingtonpost	6,178
USA Today	usa today/usatoday	7,879
Agence France-Presse	agence france presse/ agencefrancepress /afp	3,076
Forbes	forbes	2,981
bloomberg	bloomberg	1,981
Wall Street Journal	wallstreetjournal/wallstreet journal	1,484
TMZ	Tmz	1,134
Total		149,073 = 1.49%

Some search keywords mislead the counts of mentions as they might be simple components in normal words, for example, "ap" and "abc." By just using "ap" to count

the frequency of AP news mentions among the 10 million tweets the result was 475,951 tweets. However, normal words like "apple" or "appeal" contain the "ap" keyword, which means it that some tweets were false counted by only considering this simple combination of letters. Thus, we had to restrict the counts by combining with "#" and "@."

While table 3.2 shows the percentage of original tweets (not RT) versus the number of original that have links. This study helped us investigate the significance of sharing links among the users, which could be a door for another type of research question in the future work, for example, analyzing the links' web pages or documents to enhance the sentiment analysis of the tweet. Table 3.3 shows the percentages of tweets which have one adjective and more than one adjective in the same tweet out of 100,000 tweets.

Table 3.2. The percentage of original tweets and the original ones that have links

Number of analyzed tweets	**Original**	**Original & has link**
10,000,000	5,881,697 (58.8%)	2,719,402 (27.2%)

Table 3.3. The percentages of tweets which have one adjective and more than one adjective

Number of analyzed tweets	**One adjective**	**More than one adjective**
100,000	13,668 (13.7%)	6,103(6.1%)

We show the analysis settings of our experiments that produce the clusters which express the sources of opinions as hidden variables. We used different inputs and filtering categories for the finding the trending topics and the sentiment assignment steps, as was shown in figure 1 in the previous chapter. In this section we chow the categories of filtered used and the combination of different analysis settings. In the framework we apply different types of filtering categories. The filtering category depends on the property on which the tweets are filtered upon. In table 4.1 we summaries the category versus the property of filtering and the definition of property.

Table 4.1. Definitions of the filtering categories.

Property	Definition & Reasoning
RT	RTs are not the scope of our analysis, and considered as noisy data
News	Tweets which have at least one news channel mentioned
1-Topic	Tweets which have at least one topic mentioned
n-Topic	Tweets which have at least n topics mentioned
Adjective	Tweets which have only one adjective describing its sentiment

We filter out the RTs, unlike Myers Seth et al., since our scope is focused on finding the influence through comparing the sentiment of original tweets. Basically, it is worthless to analyze opinions which contain all zero vector, and that could result from either no adjective used or a trending topic mentioned. And thus, finding the frequent itemsets plays its role in reducing matrix sparsity, so when the sentiment is assigned to a topic we guarantee with high probability that the tweet would contain another topic. This is also the same reason, we use the adjective filter to decrease the sparsity of the

sentiment matrix used in the opinion clustering step. Nevertheless, we exclude the tweets which have more than one adjective, since we cannot handle multi-sentiment tweets. We do not apply any technique to differentiate the reference of each adjective in a multi-sentiment tweet. We also apply the one-topic filter to guarantee at least one topic mentioned per analyzed tweet, thus it is mandatory. However, it is not necessary to filter using n-topic filters.

Trending Topics:

In the trending topics step we apply the Apriori algorithm on two different groups of words: the most frequent general (not hashtags specifically) 30 words and on all hashtags. When collecting the tweets for both settings we filter the RTs out. However, the difference in application is due to the purpose of using the outputs of both settings. When we use all hashtags we are looking at the most frequent itemsets to be the trending topics. While when using the most general 30 words we look for the association rules between those 30 words and the news channels. The purpose is to use associated words to the news channels, in the future, to conduct validation analysis using the web archives of the news channels. The articles searched by the general frequent words would be compared with the tweets sentiment wise. We avoid using the hashtags since they are very particular to the tweeting behavior of the users, and many hash-tagged words are not usable for searching news archives.

Using hashtags:

We start with harvesting and filtering the tweets by RT category, then sort the hashtags to come up with counts shown in the previous list. As the Apriori algorithm's implementation was shown in the Methodology chapter, it uses numbers to index the hashtags for simplifying the input for the program, especially, because it is written in C++, appendix A. Lastly, we map the resulted indexes into the actual hashtags. The minimum support count was adjusted according to the average of the counts of all single hashtags. A very low minimum support count would result into a computationally expensive implementation and consider low frequent unimportant topics. While choosing a high support count would result into few hashtags and ignore important topics. Thus, we considered the average of all 1-frequent itemsets to be the minimum support count, which is 5,246.

The last frequent itemset contains 8 items and figure 4 shows the counts of all the frequent itemsets. Each hashtag in the frequent itemsets is a topic. Appendix B shows an image of all 621 frequent itemsets of sizes from three to eight. We ended up with the following list of 30 topics:

[obama, usa, tcot (Top Conservatives On Twitter), p2 (Progressive Propaganda), news, cnn, romney, teaparty, tiot (Top Independents On Twitter), usopen, dnc (Democratic National Committee), teamfollowback or tfb (you will follow back), economy, election, iran, israel, job, media, navy, nyc (New York City), ows (Occupy Wall Street), politics, twisters, usopen (Tennis Championship), vote, jakarta, london, politics, republican, fl(Fruity Loops studio)]

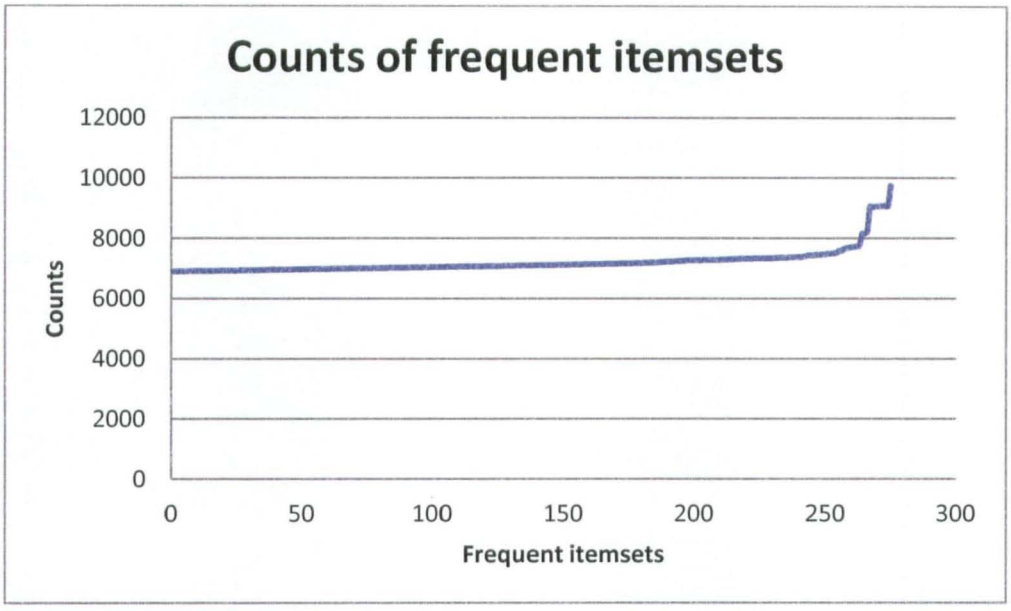

Figure 4. The Counts of the frequent itemsets.

The short hashtags which have political meaning are considered, and thus, we do not use hashtags to find association rules between topics and channels, since such short words could be misleading for the search engines. There are repeated entities being expressed by different hashtags like "mittromney" and "mitt". We combined those hashtags as the same by part of word searching, and the matches are recognized as the same entity. Thus, in the sentiment assignment step we use all possible hashtags that are

used to express the same topic or entity. We used the website[3] to define the short hashtags.

Association rules:

As mentioned in the head of this section, we are searching for association rules between general 30 frequent words and the news channels to be used in searching articles in the news web archives, where these articles in our future work will be compared with the sentiment of the tweets, as a validation schema. We found that 20 is the average count of 1-frequent itemsets, which lead to gaining at least 2 search keyword per channel. Table 4.2 summarizes those keywords and shows the calculation of their confidence.

The higher the confidence level of keywords that appear with a channel the more it is suitable to be used for searching in the web archive to find related articles from that particular channel. The percentages marked in red are the frequent itemsets chosen to be associated with the channels marked.

[3] #tagdef. http://tagdef.com/ (accessed February 15, 2014).

Table 4.2. Association rules between the channels and the most 30 frequent words and their confidence level.

Frequent itemsets	Support count with channel	Support count without channel	Confidence level
NBC:			
Romney, Obama	140	306	45.8%
Romney, Health	30	144	20.8%
Obama, Health	70	280	25%
Obama, Job	70	110	63.6%
NY Times:			
Romney, Job	100	533	5.3%
Romney, Taxes, Republican	20	63	31.7%
Romney, Gas, Employment	55	140	39.3%
Reuters:			
Obama, Mitt	497	514	96.7%
Romney, Obama, Job	222	306	72.5%
Fox:			
Romney, Elections	220	650	33.8%
Obama, Health	240	280	85%
ABC:			
Obama, Romney	30	306	9.8%
Romney, Economy	25	84	29.8%
CNN:			
Obama, Employment	55	70	78.6%
Romney, Taxes	20	63	31.7%

Observations & Inferences:

In this section we show our observations and the inferred meanings from the opinion clustering step through graphs and statistics calculated for each experiment setting using Weka. The original results from the scoring sentiment assignment method are shown first, and then we compare these results using the adjective hierarchy sentiment assignment method.

Weka Explorer provides a GUI to load data, preprocess it, and then apply various types of machine learning algorithms on the data. The Weka Explorer also provides the option of ignoring attributes and choosing the adequate evaluation settings. By using the "training set," this is the default evaluation choice; Weka classifies the training instances into clusters according to the cluster representation and computes the percentage of instances falling in each cluster after generating them. For probabilistic cluster representation, it is more suitable to evaluate clustering on a separate test dataset using "Supplied test set." This option provides loading a file or linking to a web page. The third and last method of evaluation in Weka is by assigning classes to clusters based on the majority value of the class attribute within each cluster. Then Weka computes the classification error, based on this assignment and also shows the corresponding confusion matrix. This option is done by choosing "Classes to clusters evaluation." Nevertheless, we use the default "training set" option, since we do not have separate test set available.

Experiment 1:

Here we show our observations of the opinion clustering step when using the scores list. The setting of the experiment is shown in figure 5 to view the applied filtering categories through the framework. We used the of 3-topics filter to restrict the sparsity of the matrix and obtained more valuable results.

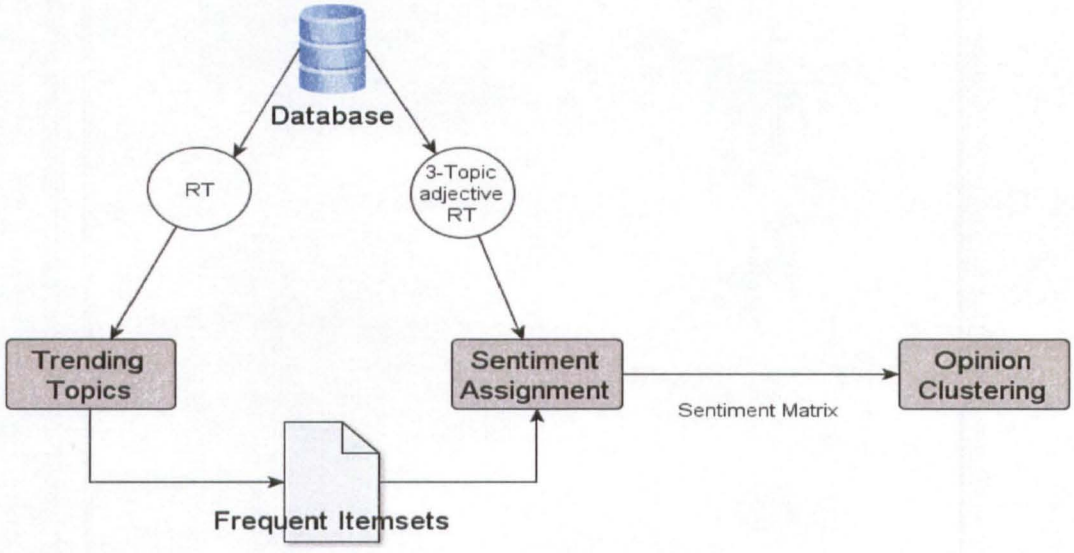

Figure 5. The category of filters applied through the framework for experiment 1.

The resulting overall clustered instances are distributed as shown in table 5.1, where 10 clusters were selected. There are no inferences that could be derived from that table; it just shows the distribution of instances among different clusters. We present the distribution of the sentiment towards each topic among the clusters using the mean and standard deviation in table 5.2.In this table we only shows the minimum and maximum of all clusters for topics which has isolated clusters, and mark those isolated clusters in red.

Table 5.1. The percentages of distributions of clusters for experiment 1.

Cluster number	Number of instances (Percentage)
0	306 (8%)
1	93 (2%)
2	43 (1%)
3	17 (0%)
4	42 (1%)
5	973 (26%)
6	1043 (28%)
7	639 (17%)
8	517 (14%)
9	73 (2%)

Table 5.2. The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation.

Topic	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
vote										
Min	-+0.5755	0.38775	-0.37505	-+0.5755	-+0.5755	-0.0709	-0.04865	-0.3767	-3.0915	-0.0657
Max		1.62805	0.87265			0.4919	0.08525	0.8063	-1.9705	0.104
Iran										
Min	-0.10085	-+0.114	-+0.114	-+0.114	-0.6271	-	0.2897	-+0.114	-+0.114	-+0.114
Max	0.15285				0.1497	+0.00005	1.7597			
Romney										
Min	-0.1171	-0.0746	-0.01165	-1.93635	-0.0915	1.7401	-0.29235	-0.2069	-1.2282	-+0.6682
Max	0.2707	0.1336	1.17005	-0.69725	0.1513	2.6301	0.15556	0.8105	0.13785	
Obama										
Min	1.27365	1.2679	0.39725	-2.25375	-2.8983	0.9315	0.7622	-0.3975	-3.17815	2.5845
Max	2.02195	2.5611	2.15075	-0.98165	-1.1273	2.2983	2.2398	1.5761	-2.01325	3.1151

From this table we can observe that cluster number 8 is an isolated cluster with respect to the "vote" topic, where the range of sentiment used by this cluster is between -1.9701 and -3.0915. The rest of the clusters express their sentiment out of this range. While for the topic "Iran" we can see that cluster number 6 is isolated from the rest at the range between 0.2897 and 1.7597. The same for topics "Romney" and "Obama," the clusters which exhibited isolation by not overlapping with other clusters, their minimums and maximums are marked in red. According to the table, in this sense it is obvious that topics "vote," "Iran," "Romney" and "Obama" have different segregated *unidimensional* opinions. Weka's visualizing tool show the segregation using the jitter option, which is quite unclear. Thus, for clearer representation about the isolated clusters figures 6-9 show simple error bar plots of the mean, minimum and maximum of sentiment scores for topics that show isolated clusters.

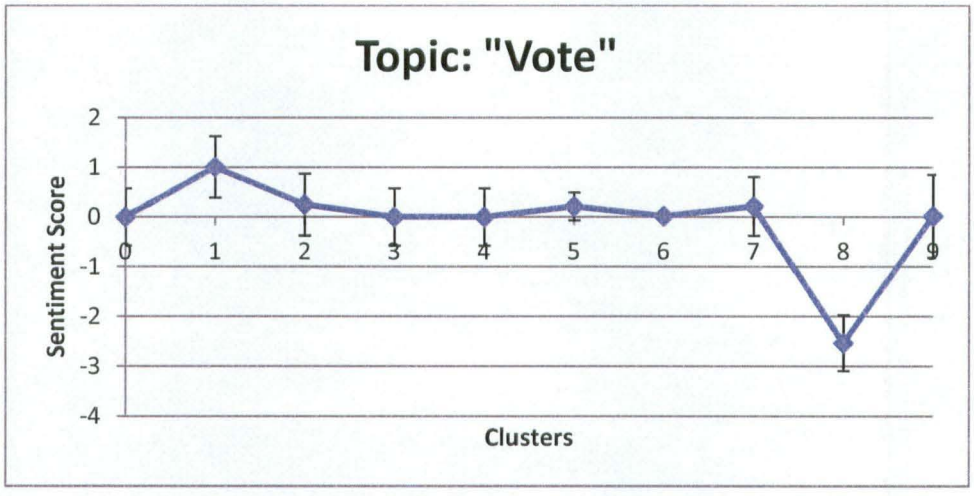

Figure 6. The distribution of clusters among the sentiment towards the topic "vote," where cluster 8 is the isolated cluster on an error bar plot using the minimum and maximum values.

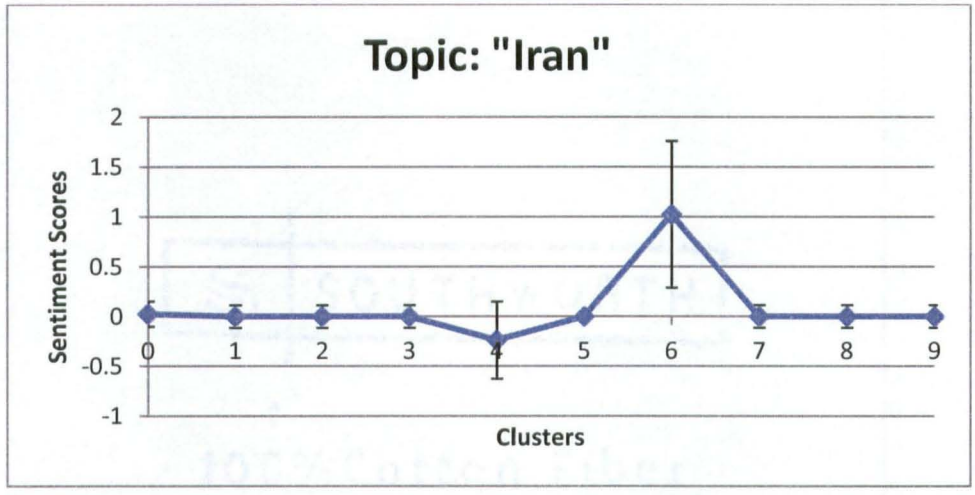

Figure 7. The distribution of clusters among the sentiment towards the topic "Iran," where cluster 6 is the isolated cluster on an error bar plot using the minimum and maximum values.

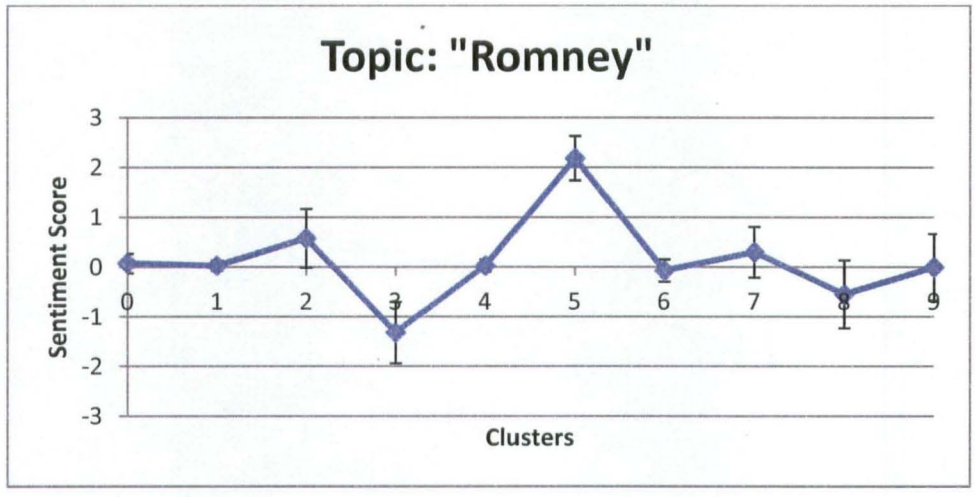

Figure 8. The distribution of clusters among the sentiment towards the topic "Romney," where cluster 5 is the isolated cluster on an error bar plot using the minimum and maximum values.

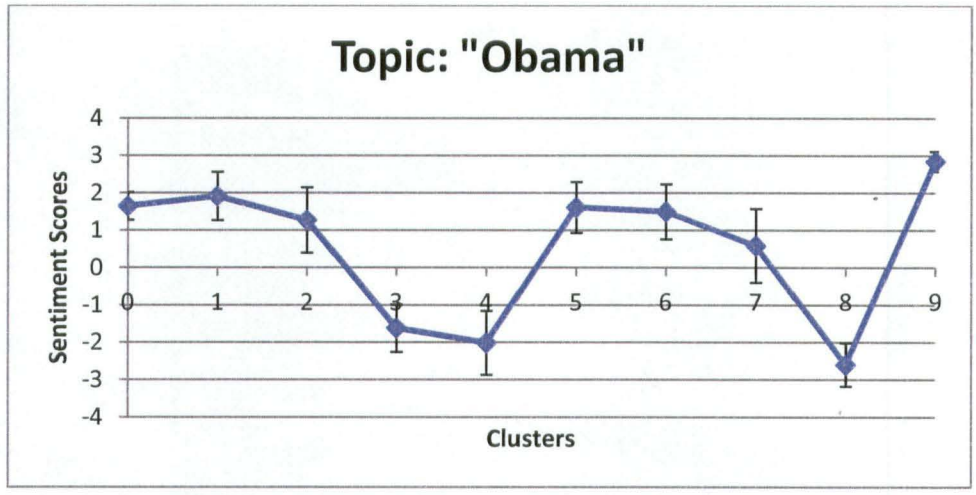

Figure 9. The distribution of clusters among the sentiment towards the topic "Obama," where cluster 9 is the isolated cluster on an error bar plot using the minimum and maximum values.

Table 5.3 shows the number of times each news channel was referred in the isolated clusters. The numbers are significantly low, and thus we apply the news filter to focus our analysis on the news channels only in the next subsection. As per our definition to the influence, at the beginning of the chapter, we categories the influence into two types: general and cluster specific influence. The general influence is the number of times and percentage of tweets that mentioned a channel from the total number of instances (from all clusters) with the biased sentiment. The cluster specific influence is also the number and percentage of tweets that mentioned a channel from the total number of instances with the biased sentiment, but in the isolated cluster only.

Table 5.3. The number of mentions for each channel in topics with isolated clusters.

Topic>>Sentiment	ABC	NY times	Fox	CNN	Reuters	NBC	Total
Vote < -1.9705							
General	1	0	1	5	0	1	3308
Cluster Specific(8)	1	0	0	0	0	1	3308
Iran > 0.2897							
General	1	0	1	5	0	1	5
Cluster Specific(6)	1	0	0	0	0	1	1
Romney> 1.7401							
General	1	0	1	10	0	3	2914
Cluster Specific(5)	1	0	0	3	0	2	2914
Obama > 2.5845							
General	4	3	6	46	0	12	3342
Cluster Specific(9)	2	0	2	13	0	4	2648

Experiment 2:

Another experiment setting, we filtered out tweets which have no news mention at all. However, in order to increase the number of tweets analyzed, we made the topic filter set at one only instead of three, as shown in figure 10. This setting has left for us 309 tweets only to be analyzed.

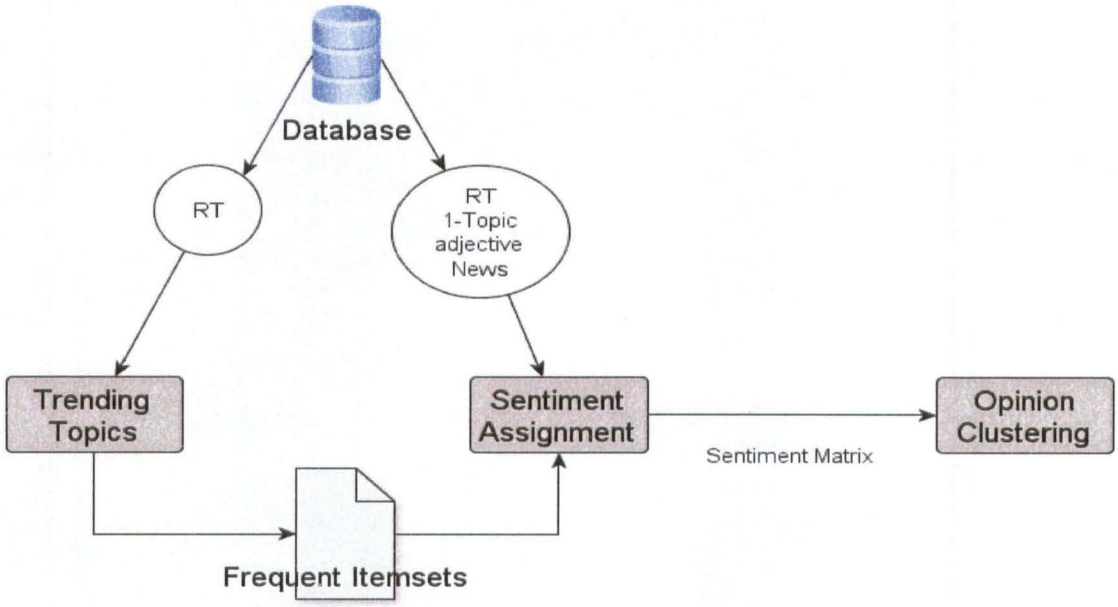

Figure 10. The category of filters applied through the framework for experiment 2.

Table 6.1. The percentages of distributions of clusters for experiment 2.

Cluster number	Number of instances (percentage)
0	56 (18%)
1	70 (23%)
2	75 (24%)
3	81 (26%)
4	27 (9%)

This setting has resulted in 5 clusters selected, which took Weka 10.43 seconds. Table 6.1 shows the distribution of the tweets among the clusters. We present the distribution of the sentiment towards each topic among the clusters using the mean and standard deviation in table 6.2. As we also red mark the clusters which express segregation from other clusters. In this table we only show the minimum and maximum

of all clusters for topics which has isolated clusters, and mark those isolated clusters in red.

Table 6.2. The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation for experiment 2.

Topic	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
OWS					
Min	-0.109	0.82365	-0.2798	0.0966	-0.38045
Max	0.221	1.80035	0.2798	0.5676	0.15225
Romney					
Min	0.3636	0.3211	-0.07445	0.68925	-1.8223
Max	1.6556	1.3571	1.57705	1.08635	-0.5339
Obama					
Min	1.36935	1.65555	-0.0416	0.8786	-2.75655
Max	2.58725	2.14185	1.7602	1.1206	-2.06565

From this table we can observe that cluster number 1 expresses segregation in opinion towards the Occupy Wall Street (OWS), where the range of sentiment used by this cluster is between 0.82365 and 1.80035. Cluster number 4 expressed segregation towards the topic "Romney," where the range of sentiment used by this cluster is between -1.8223 and -0.5339, which is not very far from other ranges of sentiment used by other clusters. Lastly, clusters number 0, 1 and 4 express interesting isolation in opinion. Cluster 0 and 1 are isolated together in the positive region between 1.36935 and 2.58725 and cluster 4 is isolated in the negative region between -2.75655 and -2.06565.

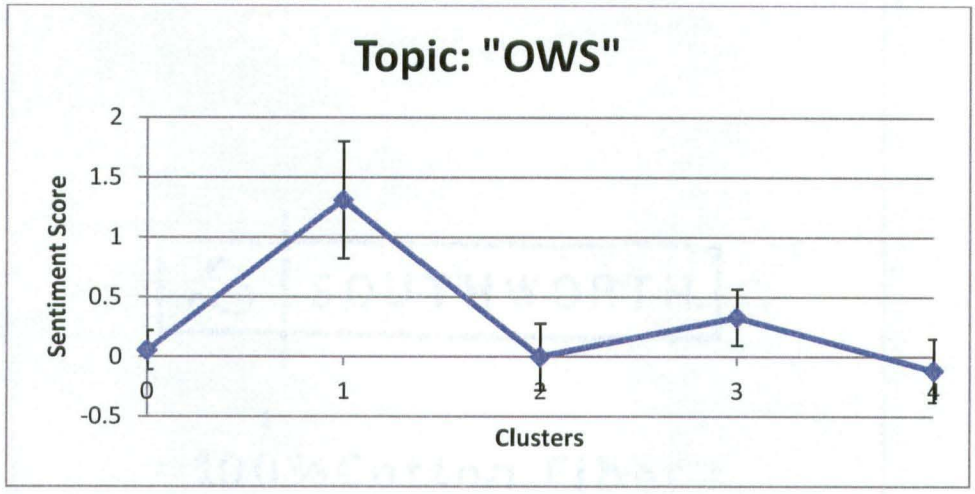

Figure 11. The distribution of clusters among the sentiment towards the topic "OWS," where cluster 1 is the isolated cluster on an error bar plot using the minimum and maximum values.

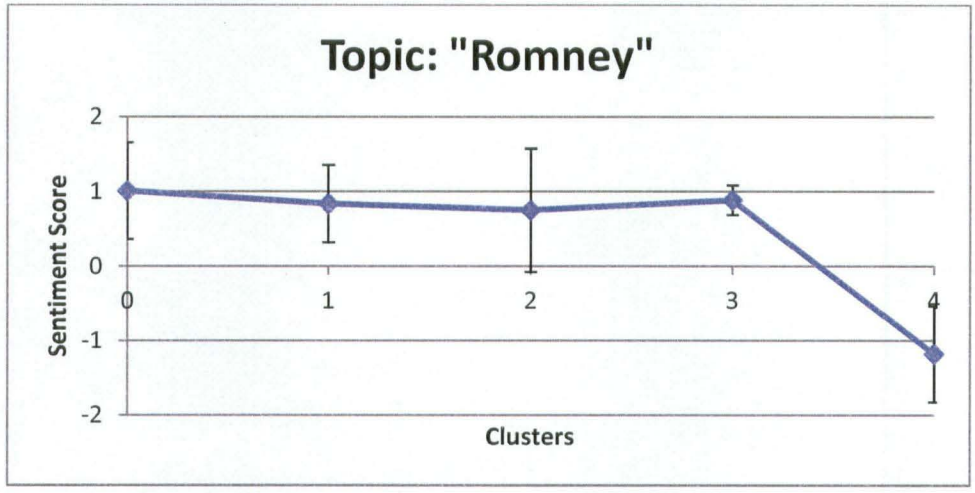

Figure 12. The distribution of clusters among the sentiment towards the topic "Romney," where cluster 4 is the isolated cluster on an error bar plot using the minimum and maximum values.

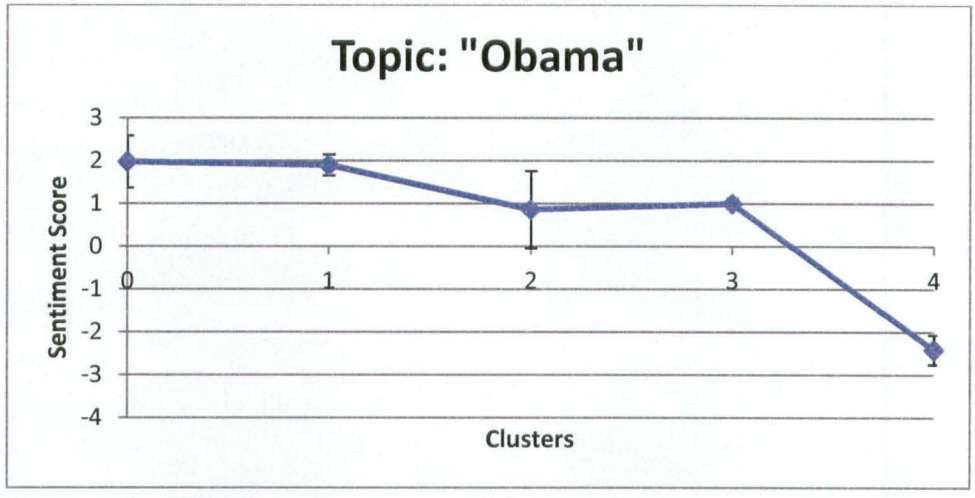

Figure 13. The distribution of clusters among the sentiment towards the topic "Obama," where cluster 4 is the isolated cluster and 0 and 1 are another two isolated clusters on an error bar plot using the minimum and maximum values.

According to the table, in this sense it is obvious that topics "OWS," "Romney" and "Obama" have different segregated *unidimensional* opinions. For clearer image about the isolated clusters figures 11-13 show simple error bar graph plots of the minimum and maximum for topics that show isolated clusters. Table 6.3 shows the number of instances and significant percentages of mentions in isolated clusters corresponding to the particular topic and sentiment that caused the isolation. The total frequency of shown at the most right column is not the sum of all channels' counts since some channels might be mentioned in the same tweet. Thus, we made a separate counter for counting the total.

Table 6.3. The number of mentions for each channel in topics with isolated clusters for experiment 2.

Topic>>Sentiment	ABC	NY times	Fox	CNN	Reuters	NBC	Total
OWS > 0.82365							
General	0	0	1	21	8	2	32 (10.3%)
Cluster Specific(1)	0	0	1	6	8	1	16 (5.1%)
Romney < -0.5339							
General	6	1	10	55	12	7	91 (29%)
Cluster Specific(4)	2	0	1	13	9	2	27 (8.7%)
Obama < -2							
General	12	3	19	105	17	19	175 (56.6%)
Cluster Specific(4)	5	0	2	29	11	10	57 (18.4%)
Obama > 1							
General	18	6	27	187	31	32	301(97%)
Cluster Specific (0&1)	5	0	2	29	11	10	57 (18.4%)

Experiment 3:

Here, we present the cluster distributions by probabilistic estimations on frequency counts among the sentiment groups. After detecting the isolated clusters, we calculate the percentage of news referrers out of these clusters in each topic to be compared with the percentages of referrers in isolated clusters used by the scoring method.

Using the same filters in experiment 1 but assigning the sentiment according to the sematic relatedness between adjectives, here we apply the 3-topic filter, without restrictions for the news reference category. This filtering process only kept 1,268 tweets to be analyzed. This setting has resulted in 8 clusters selected, which took Weka 90.28 seconds. Table 7.1 shows the distribution of the instances among the clusters. Cluster

number 3 is ignored by Weka, since it has less than one percent of tweets, and distributed uniformly among the sentiment (i.e. cluster 3 has 6 instances each expressing different sentiment using the 6 groups for all topics).

Table 7.1. The percentages of distributions of clusters for experiment 3.

Cluster number	Number of instances (percentage)
0	133 (10%)
1	242 (19%)
2	82 (6%)
3	6 (0%)
4	230 (18%)
5	105 (8%)
6	26 (2%)
7	52 (4%)
8	398 (31%)

We present the distribution of the clusters among the sentiment groups used for each topic in table 7.2. For each topic, we mark the highest probabilistic values of each cluster with green, and then we red mark the values which do not share common high sentiment concentrations with other clusters. This is the suitable method that we use for detecting isolated clusters to categorize them as segregated opinions. In Weka, EM uses discrete estimators for nominal attributes (just like naive Bayes does for classification). Weka's implementation of EM and naive Bayes assume that attributes are independent given the cluster/class. The numbers we see in the output for nominal attributes are

frequency counts (Laplace corrected). Since EM is a soft clusterer (i.e. each tweet belongs to each cluster probabilistically), the frequency counts can have fractional parts. We cannot compare those resulted clusters with the ones resulting from the scoring sentiment method. Both methods resulted into two different datasets, and this is the main reason preventing us from comparing both methods.

Table 7.2. The distribution of clusters among the sentiment towards each topic using the mean and the standard deviation for experiment 3.

Sentiment Group (0-5)	Cluster 0	Cluster 1	Cluster 2	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
Election								
0	114.1018	170.6996	80.5723	209.1856	1.148	1.1298	1.1466	358.0164
1	1.0009	71.7353	4.228	1	1.0128	1.0007	1.0221	1.0001
2	23.2457	1.0013	1.0005	26.5778	1.0148	23.766	48.3938	1.0001
3	1.0007	1.0008	1.0003	1	107.2516	1.0001	1.0139	39.7326
4	1.0085	1.0045	1.0033	1.0001	1.2109	1.0014	1.771	1.0003
5	1.002	1.9914	1.0008	1.0001	1.044	1.0067	1.955	1.0001
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
vote								
0	47.8256	182.6382	43.5889	130.4153	74.8974	23.8979	48.4482	227.2885
1	1.0006	57.7889	41.2099	1	1.0002	1	1.0003	1.0001
2	89.5182	1	1.0003	105.348	1	1.0035	1.1299	1.0001
3	1.0005	1.0001	1.0003	1	33.538	1	1.0009	170.4602
4	1.0085	1.0045	1.0033	1.0001	1.2109	1.0014	1.771	1.0003
5	1.0061	4.0011	1.0025	1.0002	1.0356	1.0019	1.9521	1.0005
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
Romney								
0	135.3035	36.8575	79.2217	5.873	33.5586	1.0043	49.9049	152.2764
1	1.0002	178.5672	5.4317	1.0002	1.0003	1.0003	1.0001	1.0001
2	1.918	1.0003	1.0001	229.8229	1.0008	23.886	1.372	1.0001
3	1.0003	1.0003	1.0001	1.0004	75.1122	1.0001	1.0001	238.8864
4	1.1353	1.0337	1.1502	1.0653	1.0016	1.0061	1.022	7.5858

5	1.0021	28.9739	1.0013	1.0019	1.0086	1.0081	1.0034	1.0007
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
Obama								
0	2.0701	60.3493	1.618	5.9348	106.1791	21.0801	15.68	2.0887
1	1.0034	156.0513	81.9417	1.0002	1.0003	1	1.0026	1.0004
2	134.8733	1.0012	1.0043	229.7572	1.0002	3.8181	35.5424	1.0033
3	1.0057	1.0008	1.0029	1.0006	2.4997	1	1.0008	388.4895
4	1.4008	1.0434	2.2352	1.0689	1.0026	1.0062	1.0764	8.1665
5	1.0061	27.9869	1.003	1.002	1.0001	1.0004	1.0004	1.0012
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
media								
0	114.5908	228.6994	81.2645	210.7115	105.6949	23.8981	13.7661	366.3746
1	1.0021	14.7024	3.294	1	1	1	1.0011	1.0002
2	22.4721	1	1.0002	25.0223	1.0002	1.001	36.5041	1
3	1.0005	1.0003	1.0002	1	2.9499	1	1.0087	29.0404
4	1.292	1.0212	1.2454	1.0298	1.0016	1.0053	1.0706	3.3342
5	1.002	1.0095	1.0007	1	1.0355	1.0004	1.9518	1
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495
republican								
0	131.0059	233.6567	9.659	220.3454	106.6877	23.9046	50.0323	343.7084
1	1.0002	9.7739	74.2258	1	1	1	1.0001	1
2	6.314	1	1.0011	15.4175	1	1.0001	1.2672	1
3	1.0002	1	1.0007	1	1.9942	1	1.0003	54.0046
4	1.0391	1.0023	1.9185	1.0007	1.0002	1	1.0027	1.0365
5	1	1	1	1	1	1	1	1
[total]	141.3594	247.4329	88.8051	239.7637	112.6821	28.9048	55.3024	401.7495

According to the table, in this sense it is obvious that topics "Elections," "Vote," "Media," "Romney," "Republicans" and "Obama" have different segregated *unidimensional* opinions. Cluster number 5 expresses high concentration of using

adjectives from group number 3, which is negatively biased group of adjectives, for the topic "elections." Cluster number 0 expresses a positive sentiment using group 2 towards the "vote" topic. Using the same sentiment group, cluster number 6 expresses its positive opinion towards the "Media" topic. Cluster number 1 expresses also a positive sentiment but using sentiment group 1 towards the topic "Romney." While cluster number 2 expresses its positive sentiment using sentiment group 1 towards the "Republican" topic. On the contrary, cluster number 7 expresses a negative sentiment towards the "Obama" topic using sentiment group 3. Table 7.3 shows the number and the percentages of influence tweets by each channel for both types of influences.

Table 7.3. The number of mentions for each channel in topics with isolated clusters for experiment 3.

Topic>>Sentiment	ABC	NY times	Fox	CNN	Reuters	NBC	Total
Media>>2							
General	2	1	0	4	0	0	7 (19.2%)
Cluster Specific	0	0	0	2	0	0	2 (5.5%)
Republicans>>1							
General	2	1	1	6	1	2	13 (17.6)
Cluster Specific	0	0	1	4	0	1	6 (8.1)
Obama>>3							
General	31	2	7	37	1	2	80 (20.6%)
Cluster Specific	29	1	7	35	0	1	73 (18.8%)

Our observation, the isolated cluster 8 is concentrated at the sentiment group 3, while a big portion comparatively to the rest of the clusters is referring to CNN. This table provides intuitive insight of the influence with the help of calculating the exact percentage to quantify the influence.

CHAPTER 5

CONCLUSION

Original contribution:

In summary, we proposed the challenge of measuring and quantifying the influence of mainstream media on Twitter users. The major assumptions for quantifying the measurement are based on the media social control theory, media bias theory and previous work done in defining the segregated opinions across the spectrum. The contribution towards this challenge is mainly about the framework and the model proposed. Basically, the framework proposed facilitated the basic input for our model, while the model is the main theme for detecting segregated opinions. The model depends totally on fitting the EM algorithm into finding the hidden variables, which are the sources of the opinions.

Methodology:

To test our framework and its model, we streamed-in tweets into our database on fedora and filtered the analyzed tweets according to three basic and two variable categories according to each experiment setting. We defined the trending topics as the frequent itemsets that are the output from the Apriori algorithm. The sentiment values where assigned using scores and semantic relatedness between adjectives used. The semantic relatedness is described through the hierarchical structure of adjectives, when

hierarchical clustering algorithm is applied on the lexicon dissimilarity matrix of the adjectives. The sentiment matrix is the output from the sentiment assignment step and the input for the opinion clustering step. The EM algorithm is applied on the sentiment matrix as the observed variables to find the hidden variables' parameters, the cluster parameters, which are the sources of the opinions. In order to characterize the anonymous sources of opinions we calculate the percentages of news mentions within all and the isolated clusters. We only consider the news mentions within the tweets which showed sentiment below the minimum or above the maximum of the isolated clusters' ranges.

Main findings:

In our three experiments, we used different setups of filtering categories, where two of them are similar in the used categories but different in the sentiment assignment. First, we filtered out the RTs to analyze original messages only, and the tweets which have no adjectives and/or less than three topics. The output result from this setup is 10 clusters which is the maximum number Weka could reach, since the training set is split randomly into 10 folds. The alternating EM process is applied 10 times maximum to increase the clusters by 1 incrementally each step starting from 1 cluster. However, the resulting isolated clusters showed insignificant percentage of tweets mentioning news channels. Thus, we change the 3 topic filter to be 1 and added the news filter, in order to focus on the tweets which mentioned the news channels only. For this setup, the isolated clusters showed significant percentage of tweets mentioning the news channels. Lastly,

we repeated the first setup but by assigning the sentiment using the semantic relatedness of adjectives. The isolated clusters also showed significant percentage in news mentions.

Future work:

We plan to use the association rules between the news channels and the most frequent 30 words in searching the web news archives for articles. By these keywords we optimize the finding of more articles related to twitter user's interests. We would apply the same sentiment analysis techniques on tweets and visualize them in comparison to the current results as a validation step. Additionally, a better idea is to visualize the sentiment versus time of these articles in comparison with the tweets, since we have the tweets' timestamps.

Disadvantages:

The disadvantage in our framework is the filtering of tweets which contain more than one adjective, since we were not able to differentiate the reference of adjectives to different nouns (topics) within the same tweet. However, in the future work we plan to use the NLTK to understand how can we differentiate between more than one adjective references using the sentence structure.

APPENDIX A

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <stdbool.h>
using namespace std;

//////////////////
//Global variables
//////////////////
struct node
{
	int pair, ctrNodes;
	struct node *next;
} node;

class Candidate
{
	public: int l;
	public: int sizeCand;
	public: int comb;
	public: int** Cand;
	public: int* suppCount;
	public: int* suggItemSet;
	public: void cand(int COMB, int SIZECAND)
	{
		sizeCand = SIZECAND;
		comb = COMB;
		Cand = (int**)malloc (comb*sizeof(int*));
		for(l=0; l<comb; l++)
			Cand[l] = (int*)malloc (sizeCand*sizeof(int));
		suppCount = (int*)malloc(comb*sizeof(int));
		for(l=0; l<comb; l++)
			suppCount[l]=0;
	}
	public: void deleteCand()
	{
		for(l=0; l<comb; l++)
		{
			free (Cand[l]);
		}
		free(suppCount);
	}
};

class Frequent
{
	public: int l;
	public: int sizeFreq;
	public: int comb;
	public: int** Freq;
	public: void freq(int COMB, int SIZEFREQ)
	{
		sizeFreq = SIZEFREQ;
		comb = COMB;
		Freq = (int**)malloc (comb*sizeof(int*));
		for(l=0; l<comb; l++)
			Freq[l] = (int*)malloc (sizeFreq*sizeof(int));
	}
	public: void deleteFreq()
	{
		for(l=0; l<comb; l++)
		{
			free (Freq[l]);
```

```
                }
        }
};

int** trans;
int colNum;
int recordNum;
int l;

//////////////////
//Global functions
//////////////////
int GetColNum(char fileName1[]);
int GetRecordNum(char fileName2[]);
void GetSourceFile(char fileName[],int recordNum,int colNum, int **trans);
int completeSugg(int* suggItemSet, int** Freq, int k, int K);
int tobeCand(int* suggItemSet, int** Freq, int comb, int k, int K);
int tobeFreq(int* suggItemSet, int** trans, int transNum, int K);

int main(int argc, char** argv) {

        //Variables
        int i,j,k, l, m, n, p,q, t, c;
        int f, tobeCandctr, tobeFreqctr, K, G;

//////////////////////
//Initiate Transaction
//////////////////////

        char inputFileName[] = {"trans.txt"};

        colNum = GetColNum(inputFileName);
        recordNum = GetRecordNum(inputFileName);
        int transNum = recordNum, itemNum = colNum, supp_Count = 5000;
        int *ctrItem=(int*)malloc (itemNum*sizeof(int));

        int **trans = (int**)malloc(recordNum*sizeof(int*));
        for(i=0; i<recordNum; i++)
        {
                trans[i] = (int*)malloc(colNum*sizeof(int));
        }
for(i=0; i<recordNum; i++)
{
        for(j=0; j<colNum; j++)
        {
                trans[i][j] = 0;
        }
}

        GetSourceFile(inputFileName, recordNum, colNum, trans);

        for(i=0; i<recordNum; i++)
        {
                for(j=0; j<colNum; j++)
                {
printf("%d, ", trans[i][j]);
                }
        }


////////////////////////////////////////////////
//1_FreqitemSet/////////////////////////////////
////////////////////////////////////////////////
K=1;
        Frequent Freq[itemNum];

        Freq[0].comb = 0;
        Freq[0].sizeFreq = K;
```

```
        for(i=0; i<itemNum; i++)
        {
                ctrItem[i] = 0;
                for(j=0; j<transNum; j++)
                {
                        ctrItem[i] += trans[j][i];
                }
                if(ctrItem[i]>=supp_Count)
                {
                        Freq[0].comb++;
                }
        }

Freq[0].freq(Freq[0].comb, Freq[0].sizeFreq);

        j=0;    //separate iterator for the Freq

        for(i=0; i<itemNum; i++)
        {
                if(ctrItem[i]>=supp_Count)
                        {
                                Freq[0].Freq[j][K-1] = i;
                                j++;
                        }
        }


////////////////////////////////////////////////
//2_FreqitemSet/////////////////////////////////
////////////////////////////////////////////////
K=2;
int tmpFlag=0;

        Freq[1].sizeFreq = K;
        Freq[1].comb = 0;

        for(p=0; p<Freq[0].comb; p++)
        {
                for(q=p+1; q<Freq[0].comb; q++)
                {
                        tmpFlag=0;
                        for(i=0; i<transNum; i++)
                        {
                                if(trans[i][Freq[0].Freq[p][0]]==1 &&
trans[i][Freq[0].Freq[q][0]]==1)
                                {
                                        tmpFlag++;
                                }       //end if trans==1
                        }       //end for i
                        if(tmpFlag>=supp_Count)
                                Freq[1].comb++;
                }       //end of q
        }       //end of p

        Freq[1].freq(Freq[1].comb, Freq[1].sizeFreq);
        Freq[1].comb=0;

        for(p=0; p<Freq[0].comb; p++)
        {
                for(q=p+1; q<Freq[0].comb; q++)
                {
                        tmpFlag=0;
                        for(i=0; i<transNum; i++)
                        {
                                if(trans[i][Freq[0].Freq[p][0]]==1 &&
trans[i][Freq[0].Freq[q][0]]==1)
```

```
                                {
                                        tmpFlag++;
                                }       //end if trans==1
                        }       //end for i
                        if(tmpFlag>=supp_Count)
                        {
                                Freq[1].Freq[Freq[1].comb][0] = Freq[0].Freq[p][0];
                                Freq[1].Freq[Freq[1].comb][1] = Freq[0].Freq[q][0];
printf("\n(%d)%d, %d cnts=%d", Freq[1].comb, Freq[1].Freq[Freq[1].comb][0]+1,
Freq[1].Freq[Freq[1].comb][1]+1, tmpFlag);
                                Freq[1].comb++;
                        }       //end of if tmpFlag==1
                }       //end of q
        }       //end of p

        Freq[0].deleteFreq();

////////////////
//Generic Driver
////////////////
        m = Freq[1].comb;
        while(m!=0)
        {

//////////////////////////////////////////////////////////////////////////////
//3_FreqitemSet as initial for generic number of Freq-itemsets//////////////////
//////////////////////////////////////////////////////////////////////////////
// We start with K=3
K++;
n=0;
        Freq[K-1].comb = m*20;
        Freq[K-1].sizeFreq = K;
        Freq[K-1].freq(Freq[K-1].comb, K);
        int* suggItemSet = (int*)malloc(K*sizeof(int));
m=0;

        for(i=0; i<Freq[K-2].comb; i++)       //Next Cand
        {
                for(j=0; j<K-1; j++)  //Initiate suggestion
                {
                        suggItemSet[j] = Freq[K-2].Freq[i][j];
                }
                for(k=i+1; k<Freq[K-2].comb; k++)           //Complete suggestion
                {
                        f = completeSugg(suggItemSet, Freq[K-2].Freq, k, K);
                        if(f==K-2)            //confirm suggestion
                        {
                                tobeCandctr = tobeCand (suggItemSet, Freq[K-2].Freq,
Freq[K-2].comb, k, K);
                                if(tobeCandctr==K-2)
                                {
                                        tobeFreqctr = tobeFreq(suggItemSet, trans, transNum,
K);

                                        if(tobeFreqctr>=supp_Count)   //Check support count
                                        {
                                                for(l=0; l<K; l++)
                                                {
                                                        Freq[K-1].Freq[m][l] =
suggItemSet[l];
printf("%d, ", Freq[K-1].Freq[m][l]+1);
                                                }
                                                m++;
printf("\tcnts=%d\n", tobeFreqctr);
                                        }
                                }       //end if tobeCandctr==K-2
                        }       //end if f==K-2
                }       //end for k
```

```
        }       //end for i

        Freq[K-2].deleteFreq();
        }       //end of while freqFlag (m)

}       //end of main


////////////////
//to be Frequent
////////////////

int tobeFreq(int* suggItemSet, int** trans, int transNum, int K)
{
        int j, k, p, f, supp=0;

        for(j=0; j<transNum; j++)
        {
                f=0;
                for(k=0; k<K; k++)
                {
                        f += trans[j][suggItemSet[k]]==1;

                }       //      end of k
                if(f==K)
                {
                        supp++;
                }
        }       //end of j

return supp;

}

/////////////////
//to be Candidate
/////////////////

int tobeCand(int* suggItemSet, int** Freq, int comb, int k, int K)
{
        int f, nextFreq, sugg, freq, tobeCandctr = 0;

        for(nextFreq=k+1; nextFreq<comb; nextFreq++)
        {
                f=0;
                for(sugg=0; sugg<K; sugg++)
                {
                        for(freq=0; freq<K-1; freq++)
                        {
                                if(suggItemSet[sugg] == Freq[nextFreq][freq])
                                {
                                        f++;
                                }       //end if sugg==Freq
                        }       //end for freq
                }       //end for sugg
                if(f==K-1)
                {
                        tobeCandctr++;
                }       //end if K-1
        }       //end for nextFreq
return tobeCandctr;
}


//////////////////////
//Complete Suggestion
//////////////////////
```

```
int completeSugg(int* suggItemSet, int** Freq, int k, int K)
{
	int sugg, freq, f=0, maySugg, i, flag=0;
	for(sugg=0; sugg<K-1; sugg++)
	{
		for(freq=0; freq<K-1; freq++)
		{
			if(suggItemSet[sugg]==Freq[k][freq])
			{
				f++;
			}	//end of if sugg==Freq
			else
			{
				maySugg = Freq[k][freq];
			}	//end of else sugg==Freq
		}	//end of freq
	}	//end of sugg

	if(f==K-2)
	{
		for(i=0; i<K-1; i++)
		{
			if(suggItemSet[i]==maySugg)
			{
				flag=1;
				f=0;
			}
		}
		if(flag!=1)
		{
			suggItemSet[K-1] = maySugg;
		}
	}	//end if f==K-2

return f;
}

////////////////////
//Get number of col
////////////////////
int GetColNum(char fileName1[])
{
	FILE *in = fopen (fileName1, "r");

	char ch;
	while(ch!='\n')
	{
		if(ch==',')
		{
			colNum++;
			ch=fgetc(in);
		}
		ch=fgetc(in);
	}
	//colNum++;
	fclose(in);
	return colNum;
}

////////////////////
//Get number of rows
////////////////////
int GetRecordNum(char fileName2[])
{
	FILE *in = fopen(fileName2, "r");

	char ch=(char)NULL;
	while(!feof(in))
```

```
	{
		if(ch=='\n')
		{
			recordNum++;
			ch=fgetc(in);
		}
		ch=fgetc(in);
	}

	fclose(in);
	return recordNum;
}

//////////////////////
//Get data from the file
//////////////////////
void GetSourceFile(char fileName[],int rNum,int cNum, int** trans)
{
	FILE *in = fopen(fileName, "r");

	char ch=0;
	int i=0,j=0, k, item, copy;

	while(i<rNum)
	{
		char tmp[100];
		int idx=0;
		ch=fgetc(in);
		while(j<colNum)
		{
			if(ch==',')
			{
				item++;
				tmp[idx]='\0';

				for(k=1; k<cNum; k++)
				{
					if(j==k)
					{
						trans[i][k-1] = (int)atof(tmp);
					}
				}

				j++;
				idx=0;
			}
			ch=fgetc(in);
			if(ch=='\n')
				{break;}
			tmp[idx]=ch;
			idx++;
		}
				j=0;
		item=0;
		i++;
	}
	fclose(in);
```

APPENDIX B

3-Frequent itemsets	3-Frequent itemsets
#p2 #usopen #election	#vote #catholic #2
#p2 #usopen #dnc2012	#dnc2012 #ows #joebidenbikergangs
#p2 #usopen #ows	#dnc2012 #ows #fl
#p2 #usopen #joebidenbikergangs	#dnc2012 #ows #orgulhosophiaabrahao
#p2 #usopen #fl	#dnc2012 #joebidenbikergangs #fl
#p2 #usopen #orgulhosophiaabrahao	#dnc2012 #joebidenbikergangs #orgulhosophiaabrahao
#p2 #election #dnc2012	#dnc2012 #fl #orgulhosophiaabrahao
#p2 #election #ows	#ows #joebidenbikergangs #fl
#p2 #election #joebidenbikergangs	#ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #election #fl	#ows #fl #orgulhosophiaabrahao
#p2 #election #orgulhosophiaabrahao	#mittromney #twisters #1
#p2 #dnc2012 #ows	#mittromney #twisters #tfb
#p2 #dnc2012 #joebidenbikergangs	#mittromney #twisters #2
#p2 #dnc2012 #fl	#mittromney #job #1
#p2 #dnc2012 #orgulhosophiaabrahao	#mittromney #job #nyc
#p2 #ows #joebidenbikergangs	#mittromney #job #nfl
#p2 #ows #fl	#mittromney #job #tiot
#p2 #ows #orgulhosophiaabrahao	#mittromney #job #navy
#p2 #mittromney #tiot	#mittromney #job #tfb
#p2 #mittromney #2	#mittromney #job #2
#p2 #1 #tiot	#mittromney #1 #nyc
#p2 #1 #2	#mittromney #1 #israel
#p2 #tiot #2	#mittromney #1 #romneyryan
#p2 #joebidenbikergangs #fl	#mittromney #1 #cnn
#p2 #joebidenbikergangs #orgulhosophiaabrahao	#mittromney #1 #tiot
#p2 #fl #orgulhosophiaabrahao	#mittromney #1 #navy
#news #mittromney #london2012	#mittromney #1 #vacation
#news #1 #london2012	#mittromney #1 #iran
#news #1 #tfb	#mittromney #1 #tfb
#news #1 #2	#mittromney #1 #2
#news #london2012 #2	#mittromney #1 #verdadeiramente
#obama2012 #usopen #tiot	#mittromney #nyc #nfl
#obama2012 #nobama #mittromney	#mittromney #nyc #tiot
#obama2012 #nobama #tiot	#mittromney #nyc #navy
#obama2012 #nobama #tfb	#mittromney #nyc #tfb
#obama2012 #vote #catholic	#mittromney #nyc #2
#obama2012 #vote #navy	#mittromney #israel #tiot
#obama2012 #forward2012 #mittromney	#mittromney #israel #tfb
#obama2012 #mittromney #job	#mittromney #israel #2
#obama2012 #mittromney #1	#mittromney #romneyryan #vacation
#obama2012 #mittromney #nyc	#mittromney #romneyryan #2
#obama2012 #mittromney #israel	#mittromney #jakarta #navy
#obama2012 #mittromney #cnn	#mittromney #cnn #tiot
#obama2012 #mittromney #london2012	#mittromney #cnn #iran
#obama2012 #mittromney #tiot	#mittromney #cnn #tfb
#obama2012 #mittromney #navy	#mittromney #cnn #2
#obama2012 #mittromney #tfb	#mittromney #mitt2012 #obama.
#obama2012 #mittromney #2	#mittromney #london2012 #tiot
#obama2012 #job #1	#mittromney #london2012 #navy
#obama2012 #job #nyc	#mittromney #london2012 #tfb
#obama2012 #job #tiot	#mittromney #london2012 #2
#obama2012 #job #navy	#mittromney #tiot #navy
#obama2012 #job #tfb	#mittromney #tiot #josÃ©victornossoeternoprincipe
#obama2012 #job #2	#mittromney #tiot #iran
#obama2012 #1 #nyc	#mittromney #tiot #tfb
#obama2012 #1 #israel	#mittromney #tiot #2
#obama2012 #1 #cnn	#mittromney #navy #tfb
#obama2012 #1 #tiot	#mittromney #navy #2
#obama2012 #1 #navy	#mittromney #vacation #2

#obama2012 #1 #tfb	#mittromney #iran #tfb
#obama2012 #1 #2	#mittromney #iran #2
#obama2012 #nyc #tiot	#mittromney #tfb #2
#obama2012 #nyc #navy	#mittromney #2 #verdadeiramente
#obama2012 #nyc #tfb	#twisters #1 #tiot
#obama2012 #nyc #2	#twisters #1 #music
#obama2012 #israel #tiot	#twisters #1 #tfb
#obama2012 #israel #tfb	#twisters #1 #2
#obama2012 #israel #2	#twisters #tiot #tfb
#obama2012 #romneyryan #navy	#twisters #tiot #2
#obama2012 #romneyryan #vacation	#twisters #music #tfb
#obama2012 #catholic #navy	#twisters #music #2
#obama2012 #cnn #tiot	#twisters #tfb #2
#obama2012 #cnn #tfb	#job #1 #nyc
#obama2012 #cnn #2	#job #1 #2
#obama2012 #london2012 #navy	#job #nyc #nfl
#obama2012 #tiot #navy	#job #nyc #tiot
#obama2012 #tiot #tfb	#job #nyc #navy
#obama2012 #tiot #2	#job #nyc #tfb
#obama2012 #navy #vacation	#job #nyc #2
#obama2012 #navy #topprog	#job #tiot #tfb
#obama2012 #navy #tfb	#1 #nyc #tfb
#obama2012 #navy #2	#1 #nyc #2
#obama2012 #tfb #2	#1 #israel #tiot
#usopen #election #dnc2012	#1 #israel #tfb
#usopen #election #ows	#1 #israel #2
#usopen #election #joebidenbikergangs	#1 #romneyryan #vacation
#usopen #election #fl	#1 #romneyryan #2
#usopen #election #orgulhosophiaabrahao	#1 #catholic #2
#usopen #dnc2012 #ows	#1 #cnn #tiot
#usopen #dnc2012 #joebidenbikergangs	#1 #cnn #iran
#usopen #dnc2012 #fl	#1 #cnn #tfb
#usopen #dnc2012 #orgulhosophiaabrahao	#1 #cnn #2
#usopen #ows #joebidenbikergangs	#1 #mitt2012 #tfb
#usopen #ows #fl	#1 #mitt2012 #obama.
#usopen #ows #orgulhosophiaabrahao	#1 #mitt2012 #2
#usopen #mittromney #tiot	#1 #london2012 #tfb
#usopen #tiot #navy	#1 #london2012 #2
#usopen #tiot #josÃ©victornossoeternoprincipe	#1 #tiot #navy
#usopen #joebidenbikergangs #fl	#1 #tiot #tfb
#usopen #joebidenbikergangs #orgulhosophiaabrahao	#1 #tiot #2
#usopen #fl #orgulhosophiaabrahao	#1 #navy #tfb
#election #dnc2012 #ows	#1 #navy #2
#election #dnc2012 #joebidenbikergangs	#1 #music #tfb
#election #dnc2012 #fl	#1 #music #2
#election #dnc2012 #orgulhosophiaabrahao	#1 #vacation #2
#election #ows #joebidenbikergangs	#1 #iran #tfb
#election #ows #fl	#1 #iran #2
#election #ows #orgulhosophiaabrahao	#1 #tfb #obama.
#election #joebidenbikergangs #fl	#1 #tfb #2
#election #joebidenbikergangs #orgulhosophiaabrahao	#1 #obama. #2
#election #fl #orgulhosophiaabrahao	#1 #2 #verdadeiramente
#teamfollowback #mittromney #cnn	#nyc #tiot #tfb
#teamfollowback #mittromney #tiot	#nyc #tfb #2
#teamfollowback #1 #tiot	#nyc #obama. #fl
#teamfollowback #cnn #tiot	#israel #mitt2012 #tfb
#teamfollowback #cnn #2	#israel #mitt2012 #obama.
#teamfollowback #tiot #2	#israel #tiot #tfb
#nobama #mittromney #1	#israel #tiot #2
#nobama #mittromney #israel	#israel #tfb #obama.
#nobama #mittromney #tiot	#israel #tfb #2
#nobama #mittromney #tfb	#romneyryan #navy #vacation
#nobama #mittromney #2	#romneyryan #vacation #2

#nobama #1 #israel	#jakarta #tiot #navy
#nobama #1 #tiot	#cnn #mitt2012 #obama.
#nobama #1 #tfb	#cnn #tiot #iran
#nobama #1 #2	#cnn #tiot #tfb
#nobama #israel #mitt2012	#cnn #tiot #obama.
#nobama #israel #tiot	#cnn #tiot #2
#nobama #israel #tfb	#cnn #iran #tfb
#nobama #israel #obama.	#cnn #iran #2
#nobama #israel #2	#cnn #tfb #2
#nobama #cnn #tiot	#mitt2012 #tiot #obama.
#nobama #mitt2012 #tfb	#mitt2012 #tfb #obama.
#nobama #mitt2012 #obama.	#mitt2012 #tfb #2
#nobama #tiot #tfb	#mitt2012 #obama. #2
#nobama #tiot #2	#london2012 #tfb #2
#nobama #tfb #obama.	#tiot #navy #tfb
#nobama #tfb #2	#tiot #navy #2
#vote #mittromney #catholic	#tiot #tfb #2
#vote #1 #catholic	#navy #tfb #2
#vote #1 #2	#joebidenbikergangs #fl #orgulhosophiaabrahao
#vote #catholic #tiot	#music #tfb #2
#vote #catholic #navy	#iran #tfb #2
#vote #catholic #tfb	#tfb #obama. #2

4-Frequent itemsets	4-Frequent itemsets
#p2 #usopen #election #dnc2012	#election #dnc2012 #ows #orgulhosophiaabrahao
#p2 #usopen #election #ows	#election #dnc2012 #joebidenbikergangs #fl
#p2 #usopen #election #joebidenbikergangs	#election #dnc2012 #joebidenbikergangs
#p2 #usopen #election #fl	#orgulhosophiaabrahao
#p2 #usopen #election #orgulhosophiaabrahao	#election #dnc2012 #fl #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #ows	#election #ows #joebidenbikergangs #fl
#p2 #usopen #dnc2012 #joebidenbikergangs	#election #ows #joebidenbikergangs
#p2 #usopen #dnc2012 #fl	#orgulhosophiaabrahao
#p2 #usopen #dnc2012 #orgulhosophiaabrahao	#election #ows #fl #orgulhosophiaabrahao
#p2 #usopen #ows #joebidenbikergangs	#election #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #usopen #ows #fl	#teamfollowback #mittromney #cnn #tiot
#p2 #usopen #ows #orgulhosophiaabrahao	#teamfollowback #cnn #tiot #2
#p2 #usopen #joebidenbikergangs #fl	#nobama #mittromney #1 #tiot
#p2 #usopen #joebidenbikergangs #orgulhosophiaabrahao	#nobama #mittromney #1 #tfb
#p2 #usopen #fl #orgulhosophiaabrahao	#nobama #mittromney #1 #2
#p2 #election #dnc2012 #ows	#nobama #mittromney #israel #tfb
#p2 #election #dnc2012 #joebidenbikergangs	#nobama #mittromney #tiot #tfb
#p2 #election #dnc2012 #fl	#nobama #mittromney #tiot #2
#p2 #election #dnc2012 #orgulhosophiaabrahao	#nobama #mittromney #tfb #2
#p2 #election #ows #joebidenbikergangs	#nobama #1 #israel #tfb
#p2 #election #ows #fl	#nobama #1 #israel #2
#p2 #election #ows #orgulhosophiaabrahao	#nobama #1 #tiot #tfb
#p2 #election #joebidenbikergangs #fl	#nobama #1 #tiot #2
#p2 #election #joebidenbikergangs	#nobama #1 #tfb #2
#orgulhosophiaabrahao	#nobama #israel #mitt2012 #tfb
#p2 #election #fl #orgulhosophiaabrahao	#nobama #israel #mitt2012 #obama.
#p2 #dnc2012 #ows #joebidenbikergangs	#nobama #israel #tiot #tfb
#p2 #dnc2012 #ows #fl	#nobama #israel #tfb #obama.
#p2 #dnc2012 #ows #orgulhosophiaabrahao	#nobama #israel #tfb #2
#p2 #dnc2012 #joebidenbikergangs #fl	#nobama #mitt2012 #tfb #obama.
#p2 #dnc2012 #joebidenbikergangs	#nobama #tiot #tfb #2
#orgulhosophiaabrahao	#vote #1 #catholic #2
#p2 #dnc2012 #fl #orgulhosophiaabrahao	#dnc2012 #ows #joebidenbikergangs #fl
#p2 #ows #joebidenbikergangs #fl	#dnc2012 #ows #joebidenbikergangs
#p2 #ows #joebidenbikergangs #orgulhosophiaabrahao	#orgulhosophiaabrahao
#p2 #ows #fl #orgulhosophiaabrahao	#dnc2012 #ows #fl #orgulhosophiaabrahao
#p2 #1 #tiot #2	#dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao

#p2 #joebidenbikergangs #fl #orgulhosophiaabrahao
#news #1 #london2012 #2
#obama2012 #nobama #mittromney #tiot
#obama2012 #vote #catholic #navy
#obama2012 #mittromney #job #1
#obama2012 #mittromney #job #nyc
#obama2012 #mittromney #job #tiot
#obama2012 #mittromney #job #navy
#obama2012 #mittromney #job #tfb
#obama2012 #mittromney #job #2
#obama2012 #mittromney #1 #nyc
#obama2012 #mittromney #1 #cnn
#obama2012 #mittromney #1 #tiot
#obama2012 #mittromney #1 #navy
#obama2012 #mittromney #1 #tfb
#obama2012 #mittromney #1 #2
#obama2012 #mittromney #nyc #tiot
#obama2012 #mittromney #nyc #navy
#obama2012 #mittromney #nyc #tfb
#obama2012 #mittromney #nyc #2
#obama2012 #mittromney #cnn #tiot
#obama2012 #mittromney #cnn #2
#obama2012 #mittromney #london2012 #navy
#obama2012 #mittromney #tiot #navy
#obama2012 #mittromney #tiot #tfb
#obama2012 #mittromney #tiot #2
#obama2012 #mittromney #navy #tfb
#obama2012 #mittromney #navy #2
#obama2012 #mittromney #tfb #2
#obama2012 #job #1 #nyc
#obama2012 #job #1 #2
#obama2012 #job #nyc #tiot
#obama2012 #job #nyc #navy
#obama2012 #job #nyc #tfb
#obama2012 #job #nyc #2
#obama2012 #1 #nyc #2
#obama2012 #1 #cnn #2
#obama2012 #1 #tiot #navy
#obama2012 #1 #tiot #tfb
#obama2012 #1 #tiot #2
#obama2012 #1 #navy #2
#obama2012 #1 #tfb #2
#obama2012 #romneyryan #navy #vacation
#obama2012 #cnn #tiot #2
#obama2012 #tiot #navy #tfb
#obama2012 #tiot #navy #2
#obama2012 #tiot #tfb #2
#usopen #election #dnc2012 #ows
#usopen #election #dnc2012 #joebidenbikergangs
#usopen #election #dnc2012 #fl
#usopen #election #dnc2012 #orgulhosophiaabrahao
#usopen #election #ows #joebidenbikergangs
#usopen #election #ows #fl
#usopen #election #ows #orgulhosophiaabrahao
#usopen #election #joebidenbikergangs #fl
#usopen #election #joebidenbikergangs
#orgulhosophiaabrahao
#usopen #election #fl #orgulhosophiaabrahao
#usopen #dnc2012 #ows #joebidenbikergangs
#usopen #dnc2012 #ows #fl
#usopen #dnc2012 #ows #orgulhosophiaabrahao
#usopen #dnc2012 #joebidenbikergangs #fl
#usopen #dnc2012 #joebidenbikergangs
#ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#mittromney #twisters #1 #tfb
#mittromney #twisters #1 #2
#mittromney #twisters #tfb #2
#mittromney #job #1 #nyc
#mittromney #job #1 #2
#mittromney #job #nyc #nfl
#mittromney #job #nyc #tiot
#mittromney #job #nyc #navy
#mittromney #job #nyc #tfb
#mittromney #job #nyc #2
#mittromney #1 #nyc #2
#mittromney #1 #israel #2
#mittromney #1 #romneyryan #vacation
#mittromney #1 #cnn #tiot
#mittromney #1 #cnn #iran
#mittromney #1 #cnn #tfb
#mittromney #1 #cnn #2
#mittromney #1 #tiot #tfb
#mittromney #1 #tiot #2
#mittromney #1 #navy #2
#mittromney #1 #iran #tfb
#mittromney #1 #iran #2
#mittromney #1 #tfb #2
#mittromney #1 #2 #verdadeiramente
#mittromney #nyc #tiot #tfb
#mittromney #israel #tiot #tfb
#mittromney #romneyryan #vacation #2
#mittromney #cnn #tiot #iran
#mittromney #cnn #tiot #tfb
#mittromney #cnn #tiot #2
#mittromney #cnn #iran #tfb
#mittromney #cnn #iran #2
#mittromney #cnn #tfb #2
#mittromney #tiot #navy #tfb
#mittromney #tiot #tfb #2
#mittromney #iran #tfb #2
#twisters #1 #tiot #tfb
#twisters #1 #tiot #2
#twisters #1 #music #tfb
#twisters #1 #music #2
#twisters #1 #tfb #2
#twisters #tiot #tfb #2
#twisters #music #tfb #2
#job #1 #nyc #2
#job #nyc #tiot #tfb
#1 #nyc #tfb #2
#1 #israel #tiot #2
#1 #israel #tfb #2
#1 #romneyryan #vacation #2
#1 #cnn #tiot #tfb
#1 #cnn #tiot #2
#1 #cnn #iran #tfb
#1 #cnn #iran #2
#1 #cnn #tfb #2
#1 #mitt2012 #tfb #obama.
#1 #mitt2012 #tfb #2
#1 #mitt2012 #obama. #2
#1 #london2012 #tfb #2
#1 #tiot #navy #2
#1 #tiot #tfb #2
#1 #navy #tfb #2
#1 #music #tfb #2

#orgulhosophiaabrahao	#1 #iran #tfb #2
#usopen #dnc2012 #fl #orgulhosophiaabrahao	#1 #tfb #obama. #2
#usopen #ows #joebidenbikergangs #fl	#israel #mitt2012 #tfb #obama.
#usopen #ows #joebidenbikergangs	#cnn #tiot #tfb #2
#orgulhosophiaabrahao	#cnn #iran #tfb #2
#usopen #ows #fl #orgulhosophiaabrahao	#mitt2012 #tfb #obama. #2
#usopen #joebidenbikergangs #fl #orgulhosophiaabrahao	
#election #dnc2012 #ows #joebidenbikergangs	
#election #dnc2012 #ows #fl	

5-Frequent itemsets
#p2 #usopen #election #dnc2012 #ows
#p2 #usopen #election #dnc2012 #joebidenbikergangs
#p2 #usopen #election #dnc2012 #fl
#p2 #usopen #election #dnc2012 #orgulhosophiaabrahao
#p2 #usopen #election #ows #joebidenbikergangs
#p2 #usopen #election #ows #fl
#p2 #usopen #election #ows #orgulhosophiaabrahao
#p2 #usopen #election #joebidenbikergangs #fl
#p2 #usopen #election #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #election #fl #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #ows #joebidenbikergangs
#p2 #usopen #dnc2012 #ows #fl
#p2 #usopen #dnc2012 #ows #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #joebidenbikergangs #fl
#p2 #usopen #dnc2012 #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #fl #orgulhosophiaabrahao
#p2 #usopen #ows #joebidenbikergangs #fl
#p2 #usopen #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #ows #fl #orgulhosophiaabrahao
#p2 #usopen #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #election #dnc2012 #ows #joebidenbikergangs
#p2 #election #dnc2012 #ows #fl
#p2 #election #dnc2012 #ows #orgulhosophiaabrahao
#p2 #election #dnc2012 #joebidenbikergangs #fl
#p2 #election #dnc2012 #joebidenbikergangs #orgulhosophiaabrahao
#p2 #election #dnc2012 #fl #orgulhosophiaabrahao
#p2 #election #ows #joebidenbikergangs #fl
#p2 #election #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #election #ows #fl #orgulhosophiaabrahao
#p2 #election #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #dnc2012 #ows #joebidenbikergangs #fl
#p2 #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #dnc2012 #ows #fl #orgulhosophiaabrahao
#p2 #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#obama2012 #mittromney #job #1 #nyc
#obama2012 #mittromney #job #1 #2
#obama2012 #mittromney #job #nyc #tiot
#obama2012 #mittromney #job #nyc #navy
#obama2012 #mittromney #job #nyc #tfb
#obama2012 #mittromney #job #nyc #2
#obama2012 #mittromney #1 #nyc #2
#obama2012 #mittromney #1 #cnn #2
#obama2012 #mittromney #1 #tiot #tfb
#obama2012 #mittromney #1 #tiot #2
#obama2012 #mittromney #1 #navy #2
#obama2012 #mittromney #1 #tfb #2
#obama2012 #mittromney #tiot #navy #tfb
#obama2012 #mittromney #tiot #tfb #2
#obama2012 #job #1 #nyc #2
#obama2012 #1 #tiot #navy #2

#obama2012 #1 #tiot #tfb #2
#usopen #election #dnc2012 #ows #joebidenbikergangs
#usopen #election #dnc2012 #ows #fl
#usopen #election #dnc2012 #ows #orgulhosophiaabrahao
#usopen #election #dnc2012 #joebidenbikergangs #fl
#usopen #election #dnc2012 #joebidenbikergangs #orgulhosophiaabrahao
#usopen #election #dnc2012 #fl #orgulhosophiaabrahao
#usopen #election #ows #joebidenbikergangs #fl
#usopen #election #ows #joebidenbikergangs #orgulhosophiaabrahao
#usopen #election #ows #fl #orgulhosophiaabrahao
#usopen #election #joebidenbikergangs #fl #orgulhosophiaabrahao
#usopen #dnc2012 #ows #joebidenbikergangs #fl
#usopen #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#usopen #dnc2012 #ows #fl #orgulhosophiaabrahao
#usopen #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#usopen #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#election #dnc2012 #ows #joebidenbikergangs #fl
#election #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#election #dnc2012 #ows #fl #orgulhosophiaabrahao
#election #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#election #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#nobama #mittromney #1 #tfb #2
#nobama #1 #tiot #tfb #2
#nobama #israel #mitt2012 #tfb #obama.
#dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#mittromney #twisters #1 #tfb #2
#mittromney #job #1 #nyc #2
#mittromney #1 #cnn #tiot #tfb
#mittromney #1 #cnn #tiot #2
#mittromney #1 #cnn #iran #tfb
#mittromney #1 #cnn #iran #2
#mittromney #1 #cnn #tfb #2
#mittromney #1 #tiot #tfb #2
#mittromney #1 #iran #tfb #2
#mittromney #cnn #tiot #tfb #2
#mittromney #cnn #iran #tfb #2
#twisters #1 #tiot #tfb #2
#twisters #1 #music #tfb #2
#1 #cnn #tiot #tfb #2
#1 #cnn #iran #tfb #2
#1 #mitt2012 #tfb #obama. #2

6-Frequent itemsets
#p2 #usopen #election #dnc2012 #ows #joebidenbikergangs
#p2 #usopen #election #dnc2012 #ows #fl
#p2 #usopen #election #dnc2012 #ows #orgulhosophiaabrahao
#p2 #usopen #election #dnc2012 #joebidenbikergangs #fl
#p2 #usopen #election #dnc2012 #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #election #dnc2012 #fl #orgulhosophiaabrahao
#p2 #usopen #election #ows #joebidenbikergangs #fl
#p2 #usopen #election #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #election #ows #fl #orgulhosophiaabrahao
#p2 #usopen #election #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #ows #joebidenbikergangs #fl
#p2 #usopen #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #ows #fl #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #usopen #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #election #dnc2012 #ows #joebidenbikergangs #fl
#p2 #election #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #election #dnc2012 #ows #fl #orgulhosophiaabrahao
#p2 #election #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao

#p2 #election #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#obama2012 #mittromney #job #1 #nyc #2
#obama2012 #mittromney #1 #tiot #tfb #2
#usopen #election #dnc2012 #ows #joebidenbikergangs #fl
#usopen #election #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#usopen #election #dnc2012 #ows #fl #orgulhosophiaabrahao
#usopen #election #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#usopen #election #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#usopen #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#election #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#mittromney #1 #cnn #tiot #tfb #2
#mittromney #1 #cnn #iran #tfb #2

7-Frequent itemsets
#p2 #usopen #election #dnc2012 #ows #joebidenbikergangs #fl
#p2 #usopen #election #dnc2012 #ows #joebidenbikergangs #orgulhosophiaabrahao
#p2 #usopen #election #dnc2012 #ows #fl #orgulhosophiaabrahao
#p2 #usopen #election #dnc2012 #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #usopen #election #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #usopen #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#p2 #election #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao
#usopen #election #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao

8-Frequent itemsets
#p2 #usopen #election #dnc2012 #ows #joebidenbikergangs #fl #orgulhosophiaabrahao

REFERENCES

Afolabi, I. T.; Musa, G. A.; Ayo, C. K.; Sofoluwe, A. B. Knowledge discovery in online repositories: a text mining approach. *European Journal of Scientific Research.* **2008**, *22*, 241-250.

Rakesh, A.; Srikant, R. Fast algorithms for mining association rules. *In Proceeding 20th international conference very large databases, VLDB.* **1994**, *1215*, 487-499.

Elif, A.; Allan, J. Sentiment diversification with different biases. *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, **2013**, 593-602.

Sophia, A.; Pyysalo, S.; Tsujii, J.; Kell, D. B. Event extraction for systems biology by text mining the literature. *Trends in biotechnology 28*, **2010**, *7*, 381-390.

Yates, B.; Ricardo; Moffat, A.; Navarro G. Searching large text collections. *In Handbook of massive data sets.* Springer US, **2002**, 195-243.

Baghela; Singh, V.; Tripathi, S. P. Text Mining Approaches To Extract Interesting Association Rules Association Rules from Text Documents Text Documents Text Documents. **2012**.

Becker, Hila; Naaman, M.; Gravano, L. Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM 11.* **2011,** 438-441.

Capon, B.; Trevor; Coenen, F.; Leng, P. An experiment in discovering association rules in the legal domain. *Proceedings 11th International Workshop on in Database and Expert Systems Applications.* **2000,** 1056-1060.

Bollen, J.; Mao, H.; Zeng, X. Twitter mood predicts the stock market. *Journal of Computational Science 2,* **2011,** *1,* 1-8.

Brill, E. A simple rule-based part of speech tagger. *Association for Computational Linguistics in Proceedings of the workshop on Speech and Natural Language.* **1992,** 112-116.

Cappelli, C. Identifying word senses from synonyms: a cluster analysis approach. *Quaderni di Statistica 5.* **2003,** 105-117.

Valentina, C.; Després, S. Text mining supported terminology construction. *In proceedings of the 5th International Conference on Knowledge Management.* Graz, Austria, **2005.**

Chen, Z.; Mukherjee, A.; Liu, B.; Hsu, M.; Castellanos, M.; Ghosh, R. Discovering coherent topics using general knowledge. *In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management,* ACM: **2013,** 209-218.

Chen, Z.; Liu, B.; Hsu, M.; Castellanos; Ghosh, R. Identifying Intention Posts in Discussion Forums. *In Proceedings of NAACL-HLT.* **2013**, 1041-1050.

Dai, Y.; Kakkonen, T.; Sutinen, E. MinEDec: a decision-support model that combines text-mining technologies with two competitive intelligence analysis methods. *International Journal of Computer Information Systems and Industrial Management Applications 3.* **2011,** 165-173.

Daille, B; Gaussier, E.; Langé, J. Towards automatic extraction of monolingual and bilingual terminology. *In Proceedings of the 15th conference on Computational linguistics*, Association for Computational Linguistics: **1994**, *1*, 515-521.

D'ALESSIO, D.; Allen, M. Selective exposure and dissonance after decisions. *Psychological Reports 91.* **2002**, *2*, 527-532.

De Haaff, M. Sentiment Analysis, Hard But Worth It! *CustomerThink.* **2010**, 2010-03-12.

Delgado, M.; Martín-Bautista M. J.; Sánchez, D.; Vila M. A. Mining text data: special features and patterns. *In Pattern Detection and Discovery.* Springer Berlin Heidelberg: **2002**, 140-153.

DeMarzo, P. M.; Vayanos, D.; Zwiebel, J. Persuasion bias, social influence, and unidimensional opinions. *The Quarterly Journal of Economics 118.* **2003**, *3*, 909-968.

Dempster, A. P.; Laird, N. M.; Rubin, Donald B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society 39.* **1977**, *1*, 1-38.

Dunham, M. H. Data Mining Introductory and Advanced Topics, 2003. **1997**.

Wu, F.; Weld, D. S.; *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* **2010**, 118-127.

Fei, G.; Mukherjee, A.; Liu, B.; Hsu, M,; Castellanos, M.; Ghosh, R. Exploiting Burstiness in Reviews for Review Spammer Detection. *In Seventh International AAAI Conference on Weblogs and Social Media.* **2013**.

Feldman, R.; Fresko M.; Kinar Y.; Lindell, Y.; Liphstat, O.; Rajman, M.; Schler, Y.; Zamir, O. Text mining at the term level. *In Principles of Data Mining and Knowledge Discovery.* Springer Berlin Heidelberg: **1998**, 65-73.

Festinger, L.; A theory of cognitive dissonance. Stanford university press: **1957**, *2*.

Gupta, V.; Lehal, G. S. A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence 1.* **2009**, *1*, 60-76.

Hall, S. Encoding and decoding in the television discourse. Centre for Cultural Studies, University of Birmingham: **1973**, *7*.

Hansen, L. K.; Arvidsson, A.; Nielsen F. A.; Colleoni, E.; Etter, M. Good friends, bad news-affect and virality in twitter. *In Future information technology.* Springer Berlin Heidelberg: **2011**, 34-43.

Herman, E. S.; Chomsky, N. Manufacturing consent: The political economy of the mass media. *Random House*. **2008**.

Hotho, A.; Nürnberger, A.; Paaß, G. A brief survey of text mining. *LDV Forum—GLDV Journal for Computational Linguistics and Language Technology 20.* **2005**, *1*, 19–62, ISSN 0175-1336.

Huang, C. J.; Liao, J. J.; Yang, D. X.; Chang, T. Y.; Luo, Y. C. Realization of a news dissemination agent based on weighted association rules and text mining techniques. *Expert Systems with Applications 37.* **2010**, *9*, 6409-6413.

Janetzko, D.; Cherfi, H.; Kennke, R.; Napoli, A.; Toussaint. Y. Knowledge-based selection of association rules for text mining. *In ECAI.* **2004**, *16*, 485.

Kim, S.; Li, F.; Lebanon, G; Essa, I. Beyond sentiment: The manifold of human emotions. arXiv preprint arXiv:2012, 1202-1568.

Kodratoff, Y. Knowledge discovery in texts: a definition, and applications. *In Foundations of Intelligent Systems*. Springer Berlin Heidelberg: **1999**, 16-29.

Kongthon, A. A text mining framework for discovering technological intelligence to support science and technology management. PhD diss., Georgia Institute of Technology, **2004**.

Kwak, H.; Lee, C.; Park, H.; Moon, S. What is Twitter, a social network or a news media? *In Proceedings of the 19th international conference on World wide web.* ACM: 2010, 591-600.

Landau, D; Feldman, R.; Aumann, Y.; Fresko, M.; Lindell, Y.; Lipshtat, O.; Zamir, O. Textvis: An integrated visual environment for text mining. *In Principles of Data Mining and Knowledge Discovery*. Springer Berlin Heidelberg: **1998**, 56-64.

Latiri, C. C.; Yahia, S. B. Textmining: Generating association rules from textual data. *In INFORSID*, **2001**, 27-39.

Lebart, L.; Salem, A.; Berry, L. Exploring textual data. Springer: **1998**, *4*.

Leopold, E.; Kindermann, J. Text categorization with support vector machines. How to represent texts in input space? *Machine Learning 46*, **2002**, *3*, 423-444.

Lewis, D. D. An evaluation of phrasal and clustered representations on a text categorization task. *In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM: **1992**, 37-50.

Lourenço, A.; Conover, M.; Wong, A.; Pan, F.; Abi-Haidar, A.; Nematzadeh, A.; Shatkay, H.; Rocha, L. M. Testing extensive use of NER tools in article classification and a statistical approach for method interaction extraction in the protein-protein interaction literature. **2010**.

Lu, H.; Feng, L.; Han, J. Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. *ACM Transactions on Information Systems (TOIS) 18*. 2000, *4*, 423-454.

Lumezanu, C.; Feamster, N.; Klein, H. # bias: Measuring the Tweeting Behavior of Propagandists. *In Sixth International AAAI Conference on Weblogs and Social Media*. **2012**.

Mahgoub, H.; Rösner, D.; Ismail, N.; Torkey, F. A Text Mining Technique Using Association Rules Extraction. *International journal of computational intelligence 4*. **2008**, *1*.

Mahgoub, H. Mining Association Rules from Unstructured Documents. *Enformatika 14* **2006**.

Manne, S.; Fatima, S. S. A Novel Approach for Text Categorization of Unorganized data based with Information Extraction. *International Journal on Computer Science & Engineering 3*. **2011**, *7*.

Manning, C. D. Foundations of statistical natural language processing. Edited by Hinrich Schütze. MIT press: **1999**.

Matos, P. F.; Lombardi, L. O.; Pardo, T. A.; Ciferri, C. D.; Vieira, M. T.; Ciferri, R. R. An environment for data analysis in biomedical domain: information extraction for decision support systems. *In Trends in Applied Intelligent Systems*. Springer Berlin Heidelberg: **2010**, 306-316.

Mooney, R. J.; Bunescu, R. Mining knowledge from text using information extraction. *ACM SIGKDD explorations newsletter 7*. **2005**, *1*, 3-10.

Mosley J.; Roosevelt, C. Social media analytics: Data mining applied to insurance Twitter posts. *In Casualty Actuarial Society E-Forum.* **Winter 2012**, 2, *1*.

Mukherjee, A.; Liu, B. Mining contentions from discussions and debates. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM: **2012**, 841-849.

Myers, S. A.; Zhu, C.; Leskovec, J. Information diffusion and external influence in networks. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM: **2012**, 33-41.

Naaman, M.; Becker, H.; Gravano, L. Hip and trendy: Characterizing emerging trends on Twitter. *Journal of the American Society for Information Science and Technology 62*. **2011**, *5*, 902-918.

Nasukawa, T; Nagano, T. Text analysis and knowledge mining system. *IBM systems journal 40*. **2011**, *4*, 967-984.

Rajman, M.; Besançon, R. Text mining-knowledge extraction from unstructured textual data. *In Advances in Data Science and Classification*. Springer Berlin Heidelberg: **1998**, 473-480.

Roddick, J, F.; Spiliopoulou, M. A survey of temporal knowledge discovery paradigms and methods. *Knowledge and Data Engineering IEEE Transactions on 14*. **2002**, *4*, 750-767.

Barbosa, R.; Angélica, G.; Silva, I. S.; Zaki, M.; Meira, W.; Prates, R. O.; Veloso, A. Characterizing the effectiveness of twitter hashtags to detect and track online population sentiment. *In Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts.* ACM: **2012**, 2621-2626.

Ruan, Y.; Purohit, H.; Fuhry, D.; Parthasarathy, S.; Sheth, A. Prediction of topic volume on twitter. *WebSci (short papers).* **2012**.

Sebastiani, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR) 34.* **2002**, *1*, 1-47.

Shatkay, H.; Feldman, R. Mining the biomedical literature in the genomic era: an overview. *Journal of computational biology 10.* **2003**, *6*, 821-855.

Tan, A. Text mining: The state of the art and the challenges. *In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases.* **1999**, 65-70.

Tan, P.N.; Kumar, V.; Srivastava, J. Selecting the right interestingness measure for association patterns. *In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM: **2002**, 32-41.

Tung, A.; Lu, H.; Han, J.; Feng, L. Efficient mining of intertransaction association rules. *Knowledge and Data Engineering, IEEE Transactions on 15.* **2003**, *1*, 43-56.

Wang, C.; Danilevsky, M.; Liu, J.; Desai, N.; Ji, H.; Han, J. Constructing Topical Hierarchies in Heterogeneous Information Networks.

Wang, J.; Yu, C. T.; Yu, P. S.; Liu, B.; Meng, W. Diversionary comments under political blog posts. *In Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM: **2012**, 1789-1793.

Wei, Z.; He, Y.; Gao, W.; Li, Zhou, L.; Wong, K. Mainstream media behavior analysis on Twitter: a case study on UK general election. *In Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM: **2013**, 174-178.

Wiebe, J.; Breck, E.; Buckley, C.; Cardie, C.; Davis, P.; Fraser, B.; Litman D. J. et al. Recognizing and Organizing Opinions Expressed in the World Press. *In New Directions in Question Answering*. **2003**, 12-19.

Wong, P. C.; Whitney, P.; Thomas, J. Visualizing association rules for text mining. *In Information Visualization, (Info Vis' 99) Proceedings IEEE Symposium*. IEEE: **1999**, 120-123.

Wu, S.; Li, Y.; Xu, Y. Deploying approaches for pattern refinement in text mining. *In Data Mining, 2006, (ICDM'06) Sixth International Conference*. IEEE: **2006**, 1157-1161.

Yang, X.; Ghoting, A.; Ruan, Y.; Parthasarathy, S. A framework for summarizing and analyzing twitter feeds. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM: **2012**, 370-378.

Yu, L.; Chan, C.; Lin, C.; Lin, I. Mining association language patterns using a distributional semantic model for negative life event classification. *Journal of biomedical informatics 44*. **2011**, *4*, 509-518.

Zhang, C.; Zhang, S. Association rule mining: models and algorithms. Springer-Verlag: **2002**.

Zhang, L.; Liu, B. Aspect and Entity Extraction for Opinion Mining. *In Data Mining and Knowledge Discovery for Big Data*. Springer Berlin Heidelberg: **2014**, 1-40.

Zhao, Q.; Bhowmick, S. S. Association rule mining: A survey. Nanyang Technological University, Singapore, **2003**.