THEME :

«**Analyzing Sentiment Bipolarity Phenomenon in Social Networks using Python**»

ΘΕΜΑ :

«Ανάλυση του φαινομένου της Συναισθηματικής Διπολικότητας σε Δίκτυα Κοινωνικής Δικτύωσης με τη χρήση Python»

A Thesis:
Submitted for partial fulfillment for Master's Degree in Business Analytics and Data Science to the department of Business Administration, University of Macedonia

By:

Chrysoula Paschalidou

Supervisor:

Dr. Georgiou Andreas Professor Department of Business Administration

Examiners:

Dr. Moschidis Odisseas Professor Department of Business Administration

Dr. Kaparis Konstantinos Assistant Professor Department of Business Administration

Declaration by student

I, Chrysoula Paschalidou, hereby declare that the work presented herein is original work done by me and has not been published or submitted elsewhere for the requirement of a degree programme. Any literature date or work done by other and cited within this thesis has given due acknowledgement and listed in the reference section.

Chrysoula Paschalidou

Place: University of Macedonia

Date: 26/08/2021

Summary

This study attempts to investigate the effect of sentiment interactions between users of a social network and diffusion of ideas. In addition, the study focused on identifying and analyzing bipolarity and controversy within a network in Twitter. Controversy is defined to be the phenomenon where users within a network are responding with opposing ideas to another's user post.

Data mining techniques implemented for constructing a network consisting of 62,799 single users and 64,693 connections between them. The network was representing all Tweets, Replies and Retweets posted in Twitter under hashtag "#Trump" for a given time in August 2020. From there, sentiment analysis was conducted for all posts by dividing users in two different polars, that is pro and against Trump candidacy, as well as dividing connections in controvert and assenting replies. A graphical visualization analysis followed, to draw relationships between influencing activity and polarity in our system.

The Insights from graph analysis implied the strong relation between controversy in the network and diffusion of information. It appeared that almost 20% of network communications occurred after a controvert reply against another post was made. In addition, 3,442 homogenous communities with strong ties appeared in the network, each of them representing a group of users with the same sentiment towards Trump candidacy. It was also revealed that strong connected communities belonging to different polars are more probably to form communications between them showing controversy. Another interesting finding reported from graph analysis was the tendency of users polarized pro Trump candidacy to show homophony between their communications, while the other polar showed more controversy within users' opinions.

Keywords:
Social Network, Sentiment Analysis, Machine Learning, BERT Model, TensorFlow, Keras, Bipolarity, Diffusion, Controversy, Polars, Gephi, Graph Visualizations, Python, Aspect-based Sentiment Analysis, Twitter, data extraction, political text analysis, Force Atlas Algorithm.

Περίληψη

Η μελέτη προσπαθεί να διερευνήσει τις συναισθηματικές επιδράσεις μεταξύ των χρηστών ενός κοινωνικού δικτύου και τη διάχυση ιδεών που απορρέουν σε αυτό. Επιπλέον, η μελέτη επικεντρώθηκε στον εντοπισμό και την ανάλυση της διπολικότητας, αλλά και της σύγκρουσης απόψεων στο δίκτυο του Twitter. Η σύγκουση απόψεων ορίζεται ως το φαινόμενο όπου χρήστες του συγκεκριμένου δικτύου ανταποκρίνονται με καινούριες αναρτήσεις έναντι άλλων χρηστών, προβάλοντας διαφορετική γνώμη για ένα συγκεκριμένο θέμα συζήτησης.

Τεχνικές εξόρυξης δεδομένων εφαρμόστηκαν για τη δημιουργία ενός δικτύου αποτελούμενο από 62,799 μοναδικούς χρήστες και 64,693 αλληλεπιδράσεις μεταξύ τους. Το δίκτυο αυτό αντικατοπτρίζει όλα τα Tweets, Replies, και Retweets που αναρτήθηκαν στον ιστοχώρο του Twitter με hashtag "#Trump" για μία συγκεκριμένη χρονική περίοδο τον Αύγουστο του 2020. Στα δεδομένα που εξορύχθηκαν πραγματοποιήθηκε ανάλυση συναισθήματος με αποτέλεσμα, αρχικά το διαχωρισμό των χρηστών σε δύο πόλους υπέρ και κατά της υποψηφιότητας Τραμπ, κατά δεύτερον τη διαίρεση των αλληλεπιδράσεων τους σε αντιθετικές και συναινετικές. Ακολούθησε η γραφική εναπαράσταση και η ανάλυση των σχέσεων μεταξύ των χρηστών με σκοπό την επεικόνιση της πολικότητας του συστήματος, την αντιμαχία μεταξύ των δύο πόλων και την σύνδεσης των δύο αυτών φαινομένων με τη διάχυση της πληροφορίας στο σύνολο του συστήματος και την υπεραξία του.

Τα αποτελέσματα της γραφικής ανάλυσης καταδεικνύουν την ισχυρή σχέση μεταξύ της σύγκρουσης των απόψεων μεταξύ των χρηστών και την αυξημένη διάχυση της πληροφορίας στο δίκτυο. Αποδείχθηκε ότι το 20% των αναρτήσεων στο δίκτυο εμφανίστηκαν έπειτα από μία αντιθετική απάντησε ενός από τους χρήστες σε κάποιο Tweet. Επιπλέον, 3,442 ομογενείς υποκοινότητες με ισχυρούς δεσμούς εμφανίζονται στο δίκτυο, στις υποκοινότητες διαφένεται πως οι απόψεις των χρηστών συγκλίνουν υποστηρίζοντας την ίδια άποψη για την υποψηφιότητα του αμερικάνου προέδρου. Οι χρήστες που ανήκουν σε υποκοινότητες που αντιπροσωπεύουν διαφορετική πολιτική αποδείχθηκε να αλληλεπιδρούν με αρνητικό συναίσθημα, διαφονώντας μεταξύ τους στη πλατφόρμα του Twitter. Τέλος από την ανάλυση συμπεραίνεται ότι οι χρήτες που συγκεντρώνονται υπέρ της υποψηφιότητας Τραμπ τείνουν να συμφωνούν της ομόιγυρης τους στο δίκτυο, καθώς στις αλληλεπιδράσεις τους υπάρχει ομοιογένεια απόψεων. Αντίθετα στο δεύτερο πόλο παρατηρείται μεγαλύτερη ασυμφωνία μεταξύ των χρηστών, με αποτέλεσμα να δημιουργούνται εκτενέστεροι διάλογοι μέσα στο σύστημα και να διαχέεται περισσότερη πληροφορία συνολικά στο δίκτυο.

Λέξεις Κλειδιά :

Κοινωνικό Δίκτυο, Εξόρυξη Δεδομένων, Ανάλυση Συναισθήματος, Μηχανική Μάθηση, Διπολικότητα, Διάχυση Πληροφορίας, BERT Μοντέλο, TensorFlow, Keras, Gephi, Python, Force Atlas Algorithm, Γραφική Ανάλυση, Aspect-based Sentiment, Twitter, Ανάλυση πολιτικού κειμένου.

Introduction

Social media have entered our lives into the last decades with the users having a more and more extensive use the passing years. This abnormal change on social reality has massively affected our interactions, the way we communicate with the others and mostly the way we conceive new ideas and perspectives. All users connected within a social network are forming a system of interactions, subject to the rules and limitations of the service provider. Twitter has proved to be a platform with the least limitations when it comes to connectivity and reach, since any user could follow and reply to all users within the network independently how “far” they might be in terms of shared connections. This ability has proved to be useful for network growth and its increasing value, the more connections created within a network, Tweets, Replies, Retweets and Reactions are produced, the more information is being propagated altogether. In such spread networks sub-communities appear between users with strong ties, sharing the same values and opinions. When it comes to a social event that concentrates high attention from users, often those interconnected sub-communities are forming different polars of interest. This is the phenomena of bipolarity, where users supporting different opinions are being divided in homogenous groups supporting the same ideas. Communications between users committed to different polars tend to show negative sentiment, with the users posting conflicted messages to other’s Tweets. Controversy of ideas within a network appeared to be a turning point for more discussion to be developed and being diffused in the network.

Motivation

In our days exerted use of social media, and the new freedoms that eludes, has impacted sincerely the objective truth one perceives about politics and social aspects of his/her life. The thin line between what is real, and fake has been translated to what majority is embracing as true. This realization has driven global interests into manipulating election results in USA, in 2010's, with the scandal of Cambridge Analytica, Brexit in England, the gilets jaunes protesters in France, or enhancing government oppression via Twitter in Saudi Arabia. Social bots have been orchestrating global discussions on the web, creating the illusion there is traction pro or against a specific theme. Those technologies and their applications on daily conversations over the web are threatening our perception of reality and finally our democratic structures. Preventing the augmented reality of fake news is currently one of the biggest challenges of our decade and this thesis will try to shade some light into understanding human behavior on the web, by concentrating on what really drives human interaction in social networks. Moreover, the center of interest of this thesis will be identifying different pols and how those exert influence in other users via dialogue.

Research Aims and Objectives

- Identify the sentiment of users towards a political event
- Identify different polars within a system
- Identify controversy of ideas within a system
- Investigate the relation between exerting influence within a network and controversy
- Investigate the relation between diffusion of ideas within a network and controversy

Table of Contents:

List of Tables:

List of Figures:

CHAPTER 1: Literature Review

In this chapter a literature review is conducted with the overall aim of providing the conceptual framework of the current research. First, the Social Network Theory is presented, mainly within the context of political content exchange on Twitter. The notion of network growth as well as several diffusion techniques are further discussed, in an attempt to investigate how political ideas can flourish within this type of networks. Adding to that the parameter of influence, a thorough investigation of the opinion detection theory along with several sentiment analysis methods is carried out in order to examine how different opinions can drive polarity. Finally, the aspect of controversy is initially set for further investigation.

1.1 Social Network

Over the last decades, social networking accelerated to a vital tool of our social being, heavily influencing how people are cultivating and sharing new ideas. Today, internet users spent 22% of their total online time in Social Networks (Mohammadpour et al., 2014), while more than 54% of young users (up to 24 years old) have set up their profile page on a social networking site (Palmer & Koenig-Lewis, 2009).

The increasing internet accessibility has turned users into dynamic points of influence with unlimited reach from one user to another. Just as a computer network is a set of machines connected by a set of cables, a social network is a set of people (or organizations or other social entities) connected by a set of socially meaningful relationships.

The most common definition of a social network was introduced by Brass et al., (1998) as a set of actors and ties representing some relationship – or lack of relationship – amongst the actors. Actors or nodes in those networks could be individuals or organizations and entities connected by different type of ties. People are adding friends on Facebook, following others on Instagram and Twitter, or posting their news on Linkedin and sharing videos on YouTube. The hidden power of a network is the ability of all actors to interact at any given time with other actors with no restrictions or limitations.

1.2 Social Network Analysis

The ever-changing connections between nodes results in a massive footprint of information. Some of the strongest entrepreneurs in our world would argue that information is one of the most expensive assets to hold. In social capital theory, networks represent value, where the connections of nodes conceived to be channels of many different resources.

In addition, the structure of those connections and the sequence being created could have a major effect on measuring the value of a network in total, as well as the value of a single node interacting within the system. Granovetter, (1973) in his theory, distinguishes weak ties as the ones bonding actors with pure social interaction in real life and claims that strong ties within a network normally are connecting people who have built a closer relationship. Although a network with strong ties would be characterized of a higher density, it is more likely individuals less familiar to each other exchange new ideas and enhance different perspectives about an event. Granovetter took as an example two persons A and B, both with their own circle of strong ties within the network and observed that the replication of information was very common within those circles, since everyone already knew each other. Even new fractions of information will travel fast around a familiar network and very soon will be “old news”. In the contrary the study concluded that when A and B connect with each other, their own circles are being interconnected in parallel. As a result, when A and B get connected they will create value for the whole system.

Another important parameter, examined by Borgatti & Halgin, (2011), is the structure of connections between more than two nodes. When an individual connects with others, of whom their networks are already interconnected, no real value can be channeled. Connections of individuals with distinct and separated groups will only give the advantage of exchanging novel information, unavailable to other nodes in the network. The “flow theory” commands that flow of information within the network depends on the distance between the nodes, their position in the network and embeddedness, the number of mutual connections.

The above-mentioned theories support the upper goal of channeling new type of information faster and with less redundancy. However, Rost, (2011) argues that individuals require a space to comprehend novel information to prosper into new ideas, resolutions, and knowledge. This space for further reconnaissance can only be provided within groups of nodes connected with strong ties. In this case, strong networks are evolving into support systems where existing trust between its members sustain a social cohesion, allowing people to interact freely and forming their own ideas and opinions. In this context, Adler & Kwon,(2002) stated that “weak ties facilitate the cost-effective search for codifiable information and strong ties facilitate transfer of complex information and tacit knowledge.”

1.3 Social Media Networks

Social media are defined to be the set of online instruments used by individuals to interact. Compared to the traditional source of media, like television or radio, social media allow to the recipients of information communicate back to the source by creating and sharing their own content. After Web 2 revolution released, social media have transformed the form of our communication on the web into a spreading dialog (many-to-many) with the ability to chat, post, like, dislike, video share, unfollow. The following **Table 1** lists the most common social media systems and provides examples under each category.

Table 1: Social Media Systems & Examples

Social Networks/ Messaging	Blogs and Forums	Microblogs	Media Sharing/ Live Video	Geosocial	Ratings and Reviews	Social Bookmarking	Social Knowledge/ Podcasts
Sites/apps that connect people sharing personal or professional interests through profiles, groups, posts/ updates.	Blogspublish posts, multimedia and hyperlinks with commenting. Forums are online discussion sites.	A form of traditional blogging where the posts have been limited in size, length or type of content.	This category is for social media channels developed mainly to share image, video, or audio media content.	User-submitted (GPS) location connects local people, business and events through social media.	Reviews give an opinion. Ratings measure how good something is on a scale. Both are obtained by crowdsourcing.	These are services that allow users to save, comment, and share web website links for content discovery, curation and sharing.	Social knowledge sites allow users to ask questions and get information from real people. Podcasts are subscription episodes of audio/ video content.
Facebook	WordPress	Twitter	YouTube	Foursquare	Yelp	Reddit	Wikipedia
LinkedIn	Tumblr	Pinterest	Instagram	Google My Business	TripAdvisor	Digg	Quora
WhatsApp	Blogger	Tik Tok	Snapchat	Facebook Places/ Instagram Locations	Amazon	Buzzfeed	Podcasts
Messenger	Forums	Clubhouse	Twitch	Snapchat Geofilters	Angi HomeAdvisor		

Source : Keith A. Quesenberry, 2021

1.4 Twitter

It has been observed that within different social media networks, the distribution of content differs, depending on users' gender, location, age, and other demographic factors. Moreover, Facebook and Instagram are well known for sharing personal experiences, while Linkedin has been marketed as a professional tool. Twitter, on the other hand, forms the first choice of users for stating political notions and expressing their opinion for economic and social world issues. This thesis is concentrated into analyzing users' political opinions on Twitter and to this end, an extended review of Twitter basics follows.

Mischaud, (2007) describes the platform of Twitter as the mean to engage in short intervals of communication. The ability to spread information in speed of light by evading time and space made this platform so special for further research exploration. The feature that makes Twitter so appealing to use is the restriction of text length up to 140-characters, forcing users to be concise in an instant. The combination of this novel feature with hashtags, brief keywords prefixed by '#' symbol included to every tweet and making it searchable amongst the traffic, makes Twitter a platform where everybody can easily detect, access, and follow closely any matter of their personal interest and instantly comment on it. In addition, Twitter users can follow and comment to all accounts available in the whole network, independently of a user's location and close interactions, since ,unlike any other social media network, a user do not need authorization to follow another user. That provides to this novel system a factor of approachability surpassing any other platform, considering every single node equal to others and able to agree or disagree with all ideas commuted to the whole network.

Bullard, (2014) on his book, Twitter: Social Communication in the Twitter Age, argues that even if this new phenomenon has some characteristics contributing to social equality, human communication functions are as instinctual as breathing, independently of the instrument used to exchange ideas. Opinion leaders still function the same way and while Twitter can expose an individual to a world of opinions, the actual influence of the majority might still be limited. When it comes to influence, what matters of all would not be how many people tweeted in favor of an idea, but the visibility of those tweets in the rest of the network. Ideas prevailing trough the network are the ones coming from influencers, users followed from the majority of others. A case study on breaking news for George Zimmerman killing Trayvon Martin in 2012, reported in total 5 million tweets supporting the not-guilty verdict within the first 26 hours of the court's announcement. Researchers found that 39% of tweets aimed just spreading the news, rather than state an opinion (Jurkowitz et al., 2013). The number of tweets on the matter was outstanding but the question remains, how many of those posts were actually read?

"Medium must have an audience in order to exert influence."

1.4.1 Twitter Terminology

In this section, a brief review on specific terms of Twitter and activity types are given, also summarized in **Table 2**:

Tweet: a short text or piece of information published on Twitter

Retweet: a repost or forward (a message posted by another user) of a tweet from another user. It is indicated by 'RT' initials.

Favorites: are used by users when they like a tweet. By favoriting a tweet, a user can let the original poster know that you liked their tweet. The total number of times a tweet has been favorited is visible to everyone. It is represented by the icon of a heart.

Followers: The followers of a user are other people who receive the user's tweets and updates. When a user is followed by someone, it will show up in their followers list. The total number of followers a user has, is visible to everyone.

Followees: The followees of a user are other people who follow the user. The total number of followees a user has, is also visible to everyone.

Table 2: Activity Types

Activity Types	
Tweets (by user)	Blocks (by user)
Tweet deletes (by user)	Unblocks (by user)
@mentions (of user)	Mutes (by user)
Replies (to or from user)	Unmutes (by user)
Retweets (by user or of user)	Direct Messages sent (by user)
Quote Tweets (by user or of user)	Direct Messages received (by user)
Retweets of Quoted Tweets (by user or of user)	Typing indicators (to user)
Likes (by user or of user)	Read receipts (to user)
Follows (by user or of user)	Subscription revokes (by user)
Unfollows (by user)	Blocks (by user)

Source : Twitter Web Developer Platform, 2020

1.4.2 Twitter & Data Access

The Twitter API is a set of programmatic endpoints that can be used to learn from and engage with the conversations on Twitter. For the purposes of the current research, an Academic API was used, namely the Twitter API v2, granting free access to full-archive search and other v2 endpoints, as well as enhanced features and functionality to get more precise, complete, and unbiased data for analyzing the public conversation. A list of data points provided by the API used, is presented below:

• **archivesource:** API source of tweet (twitter-search or twitter-stream)

•**text:** contents of tweet

• **to_user_id:** numerical ID of tweet recipient (for @replies)

• **from_user:** screen name of tweet sender

• **id:** numerical ID of tweet

• **from_user_id:** numerical ID of tweet sender

• **iso_language_code:** code (e.g. en, de, fr, …) of sender's default language

• **source:** name or URL of tool used for tweeting (e.g. Web, Tweetdeck, …)

• **profile_image_url:** URL of tweet sender's profile picture

• **geo_type:** form in which sender's geographical coordinates are provided

• **geo_coordinates_0:** first element of geographical coordinates

• **geo_coordinates_1:** second element of geographical coordinates

• **created_at:** tweet timestamp in human-readable format

• **time:** tweet timestamp as numerical Unix timestamp

1.5 Network Diffusion & Social Influence

Viral marketing and network interdiction pertains to be a range of disciplines where diffusion strategies have been studied more thoroughly. The main research interest is focused on optimization problems to either maximize or minimize the propagation of information.

The idea of diffusion being maximized by commuting pieces of information through influencers, people with combined characteristics like credibility or connectivity and centrality within the network, is still under investigation by researchers. Although this mechanism is broadly seen as a number of special dots connected within the network in a way of spreading disproportionately ideas in large number of others, little empirical evidence has so far proved such findings. The main challenge would be to actively tracking the network, allowing us to detect influence, while taking into consideration that all recorded data tend to be biased more heavily weighted on "successful" diffusion examples.

In this context, Ghosh & Lerman, (2010) weighted how news on Twitter are spreading on Digg (social news platform). More work related to diffusion analysis came from Adamic & Adar, (2003) using diffusion trees, while using the results of an e-commerce site, and others sought to explore how many influencees made a recommendation critic. More recent studies on Twitter diffusion were concentrated on the metrics and measures of influence. Kwak, H., Lee, C., Park, H., & Moon, (2010) found that influential nodes ranking was differing based on the metric of measurement. Choi et al., (2009) also concluded in similar findings, by comparing the number of followers and retweets, when Weng & Lim, (2010) measured the number of followers & page rank. All the above-mentioned studies focused on identifying opinion leaders and depicting the most reliable metrics to do so. On the other hand, under the paper "Everyone's an Influencer" (Eytan Bakshy, Jake M. Hofman, Winter A. Mason, Duncan J. Watts, 2011) researchers focused on predicting diffusion and quantifying the whole network and not only influencing nodes to achieve this. Another important factor taken into consideration on this research, was content relation with information diffusion in the network. Results showed that most cost-effective influencers would be the ones exerting even less than average influence. Yang & Counts, (2010) focused as well on recognizing diffusion properties and found that some tweet properties can have a greater predicting value than users' characteristics, like the rate a user is being mentioned historically.

1.6 Natural Language Processing

Definition: "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis, for the purpose of achieving human-like language processing for a range of tasks or applications.” (Grosz, 1982)

The goal of NLP is human-like language processing. So far, the abilities of NLP processing are confined to: a) paraphrase an input text, b) translate the text into another language and c) answer questions about the contents of the text. In order for an NLP system to achieve those abilities, it utilizes different levels of language. Psycholinguistic research suggests that language processing is dynamic, as the levels of language can interact in a variety of orders. The more levels an NLP system utilizes, the more effective it can be.

Some of the levels of Natural Language Processing are briefly described below.

1.6.1 Phonology

This level has to do with interpretation of sounds. Phonological analysis uses three different types of rules 1) phonetic rules, 2) phonemic rules, 3) prosodic rules. The former two deal with sounds of different words and contemplate different pronunciations, respectively while the latter concentrates on the variance of intonation across a sentence. NLP systems are processing oral inputs and the sound waves are being analyzed into digital signals for further process under various rules.

1.6.2 Morphology

This level contemplates words, since all words are originally made of morphemes, smallest units of context. Humans are breaking down all synthetics of a word in order to find out its meaning, for example the word preannouncement breaks to following separated morphemes: the prefix pre, the root regis, the root announce and the suffix ment. NLP systems are recognizing its morpheme to capture meaning. If a specific word has as a suffix – ing it conveys that the action of the verd is continuous.

1.6.3 Lexical

Preprocessing of text can enhance the word-level understanding, in particular we have the assignment of a single part of speech tag to a word. Under this processing, words are being assigned with the most probable part of speech based on the overall content of a specific sentence they are occurring. Further to these words are also being replaced by semantic representations of their meaning. In general, a single lexical unit is being decomposed in more abstract properties, allowing NLP to simplify a sentence and unify meaning across words to produce more complex concepts. Lexicons can have simple formats, by including words and their parts of speech, or increasingly complex and maintain information on semantic limitations of each word, the arguments it takes or the semantic field where polysemous words are used.

1.6.4 Syntactic

This level focuses on analyzing words into their grammatical structure within the sentence. With the use of a grammar and a parser, this method unfolds the structural dependency relationships between words. Syntax is a great aspect of every language to acquire real meaning out of a sentence since order and dependency contribute to expressing meaning. A simple example would be comparing the sentences a) "big fish eat the small one" and b) "small fish eat the big one", where they only differ in syntax.

1.6.5 Semantic

This level of processing is contributing to disambiguation of polysemous words. Semantic disambiguation allows only a single sense of a word to be allocated in the semantic map a sentence is representing. As an example, among other meanings, 'chair' can have the use of a furniture for casual conversations or describe the role of a SEO in the organization chart of a business. When information of the entire sentence is required to accomplish disambiguation of meaning for a specific word, then the semantic process is used. For implementing disambiguation, a range of methods can be implemented, some of which require the frequency that a specific word is being used or utilize knowledge of the theme a document contemplates.

1.6.6 Discourse

Discourse focuses on full text instead of specific words, by creating connections between component sentences. The most common discourse processing are anaphora resolution and discourse structure recognition. The former is replacing pronouns with the specific nouns they are referring to, while the latter identifies the parts and side sections within a text.

1.6.7 Pragmatic

The goal in this level is extracting hidden meaning of text and the underlined ideas opposed when specific words are skipped from a sentence. It requires knowledge basis and inferencing modules for this level to be accomplished.

1.7 Approaches to Natural Language Processing

Natural language processing approaches are being distributed into four categories: symbolic, statistical, connectionist, and hybrid. The first approaches carry out deep analysis in linguistics and demand for guidance, human-developed rules and lexicons, the same way rule-based systems work.

Statistical approaches develop generalized models of linguistic phenomena using observable data as a source of input. Examples of such approaches could be tasks of speech recognition, parsing, and part-of-speech tagging. Connectionist approaches work similarly with the statistical ones, but in addition connectionist models are using theories of representation. A connectionist model could be examined as a network of connected units, where their edges are weighted based on general knowledge. Relationships between those units are encoded by the weights of connections, with the result to spread knowledge among the network and in parallel reflect structural relationships. Those type of approaches are more suitable for syntactic parsing.

1.8 Natural Language Processing Applications

Any application that processes text could use NLP. Following are the most common applications utilizing NLP and take a further look on Text Mining, as this is where next chapters are focusing:

Information Retrieval
Information Extraction (IE)
Question-Answering
Summarization
Machine Translation
Dialogue Systems

1.9 Text Mining

Text mining and more generally the knowledge discovery from textual databases is the process of extracting non-trivial patterns from documents consisting of text. Text mining is considered to be a complex task since it processes mostly unstructured and fuzzy data. Text mining has two individual phases, transforming first text documents into a chosen intermediate form and then extracting patterns or knowledge from this last form. The intermediate form can be structured or non-structured, as well as document based, or concept based. Data mining operations including predictive modeling and associative discovery can be accomplished with concept based intermediate forms. **Figure 1** (Tan, 1999) demonstrates Text Mining process :

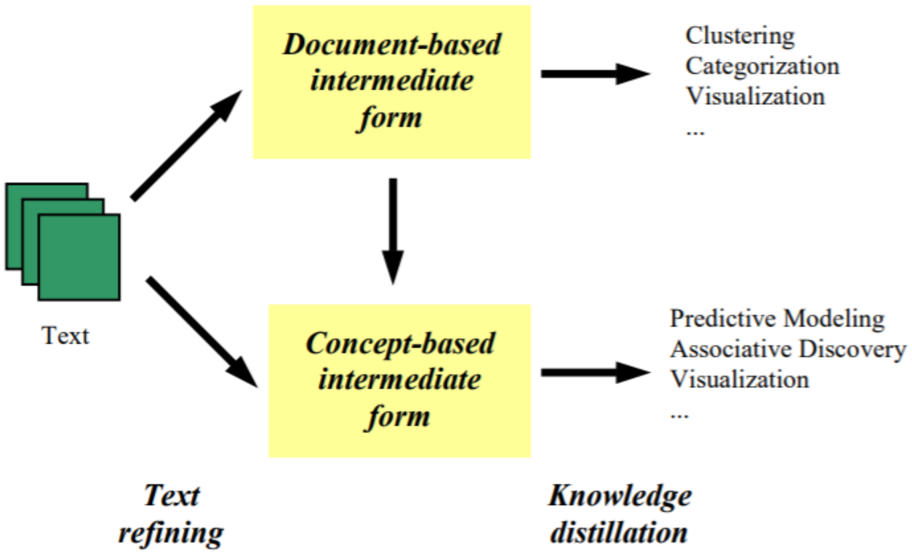

Figure 1: Text Mining Process

Source : Tan, 1999

1.10 Opinion Mining

Identifying and reflecting subjective thoughts throughout the web in an automatic way, can create substantial value. Applications like targeting marketing and fraud detection could have actual effect with the ability to understand social behavior. Affects, feelings, emotions, sentiments, and opinions are terms related to human subjectivity. Human subjectivity derives from an individual experiences and beliefs reflecting to inner judgment and instincts. For that reason, detecting subjectivity is a difficult natural language processing (NLP) task. Opinion mining or else sentiment analysis is the coupling of natural language processing and text mining, using machine learning to process text and classify it as positive or negative. It is defined as the task of classifying the opinions of authors about specific entities. In sentiment analysis we have two types of observation, facts, and opinions. The first type of information is the objective truth about an event or situation, while opinions are the attitude and the expression of emotions towards this event. The current thesis focuses on political opinions and their classification into positive or negative in relation to a specific subject. There are numerous challenges when it comes to opinion mining, such as identifying the sentiment of a single word, since it is very common for words to have both positive and negative meaning. If for example the word "high" is used for describing somebody's height then it has a positive sentiment, whereas if it is used for stating the height of a mountain by a climber then sentiment could be considered negative. There are two approaches used for sentiment analysis, the bag of words (BOW) and the feature-based sentiment (FBS). In first approach, each document is representing a set of words not connected with each other, so there isn't any semantic information between words. For that reason, this specific approach is not useful for analyzing opinions about a specific event. Furthermore, there are three components of opinion, those are opinion holder, object, and opinion. The opinion holder is an individual or a group of people holding a specific opinion about an object. The evaluation of text can be Direct or with Comparison. When there is a direct opinion about an object, then sentiment is consequently addressed specifically to that object. With the comparison opinion, on the other hand, objects in different statements are being compared, for example ice hockey is better than football.

1.11 Opinion Classification

Opinion classification categorizes opinions mainly in positive, negative, and neutral. This part of sentiment analysis is a difficult process. Taking as an example the sentence “I can’t take this anymore”, none of the containing words have negative sentiment, although the sentence itself has a negative sense. Classification of opinions can be performed on the different following levels:

1.11.1 Document level Sentiment Analysis

The document level sentiment analysis is known as the simplest form of classification, where a single document consisted of text is considered as one entry of information. For this level, it is assumed that all opinions within a document are focusing on a singular object. The document is being classified either as positive or negative and irrelevant or neutral sentences will be removed before processing. There are two approaches performing classification a) Supervised machine learning approach and b) Unsupervised machine learning approach. Under first approach, there is a finite number of classes, while a training dataset is also available. Based on the training data, the system classifies documents using algorithms like Support Vector Machine, Naïve Bayes, K Nearest Neighbours, and Maximum Entropy, etc. Research on the specific approach has been done by Wang et al., (2014), where different supervised learning algorithms were used. Also, Leena A Dhande, (2014) ,combined Naive Bayes and Neural Network classifiers with the result of sentiment analysis being increased to 80.65%. In unsupervised approach, the system first determines the sentiment orientation of words within a document. If the total sentiment of the words is considered to be positive, then document has also been classified as positive. For this approach, lexicon-based method has been used by Taboada et al., (2011), while Sharma et al., (2014) used an unsupervised dictionary-based technique (WordNet).

1.11.2 Sentence Level Sentiment Analysis

Sentence level sentiment analysis is more detailed than document one. Every sentence is considered having separate sentiment. Sentence level sentiment analysis is following two different tasks. The first task is subjectivity classification, where a sentence can be classified into subjective or objective. The subjective sentences contain opinions while the objective ones only consist of facts. The second task is sentiment classification of a sentence to positive, negative, or neutral. The same document level classifications can be applied also for sentence level, in relevant research. Jagtap & Pawar, (2013) observed that first and last lines of a review are more indicative for the total polarity.

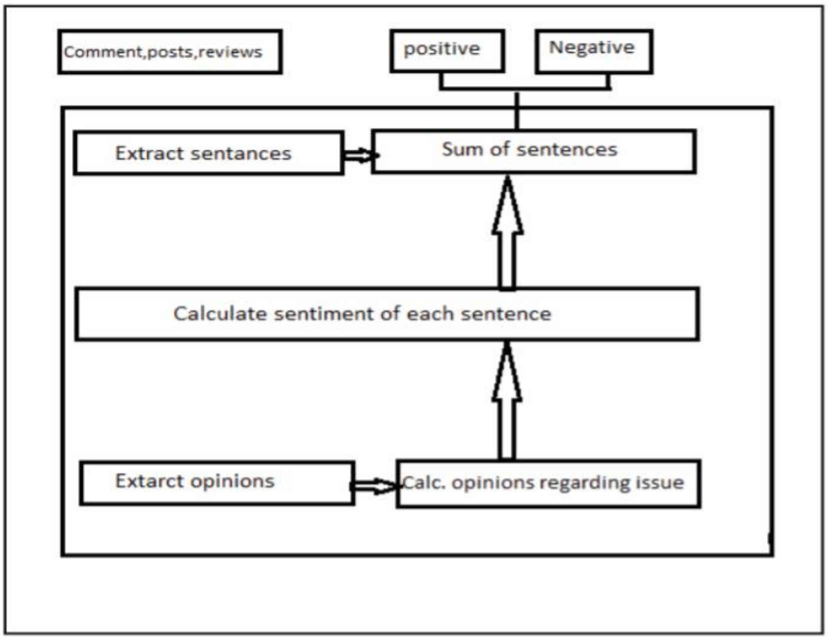

Figure 2: Architecture of Classification of a Social Event

Source: Singh & Dubey, 2014

1.11.3 Feature Level Sentiment Analysis

The basic step in Feature Level (Aspect based analysis) sentiment analysis, is to identify the features of an object. For example, in the phrase “The camera of Xiaomi Redmi Note 9 is very good”, the word “camera” is product feature (noun) and “very good” is opinion (adjective). The steps for feature-based sentiment analysis, according to (Tribhuvan et al., 2014), are the following:

1. Preparing Review Database
2. POS Tagging
3. Feature Extraction
4. Opinion Word Extraction
5. Opinion Word Polarity Identification
6. Opinion Sentence Polarity Identification
7. Summary Generation

Most related work used different NLP methods like frequent features, compactness pruning, P-support pruning and infrequent feature identification. Those methods identify all aspects in a corpus of product reviews and extract all noun phrases (NPs), while keeping just the NPs whose frequency is above some experimentally determined threshold. Another approach to feature identification is using a phrase dependency parser to find the aspects. Aspect identification can be also considered an information extraction problem and use a tagged corpus to train a sequence classifier such as a Conditional Random Field (CRF) to find the aspects.

1.12 Major Challenges in Sentiment Analysis

Language considers to be a phenomenon of human interaction and thus, there are many challenges identifying the meaning of words and its connections. The main issues and challenges in the field of sentiment analysis are briefly summarized below:

Word Sense Disambiguation considers to be the problem of identifying the meaning of a word, based on the context of the phrase it relays on. Words can have different meanings for different domains.

Comparison is the challenge of determining the polarity for comparative sentences. For example, the battery life of a pc X is better than pc Y. We can easily identify the positive word 'better' but the author's preferred object is not easy to determine, which is the key piece of information in a comparative reviews.

Negation within a sentence can ultimately change the polarity of a sentence, if not accounted. For example, in the phrase “There is a good chance this laptop will not break easily” we have a positive meaning, although negation determines this polarity.

Intensity might play a big difference identifying the degree of polarity within a sentence.

Sarcasm can be identified with difficulty within a text since it a strongly subjective phenomena of speech. Again, the presence of sarcasm within a text has the ability to utterly change the total polarity of words in sentence or a text.

Chapter 2: Methodologies

In this chapter a thorough review on sentiment and graph analysis methods follows. All sentiment classification techniques are presented, with an emphasis on Naïve Bayes, SGD algorithms, as well as BERT Model, which were utilized on current research. Moreover, a brief description on graph analysis algorithms and the software packages used is provided.

2.1 Sentiment Classification Techniques

Sentiment Classification techniques are divided to machine learning, lexicon based, and hybrid approach. In the first category, machine learning algorithms are being applied, while the lexicon-based approach uses lexicons. In the hybrid approach, both methods are used to achieve maximum results, while most of hybrid examples use sentiment lexicons. Machine learning methods are being categorized to supervised and unsupervised learning. The former is used effectively when a large number of labeled data is available, for training the algorithm. Dictionary-based approach focuses on identifying opinion seed words and mapping their sentiment based on other synonyms available in the dictionary used. The corpus-based approach starts with defining a list of opinion words and then maps all documents for more similar words, by applying statistical and semantic methods.

2.1.1 Classification Workflow

For the sentiment analysis, the two classes where our data should be allocated were positive “1” and negative “0”, forming the classification labels. The classification process has two phases, a learning phase and an evaluation phase. In the first phase, the classifier trains its model on a given dataset, while in the evaluation phase, it tests the classifier performance. A model’s performance is evaluated by many different factors, including accuracy, error, precision, and recall. The Classification Workflow is being represented in **Figure 3** (Navlani, 2018).

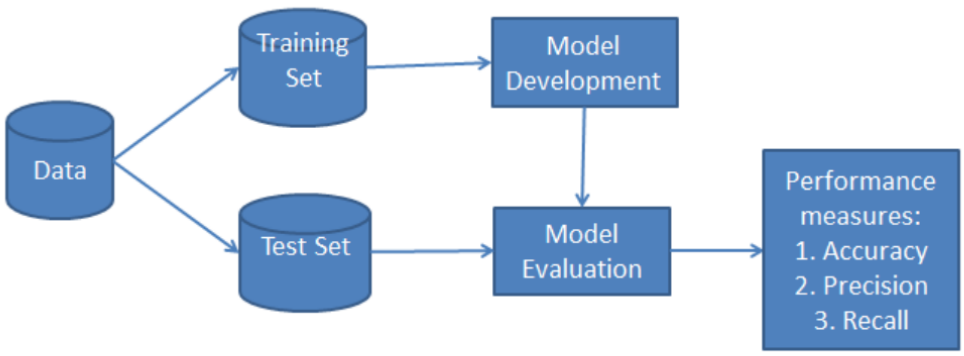

Figure 3: Classification Workflow

Source : Navlani, 2018

2.1.2 Machine learning approach

In machine learning approach, ML algorithms are used to resolve text classification problems with the help of syntactic and linguistic features. In text classification problems we have a set of training records, where each record is labeled to a class. For a given instance of unknown class, the model predicts a class label for it. When only a single label is assigned to an instance, the complexity is being maximized in solving the classification problem. Consequently, when a probabilistic value of labels is assigned to an instance, the soft classification problem exists.

2.1.3 Supervised Learning

Supervised learning methods use labeled training documents as an input. Supervised classifiers are divided into different categories, subscribed below:

Probabilistic classifiers

Probabilistic classifiers use a mixture of models, assuming that each class is a component of the mixture. A mixture component is a model that provides the probability of sampling a particular term for that component. Those classifiers are also known as generative classifiers. The most common generative classifiers are briefly described below:

Naive Bayes Classifier (NB)

Naive Bayes is a supervised learning algorithm based on Bayes' theorem, with the "naïve" assumption of conditional independence and equal weight between every pair of features given the value of the class variable. It is a straightforward and fast in execution classification algorithm, suitable for big data analysis. Conditional independence can be explained by the following example: is a job applicant desirable or not, depending on his/her competence, grades in school, master green or gender? Even if these characteristics are interdependent, there are still considered independently. This assumption deescalates the complexity of computation, and for that reason it is considered as naïve. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x1 through xn:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- P(h): the probability of hypothesis h being true (regardless of the data), also known as prior probability of h.
- P(D): the probability of the data (regardless of the hypothesis), also known as the prior probability.
- P(h|D): the probability of hypothesis h given the data D, known as posterior probability.
- P(D|h): the probability of data d given that the hypothesis h was true, known as posterior probability.

The algorithm execution requires the following steps:

1. Calculate prior probability for given class labels
2. Calculate conditional probability with each attribute for each class
3. Multiply same class conditional probability
4. Multiply prior probability with step 3 probability
5. See which class has higher probability, higher probability class belongs to given input set step.

Bayesian Network (BN)

Naïve Bayes classifiers are based on the assumption that all features within a text are independent. The opposite approach is to assume all features are dependent. The last assumption is used for the Bayesian Networks, where documents represent a graph of random nodes, interconnected with conditional dependences, represented by the edges. Hernández et al., (2015) tried solving a real-world problem with three different target variables. By using multi-dimensional network classifiers, they joined those different target variables in the same classification task for the model to consider potential relationships between them. They also extended the framework to a semi-supervised domain, taking advantage of the huge amount of information available related to the content. This approach outperformed any other sentiment analysis approaches, providing the best solution for a semi-supervised framework.

Maximum Entropy Classifier (ME)

The Maxent Classifier, known also as a conditional exponential classifier, transforms label features to vectors with encoding. Those vectors are used for calculating weights for each feature, based on which features are being distributed for the most probable label. Kaufmann, (2012) used ME classifier to detect parallel sentences between any language pairs. Results showed that ME classifiers are useful for all language pair. This can allow the creation of parallel corpora for many new languages.

Support Vector Machines Classifiers (SVM)

Support Vector Machines are broadly considered as a classification approach, although it can be employed in both types of classification and regression problems, having the ability to handle both continues and categorical variables. SVM draws a hyperplane in multidimensional space to separate different classes, in an iterative manner, in order to minimize the error. SVM is focused in maximizing the marginal hyperplane, dividing the dataset into classes.

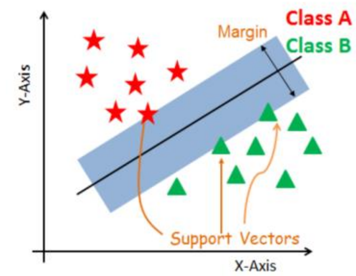

Figure 4: Support Vector Machines Visualization

Source : Navlani, 2018

As shown in **Figure 4** (Navlani, 2018), support vectors are the data points close to the hyperplane. Those points are defining the separating line by calculating margins. The hyperplane is a decision plane, separating objects having different class memberships. Margin is the gap between the two lines on the closest class points. The larger the margin, the better is considered to be for the method.

The main idea is dividing given dataset in the best possible way. The objective is selecting the hyperplane with the maximum possible margin between support vectors in the given dataset. The steps for identifying the maximum marginal hyperplane are: a) generating hyperplanes that are better segregating classes and b) selecting the right hyperplane with the maximum separation from the nearest data points. The loss function that helps maximizing the margin between the data points and the hyperplane, is the following:

$$C\big(x, y, f(x)\big) = \big(1 - y^{*} f(x)\big) +$$

If the predicted value and the actual value are both n, then cost is 0. If those values are not equal, then the loss value is being calculated. Also, as a regulatory parameter the cost function is used with the objective to balance the margin maximization and loss at the same time. The cost function develops to the below:

$$min_w \lambda \; \|w\|^{\,2} + \sum_{i=1}^{n} (1 - y_i \, \langle x_i, w \rangle)_+$$

Afterwards, the model defines derivatives with respect to the weights in order to find the gradients and the function is developed by also adding the weights:

$$\frac{\delta}{\delta w_k} \lambda \; \|w\|^{\,2} = \; 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ \; = \begin{cases} 0, & if y_i \langle x_i, w \rangle \geq 1 \\ -y_i \, x_{ik}, & else \end{cases}$$

Stochastic Gradient Descent Classifier

As misleading the title is, the Stochastic Gradient Descent – Classifier (SGD-Classifier) is not a classifier. SGD Classifier is a linear classifier such as SVM and logistic regression optimized by the SGD. While SGD is an optimization method, it also includes a classifier which is a machine learning model. For example, a machine learning algorithm defines classes for an object and the optimization method of SGD optimizes the performance of the classifier. For the next sentiment analysis attempt, the available SGDClassifier model from Sklearn was used. This estimator is implementing regularized linear models with stochastic gradient descent (SGD) learning. For the training, the loss is estimated for each sample at a time and the model is updated along the way with a decreasing strength schedule. When using the default function, the model is automatically using the linear SVM classification algorithm.

Neural Network (NN)

A neural Network consist of neurons with different inputs. Each neuron is weighted in order to compute a function of its inputs. Based on inputs and weights, the output is generated. Moraes et al., (2013) presented a comparison between SVM and ANNs (artificial neural networks) indicating that ANN produce better results in most cases. They also highlighted the computational cost of SVM in terms of running time and of ANNs in terms of training time. Tracking relationships between users as positive, neutral, and unknown, van de Camp and van den Bosch found that training sets consisted of users with relations produces better results and concluded that ANNs can have valuable results for classifying personal relationships.

Decision tree classifiers

Decision tree classifiers divide training data based on conditions. The condition is the presence or absence of words. Data are constantly being divided until leaf nodes remain to have a minimum of features used for the classification. Other condition for separating text could depend on similarity of documents. Single Attribute split use the presence of absence of specific words to proceed with separation, while similarity-based multi-attribute split uses frequent words, compares how similar documents containing those words are probable to be and proceed this way with the split. Additionally, discriminate-based multi-attribute split uses discriminating factors for the split. Hu et al., (2006) used the Maximum Spanning Tree (MST) structure and created the Topical Term Description Model for sentiment classification. Topical terms are specific aspects in a particular domain. Song & Lu, (2015) have studied a propagation approach for taking advantage the outside sentence features as well. Their approach performed better than from not using outside sentence features and outperformed previous representational approaches.

Rule-based classifiers

In rule-based classifiers, certain rules are being applied to datasets, most of those conditions are on term presence. There are many criteria that are generating rules, the most common of which are support and confidence. Any instance in the training dataset relevant to the rule, is being added to calculate support. Several combined rule algorithms have also been applied for research purposes. For example, Quinlan., (1964) studied decision tree and decision rule problems combined. The difference between them is that DT apply strict hierarchical rules on the data space, while rule-based classifiers allow overlaps.

2.1.4 Semi and Unsupervised Learning

As already mentioned, supervised learning requires a number of labeled documents as a training set in order to classify input documents to a certain category. However, the availability of those training sets is limited, making room for the unsupervised learning to tackle classification problems without the use of training sets. Ko & Seo, (2000) proposed a method that separated documents into sentences and then categorized its sentence based on the keyword lists of each category and by measuring sentence similarities. He & Zhou, (2011) on the other hand, utilized a semi-unsupervised method and obtained a classifier using information extracted from an existing sentiment lexicon. They then used this information as labeled features to constrain the predictions of the model. This method outperformed other semi supervised learning approaches and proved that could be applied to any text classification task, where some prior knowledge is available. Xianghua et al., (2013) used the unsupervised approach to discover different aspects discussed and their sentiments in Chinese social reviews. First, they applied an LDA model to identify multi-aspect global topics and then compared those with local topics and their sentiments. Their approach tent to have good topic partitioning results improving sentiment analysis accuracy. Other unsupervised methods are focused on measuring similarity between words and polarity prototypes, by using lexical associations, semantic spaces, etc.

Meta classifiers

Usually, researchers are using more than one classifier to achieve results. Lane et al., (2012) proposed an ML method to locate documents carrying positive or negative sentiment in social media. They classified documents by detecting sentiment and then assessing it further to find out if it was positive or negative. It was proved that balancing the class distribution in training data improved the performance, since algorithms are less biased to one class against another. Huaxia Rui, Yizao Liu,Andrew B. Whinston, (2010) explored ML algorithms on streaming data from Twitter and tried to investigate how word of mouth can affect movie sales. Utilizing NB and SVM methods, they classified opinions for two different categories of users, the pre-consumers, and post-consumers. Their results showed that word of mouth significantly affects more users in Twitter with more followers and at the same time, the effect is stronger in pre-consumers than users who already seen the movies. J. B. and J. Wang, (2016) tried to capture dependencies among words and turn those into a vocabulary to feed as input for algorithms. First, the algorithm learned conditional dependencies among words and transformed them into a Directed Graph, and then it was fine-tuned for a higher cross-validated accuracy. This method turned to predict better results about sentiment orientations comparing to other methods. Huaxia Rui, Yizao Liu,Andrew B. Whinston, (2010) combined supervised and unsupervised methods to create meta-classifiers and developed a polarity classification system. At first, they generated two models and applied machine learning, then integrated SentiWordNet sentiment corpus into the English using semantic orientation approach. Finally, with a meta-classifier, they combined all three steps. Walker et al., (2007) tried to classify stance, an overall opinion of a user for a specific idea or object. Stance can be considered as the point of view someone holds for a specific topic. Researchers classified stance for political debates. They acquired 104 debates between two users and identified the stance for each user. They achieved better classification accuracies on topic basis, than unigram baselines when using sentiment, subjectivity, dependency, and dialogic features.

2.1.5 Lexicon-based approaches

Opinion words with positive polarity and negative polarity are being widely used for sentiment classification tasks. Opinion phrases and idioms are called opinion lexicon. There are the main approaches to collect opinion words: manual approach, dictionary-based approach and corpus based approach. All the three above-mentioned approaches, are described below:

2.1.6 Dictionary-based approach

As a first step in this approach, an initial set of sentiment-oriented opinion words is manually collected and then this set spreads by using corpora WordNet or thesaurus for synonyms and antonyms. This Iteration process stops until no new word is being found. The disadvantage of dictionary-based approach is the inability to find domain and specific context. Huaxia Rui, Yizao Liu,Andrew B. Whinston, (2010) proposed a strategy to improve a relevance based on user's sentiment analysis. Their results were highly effective for proposing ads based on keyword extractions.

2.1.7 Corpus-based approach

The corpus-based approach overtakes the above-mentioned problem, finding opinion words with context orientation. These methods along with the technique of finding a large corpus of opinion words are based also on syntactic patterns between words. Hatzivassiloglou & McKeown, (1997) took seed opinion adjectives and used them together with linguistic constraints to find additional adjectives with certain orientations. Connectives like “AND”, “OR”, “BUT”, “WITHER-OR” are used as constraints, for example the conjunction AND declares that adjectives connected with this conjunction tend to have same orientation. Other conjunction, such as «but», «or», «however», indicate opinion changes. In this way, by identifying orientations between words based on conjunctions, learning is spread in the whole corpus. The links between adjectives form a graph and then clustering is applied categorizing opinion words to positive and negative. Conditional Random Fields (CRFs) identifies opinion words and has been used by Zhou et al., (2017) on Chinese online reviews. Xu et al., (2011) utilized complicated dependencies between words and their relations and unfixed interdependencies among relations. Following this approach, they achieved demonstrating a graph on comparative relations of products based on customer reviews. Cruz Mata et al., (2008) proposed the taxonomy-based approach for identifying feature-level opinions. Taxonomy is the semantic representation of polarity in parts of an object. Their research proved the importance of domain to extract opinion words more accurately. When compared to corpus-based approach, the dictionary-based approach is more effective for preparing large corpuses, obtaining an important number of oriented opinion words. On the other hand, the corpus-based approach has an advantage on identifying the different orientations of words in between domains. The corpus-based approach is performed using either statistical approach or semantic approach, which are described below:

2.1.8 Statistical approach

Statistical techniques can be used for finding patterns between seed opinion words. Fahrni & Klenner, (2008) proposed using the co-occurrence of adjectives in a corpus and their deriving polarities. The polarity of the word can be found by measuring the frequency of the word in the whole dataset. When the word occurs more frequently within text with positive polarity, then it is also mapped as positive and vice versa. If similar words appear together frequently within same content, then they are also classified with the same polarity. Consequently, the polarity of a word can be determined based on the relative frequency of co-occurrence with another word.

2.1.9 Semantic approach

The semantic approach relies on the principle of giving similar sentiment values to semantically close words. With the use of WordNet for example, a list of sentiment-oriented words based on the synonyms and antonyms can be found and the polarity of new words based on the relative count of positive and negative synonyms of this word can then be determined.

2.1.10 Hybrid Approach

Lexicon-based and natural language processing techniques

The lexicon-based approach is used to find the syntactical structure of sentences within a text and help in identifying semantic relationships between words. Di Caro & Grella, (2013) approach concentrated on deep NLP analysis using dependency parsing. They assumed that every linguistic element has value of sentiment based on the syntactic structure of the parsed sentence. Min & Park, (2009), on the other hand, used NLP techniques to capture tense and time expressions. Their metric had two parameters based on the time expressions captured and the time of purchasing periods. Results showed that their metric was helpful and free from undesirable biases.

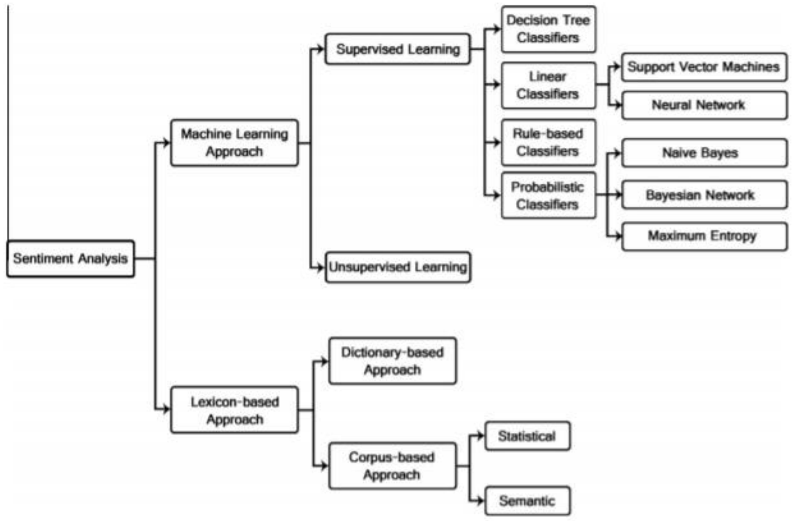

Figure 5: Sentiment Classification Techniques

Source : Medhat et al., 2014

2.2 Aspect Based Sentiment Analysis

One of the most common problems in sentiment analysis, is identifying sentiment for a specific aspect mentioned within a phrase. For most cases when analyzing text data, the main interest is not in principle limited finding the average sentiment or general attitude of an individual but capturing his/her opinion about a specific subject. As an example, the following product review for Kindle Fire in Amazon, is presented: "As a long-time Kindle fan I was eager to get my hands on a Fire. There are some great aspects; the device is quick and for the most part dead-simple to use. The screen is fantastic with good brightness and excellent color, and a very wide viewing angle. But there are some downsides too; the small bezel size makes holding it without inadvertent page-turns difficult, the lack of buttons makes controls harder, the accessible storage memory is limited to just 5GB." Assessing above feedback wholistically without investigating further the review components, would not give any additional value for the company. Within the above-mentioned review, the user criticizes both positively and negatively the different aspects of the specific product (speed, ease of use, screen quality, buttons, and storage memory size), at the same sentence. Aspect-based sentiment analysis, also called feature-based sentiment analysis, forms a more sophisticated tool compared to the conventional sentiment analysis approaches, focusing on recognizing polarity and the aspects to which it refers to. The most common approach used is identifying all aspects by extracting all noun phrases (NPs) and then retain only NPs with a certain frequency and tightly related to the initial product category. A second approach for aspect identification, is using a phrase dependency parser. Additionally, aspect identification can be treated as an information extraction problem, using tagged corpus for training a classifier to find aspects. In an even more sophisticated framework, aspects mentioned within the text but also those not mentioned but inferred, could be identified. One such case is the following phrase, criticizing the size of a camera: “the camera is quite compact.” Researchers also suggested a two-phase co-occurrence association rule mining approach to match sentiment expressions with aspects.

One way to extract such implicit aspects is suggested from Hoang et al., (2019), where a two-phase co-occurrence association rule mining approach is used to match implicit aspects (sentiment expressions) with explicit aspects. With these two sets, a simple algorithm can be used, that determines the polarity of each sentiment expression based on a sentiment lexicon, sentiment shifters (such as negation words) and special handling of adversative conjunctions, such as ‘but.’ The final polarity of each aspect is determined by a weighted average of the polarities of all sentiment expressions, inversely weighted by the distance between the aspect and the sentiment expression.

2.2.1 Aspect-Based Sentiment Analysis Using BERT

BERT

Bidirectional Encoder Representations from Transformers, broadly known as BERT, is a pre-trained language model designed to identify the meaning of a word, analyzing both left and right side of the text at the same time. For the pre-training procedure, BERT uses BooksCorpus (800M words) (Su et al., 2019) and Wikipedia (2,500M words), keeping only the text passages. BERT analysis has proven to achieve better results than any other technique in sentiment analysis. Modified Language Model Masks (MLM) are used in order for the left and right pre-training of BERT to be achieved. As a first step, a random word within a sentence is masked and replaced with a token [MASK], following by a procedure in which the model tries to predict the masked word, analyzing both right and left side of the text with the help of transformers. In the following paragraphs, a more detailed discussion about the individual elements of Bert (language) model is performed, with the overall aim of gaining deeper knowledge about this model that was utilized for the purposes of the current research. More specifically, the BERT input, the Masked LM, Next Sentence Prediction, and Fine Tuning are presented and further discussed.

BERT Input

The text input is first processed by wordpiece tokenization, by producing tokens for every single word. Two additional tokens are also the classifier token (CLS), added in the beginning of the text and the separation token [SEP], placed in the end of the text. Those tokens are later processed by three embedding layers, namely the Token Embedding Layer, the Segment Embedding Layer and the Position Embedding Layer.

Bert is using Transformer, an attention mechanism which identifies contextual relations between words. Transformer uses two basic mechanisms, the encoder for reading text input and the decoder which produces a prediction for the task. The process is showcased in **Figure 6** (Devlin et al., 2019).

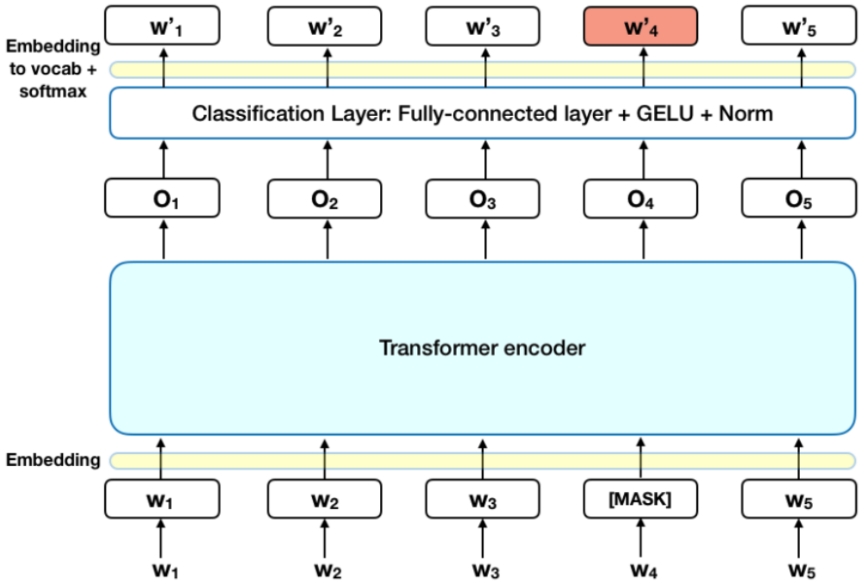

Figure 6: Transformer Encoder

Source : Devlin et al., 2019

Masked LM

Fifteen percent of the words within a sentence is replaced by a MASK token. Following that replacement, the model predicts the context of the masked words based on the remaining words within a sequence. To predict the output words, a classification layer is required to be added, the output vectors should be multiplied by the embedding matrix (by transforming them in the vocabulary dimension) and in that way, the probability of each word in the vocabulary can be calculated, using softmax.

Next Sentence Prediction (NSP)

Within the training process, the model analyzes pairs of sentences and predicts if the second one is the subsequent sentence in the original document. During training, 50% of pairs have the subsequent sentence in the original document, while the other fifty percent have a random sentence of the text. The hypothesis is that the random sentence will be disconnected from the first sentence. In order for this differentiation between two sentences to be achieved, the model processes the text input in the following order: first, a [CLS] token is placed at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. A sentence embedding is added to each token, indicating sentence A or B and a positional embedding is added to each token, indicating its position in the sequence. The above mentioned information, is depicted in the following **Figure 7**.

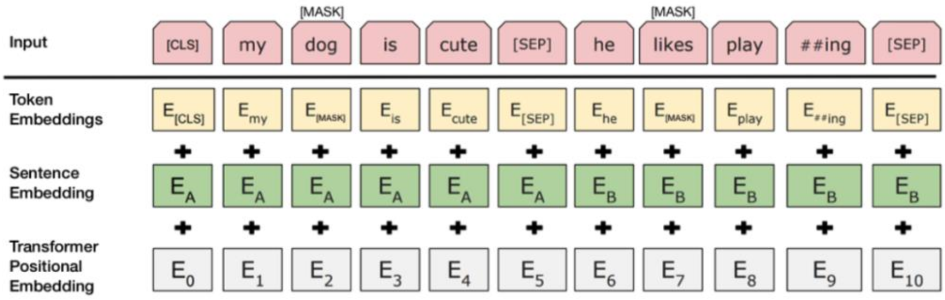

Figure 7: Next Sentence Prediction Process

Source : Devlin et al., 2019

Fine-tuning

The self-attention mechanism allows BERT to accomplish many different tasks, on single text or text pairs base. For every different task, task-specific inputs and outputs have to be inserted into BERT. The input for pre-training could be: a) sentence pairs in paraphrasing, b) hypothesis-premise pairs in entailment, c) question-passage pairs in question answering and d) a degenerate text-Ø pair in text classification or sequence tagging. The token representations are being distributed in for token level tasks, such as sequence tagging or question answering, when [CLS] representation is being distributed into an output layer for classification. Fine-tuning is less time sensitive comparing to pre-training and no more than some hours are needed for results to be replicated on Cloud TPU or GPU.

Transformer Model Architecture

The Transformer is a model that uses attention to boost the training speed of neural machine translation applications with an encoder-decoder structure. The encoder allocates an input sequence of symbol representations (x1,…,xn) to a sequence of continuous representations z=(z1,…,zn). The decoder follows by generating an output sequence (y1,…,ym) of symbols, one element at a time. At each step the model uses the previously generated symbols as an input for the next one.

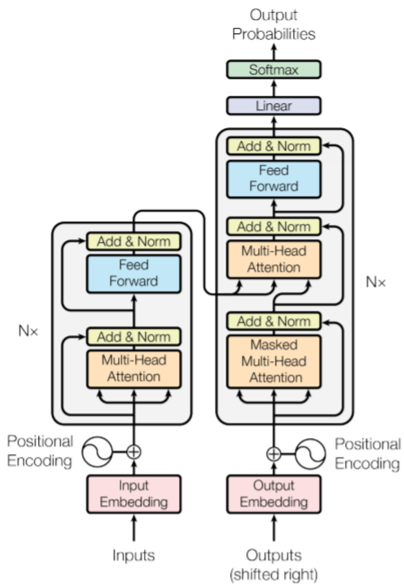

Figure 8: Transformer Architecture

Source : Vaswani et al., 2017

Encoder & Decoder

Each layer in the encoder has two sublayers: A multi-head self-attention mechanism and a position wise connected feed-forward network. A residual connection is being employed around each of the two sublayers, followed by a normalization layer. The output of every sublayer is LayerNorm(x+Sublayer(x). The decoder also inserts a third sublayer that performs a multi-head attention on the output of the encoder.

Self-Attention

In the attention function, a query and a set of key-values are positioned as an output, where all components are vectors. The first step for calculating the self-attention, is creating three vectors from each of the encoder's input vectors, for example the embedding for each word. A query vector, a key vector and a value vector is created for each word. These vectors are the result of multiplying the embedding by three different matrices created in the training process. The second step is the calculation of the self-attention score. Every word of a sentence needs to be scored against the rest ones. The next steps include dividing the score by 8, in order to have more stable gradients and using a softmax operation to normalize the score. The sofmax score determines the percentage of a word appearing at a specific position. The final steps include multiplying each value by the softmax score and summing up the weighted value vectors, in order to produce the output of the self-attention layer at the specific position. Learned embeddings are used to convert the input and output tokens to vectors of d dimensions, while softmax function is used to convert the decoder output to next token predicted probabilities.

Aspect Based Approach with BERT

The aspect sentiment classification (ASC) aims to categorize the sentiment polarity expressed about a specific aspect, within a review. The inputs of ASC are the aspect and the review, where the aspect and the output is the a class of polarity (positive, neutral, negative). A sentence could be represented as following x = ([CLS], q1, . . . , qm, [SEP], d1, . . . , dn, [SEP]), where q1, . . . , qm now is an aspect (with m tokens) and d1, . . . , dn is a review sentence containing that aspect. The distribution of polarity is predicted as l4 = softmax(W4 · h[CLS] + b4), where W4 ∈ R 3∗rh and b4 ∈ R 3 (3 is the number of polarities). Softmax is applied along the dimension of labels on [CLS]: l4 ∈ [0, 1]3.

Extended research has been made also on implementing Aspect Based Sentiment Analysis (ABSA) tasks, aiming to automatically extract "aspects" and predict text polarity against those. Aspect-Based Sentiment Analysis has proved to be an extremely powerful tool for analyzing product reviews and tweets, since most of the times user opinions are not consistent towards the subjects of interest by expressing both positive and negative polarity, respectively. Generally, the aspect terms need to be manually labeled on the input dataset before running the APC task, although some researchers are also focused on the challenging task of aspect term extraction (ATE).

So far different methods for sentiment classification have been analyzed. Since social network text data are used, it is obvious that the results derived from a sentiment analysis have to be visually represented, in order to be more easily detectable and with the overall aim of obtaining useful insights.

2.3 Graphs

A number of real-world problems could be described as a diagram of many points connected together with lines. The points could represent people connected with their friends, or cities connected with different roads, or even atoms connected with each other. A mathematical subtraction used for describing this type of situations, is called graph. A graph is consisting of a set of vertices and edges, where an edge connects two vertices in a graph. A graph could be perceived as a pair (V,E), where V is a finite set and E is a binary relation of V. For example, graph G = (V,E) is a collection of V nodes connected by E links. **Figure 9** presents a graph of 6 verticals and 7 edges, where V = {1,2,3,4,5,6} and E = {1,2}, {1,5}, {2,3}, {2,5}, {3,4}, {4,5}, {4,6} :

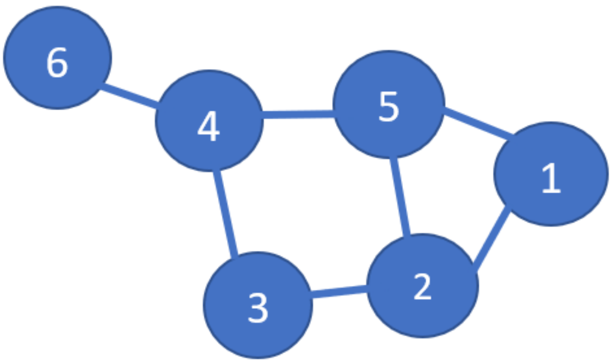

Figure 9: Graph Example

2.3.1 Aspects of Graphs

In the following sections the main aspects of graphs are described:

Path: A path is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list.

Undirected Graph: A graph in which each edge symbolizes an unordered, transitive relationship between two nodes. Such edges are rendered as plain lines or arcs.

Directed Graph/Digraph: A graph in which each edge symbolizes an ordered, non-transitive relationship between two nodes. Such edges are rendered with an arrowhead at one end of a line or arc.

Loop: A loop is a special type of edge that connects a vertex to itself. Loops are not used much in street network graphs.

Degree: The number of edges which connect a node.

In Degree: Number of edges pointing to a node.

Out Degree: Number of edges going out of a node.

Un-weighted edge: A graph in which all the relationships symbolized by edges are considered equivalent. Such edges are rendered as plain lines or arcs.

Weighted edge: Weighted edges symbolize relationships between nodes which are considered to have some value, for instance, distance or lag time. Such edges are usually annotated by a number or letter placed beside the edge. If edges have weights, we can put the weights in the lists. Weight: w: E → R

Tree: An undirected connected graph T is called tree if there are no cycles in it (a cycle in a graph is a trail in which the only repeated vertices are the first and last vertices). There is exactly one simple path between any vertices u and v. Simple path: Simple path is a path in which all the vertices are distinct.

Spanning Tree: A sub graph T of a connected graph G, which contains all the vertices of G and T is called a spanning tree of graph G. It is called spanning tree because it spans over all vertices of graph G.

2.3.2 Social Networks and Graph Theory

Social networks and graph theory are strongly interrelated with each other. Graph theory was first used for describing and analyzing social networks from Moreno, (1934), who tried to picture the population of New York city within a graph. It was proved that social networks, as many other phenomena, can be modeled based on the graph theory. A user is being represented from a single node and the relationship between two users by an edge connecting their nodes. In broad terms, there is a wide range of graphs depicting different relationships: in some graphs, groups of people representing a single node can be noticed, while there are graphs with weighted relationships, showing one-sided relationships such as following someone in Twitter. The advantage of showcasing and analyzing social networks with graphs, comparing to traditional approaches, is that they focus on the relationships between users and reflect their exchanges as a structural element of our world. On the contrary, more traditional approached are focusing on users and their characteristics, for example gallops.

2.3.3 Graph Metrics

There is a broad range of measures and metrics that characterize graphs, from simpler ones, such as the number of vertices and edges, to more complex, such as the degree of centrality, the closeness centrality, the graph density, etc. In this section, the most common graph metrics are presented:

Degree Centrality: Is defined as the number of edges incident upon a link, for example the number of followers that a user has. If the network is directed, there are two different measures of degree centrality, namely, indegree and outdegree. Indegree is the number of edges directed to a single node, while outdegree is the number of edges a specific degree directs to others. For the last cases, the degree is the sum of indegree and outdegree edges.

Betweenness Centrality: Measures the extent to which a vertex lies on paths between other vertices. Vertices with high betweenness tend to have considerable influence within a network, taking advantage of their control over spreading information between others. Diffusion of information lies on specific vertices and for that reason, their removal from the network would cost most disrupt communications.

Closeness Centrality: Allows the detection of vertices with the ability to spread information very efficiently through a network. The closeness centrality of a node measures the average inverse distance to all other nodes. Vertices with high closeness centrality have the shortest distance to all other vertices.

Eigenvector Centrality: It can be seen as the extension of degree centrality. In-degree centrality counts upon every single link a specific node receives. But not all vertices are equivalent: some are more important than others and for that reason a way to state that the endorsements from the important nodes count more, should be found. A node is counted as important, when connected with other important nodes. Eigenvector centrality differs from in-degree centrality, since a node receiving many links does not necessarily mean that it has a high eigenvector centrality. At the same time, a node with high eigenvector centrality is not necessarily highly linked since a node might have few but important connections.

Graph density: is a measure of how many ties between nodes exist comparing to how many ties could have been existed, taken into account the total number of nodes and links. The density of an undirected graph is quite simply calculated as the ratio between the observed number of edges m (the cardinality of the edge set) and the graph maximum size. In this way, we can define how tight-knit or loose-knit a graph is. Tight-knit networks feature a lot of connections and actors can reach others via multiple pathways. Loose-knit networks are more sparsely connected, and actors are only reachable to one another via a very restricted set of pathways.

Clustering Coefficient: Is a measure of the degree to which nodes of a graph tend to group together. Results suggest that especially social networks tend to create tight clusters with high density of ties. Two types of this measure exist, namely the global and the local. The global version gives an overall indication of the clustering in the network, whereas the local indicates the embeddedness of single nodes.

2.3.4 Graph Analysis Software

Graphs, either utilized to represent the relationships derived from social media data or from other kinds of data, can be designed using various software packages, with most popular being Node Excel and Gephi. The next section presents the Gephi software, which was utilized for the purposes of the current research.

Gephi

Gephi is an open-source network manipulation software. It is developed to import, visualize, and filter all types of networks in different graphs. This software can be used for analyzing large networks, from over 20,000 nodes, by running highly configurable layout algorithms that can run on real-time. Several algorithms can run at the same time without blocking user interface. In addition, filters can be applied for selected nodes, edges, or with thresholds. One of its key features is the ability to display the spatialization process, aiming at transforming the network into a map. Many of the current studies showcase dynamic network visualization can offer deep understanding in structure transition or content propagation. For that reason, using Gephi will be used as the main toll for connecting the dots and summarizing the results of this thesis.

As already mentioned, Gephi software provides capability for handling and depicting the relationships between social media users. The main algorithms employed by this software, are briefly described below:

ForceAtlas2 Algorithm in Gephi

ForceAtlas2 is a layout that spatializes a network. "The algorithm forces all nodes to repulse each other, like two neutrons interacting, while edges attract their nodes like magnets" (Jacomy et al., 2014). This rotation of nodes in a stable schema helps in better interpreting data altogether. This force-directed distribution of nodes contributes in stabilizing each node in accordance to all others and its process depends exclusively on the connections between the nodes. The coordinates of each node are not representing any specific variable or attribute, and thus the position of a node itself cannot be interpreted as it must be compared to the others. The end result of this specific technique is a visual interpretation of the structure of a network. The advantage is turning the structural proximities into visual proximities, facilitating the analysis of social networks. According to the pertinent literature, Fleck & Calvert, (2015) proved that proximities reflect communities, while Newman, (2010) concluded that actors have more relations inside their community than outside. Communities are defined as groups with denser relations. It has been proved that force-directed layouts, like ForceAtlas2, are enhancing this measure, with the group of nodes closely connected with each other appearing as communities. Moreover force-directed layouts produce visual densities, translated to structural densities.

Energy Model

Force-directed algorithms rely on certain formulas for the attraction and the repulsion force. The spring-electric layout is inspired by real life and uses the formula of electrically charged particles and the attraction formula of springs. Arafat & Bressan, (2017) created an efficient algorithm however, even though they used the spring metaphor to explain their algorithm, the attraction force was not that of a spring. Noack, (2003) explained the significant role of distance for graph spatialization. In physical systems, the power of forces depends on the distance between different objects, as entities closer to each other attract less and repulse more than more distant entities and vice versa. Noack defined the energy model of a layout as the exponent taken by distance in the formulas used to calculate attraction and repulsion. It is also worth mentioning that clusters denote structural densities when the attraction force depends less on distance, and when the repulsion force depends more on it. **Figure 10** (Jacomy et al., 2014) shows three graphs with different density, from the higher to the lower:

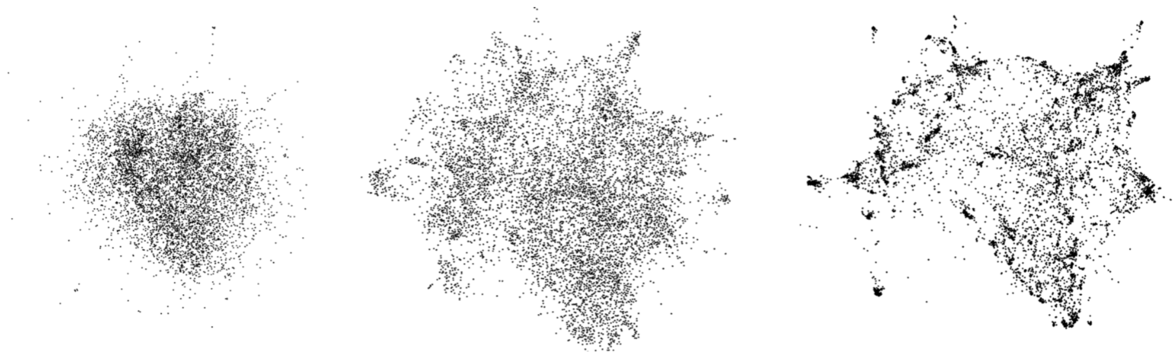

Figure 10: Network Density

Source : Jacomy et al., 2014

CHAPTER 3: Data Collection & Preparation

Online data collection provides researchers with numerous possibilities for scraping information from the web. Among the advantages of collecting information online are the reduced cost, the format flexibility, and the ability to access different populations and a large amount of data within an efficient amount of time. However, as with any method, there are also some potential challenges that researchers have to overcome before meaningful data can be obtained, including rate limitations and possible non-representations. The benefits of accuracy, low cost, speed, and data entry can seem meaningless towards inadequate information. For that reason, when it comes to online data retrieval, special attention should be paid to the sample representativeness. This research is taking advantage of the vast amount of data that can be collected online by managing the limitations mentioned above.

3.1 Twitter API

As already mentioned, the Twitter API was used for the purposes of the current thesis, in order to programmatically retrieve data from Twitter. More specifically, the newest version of Twitter API v2 was used, which is free of charge with the academic access granted. In order to use the API access, the first step was to get access from the Twitter Developing Portal. This process required a new Project or Application to be created and access tokens to be requested. It is also worth mentioning that there are five different keys allowing access the API for extracting Twitter information. Once key tokens are obtained, they can be used directly to any programming environment as the entrance to data available in Twitter. For the purposes of the current thesis, Python was the programming language used for contracting the queries needed and extracting valuable information. The following **Figure 11** taken from *Twitter Web Developer Platform*, (2020) depicts the Academic environment, where usage limitations and Apps can be seen.

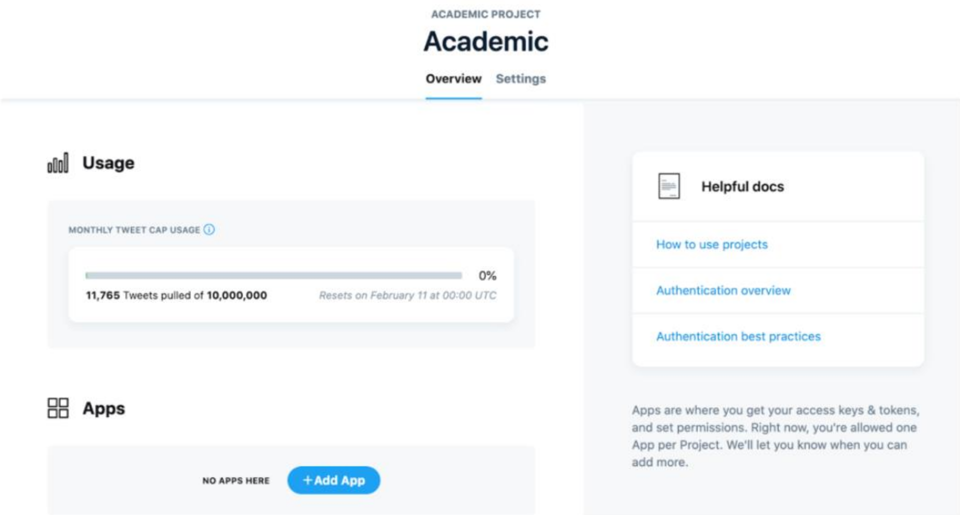

Figure 11: Tweeter Developer Platform

Source : Twitter Web Developer Platform, 2020

3.1.1 Available Endpoints

The API provides access to a variety of resources like tweets, users, lists, trends, places, or media. The following **Table 3** presents a list of categories for different endpoints that the user can access in order to retrieve relevant data, while in the following section a more detailed presentation for each category (or for some of them maybe that used for the purposes of the current research) is performed:

Table 3: Twitter API Endpoints Categories

Twitter API v2 Access_ Endpoints Categories:
➢ Lookup Tweets and lookup users
➢ Search for recent Tweets, or for Tweets from the entire archive of public Tweets
➢ Filter or Sample real-time streaming Tweets
➢ Pull user Tweet and mentions timelines
➢ Understand and manage liking and unliking of Tweets
➢ Understand and manage user following relationships
➢ Understand and manage blocking and unblocking user relationships

Lookup Tweet

The endpoints related to tweets can provide any type of information reflects to a specific tweet, as text or different kind of objects attached to a tweet, including media, place, polls, URLs and time. There is a variety of methods for selecting, delivering, or acting upon a tweet, with this group of REST endpoints returning a Tweet or group of Tweets, specified by a Tweet ID. These endpoints can be used to retrieve up-to-date details on a Tweet or update stored details following a specific event. For these endpoints, the GET HTTP method is utilized, and one or many Tweet objects are returned which deliver fields such as text, date of creation, URLS and more.

Lookup Users

The GET method is used to return information about a user or a group of users, specified by a user ID or a username. The response includes one or more user objects, which retrieves fields like follower count, location, pinned, Tweet ID and profile bio. Responses optionally can be extended in order to retrieve also Tweet objects related to a specific user, including for example tweet text, author, and other fields. These endpoints are commonly used to retrieve up-to-date information about a user or stored details following a compliance event.

Search Tweets

Tweets searching is an important feature used to surface Twitter communications about events and specific topics of interest. This functionality provides great flexibility especially along with filters usage, by ingesting Tweets to find relevant data. Two specific endpoints have been developed for that use: a) Recent search and b) full-archive search. Both queries are created with a set of operators matching with tweet and user attributes, such as message keywords, hashtags, and URLS. The recent search endpoint allows user to programmatically access filtered public Tweets posted over the last week. This endpoint can deliver up to 100 Tweets per request in reverse-chronological order and make queries up to 512 characters long. Full-archive search, only available for the Academic Research product track, allows user to programmatically access public Tweets from archive dating to the first Tweet in March 2006, based on the search query. The last endpoint can deliver 500 Tweets per request in reverse-chronological order and make queries up to 1024 characters long.

Filtered Stream

The filtered stream endpoint enables users to filter live streaming data on public Tweets. This endpoint allows users to listen for specific topics of interest in real-time, monitor communications around competitions and understand how trends are developed.

User Tweet Timeline

The user Tweet timeline endpoints give access to Tweets published by a specific Twitter account. Retrieving a user's Tweets timeline allows an in depth understanding of a person's ideas and needs. This endpoint provides access to a single Twitter account's most recent Tweets, Retweets, replies, similar to a user's profile timeline. It is a REST endpoint and can receive only a single path parameter, indicating a user's ID. The endpoint can return up to 3,200 recent Tweets, Retweets and Replies posted by a single user.

User Mention Timeline

The user mention timeline allows user to request Tweets mentioning a specific Twitter user, similar to what may be seen to a user's notifications for mentions on Twitter. The user mention timeline is a REST endpoint that receives a single path parameter indicating a single user ID. The endpoint can return up to 800 most recent mentions for that user.

Likes

Liking Tweets is one of the most core features available in Twitter for people to engage in public conversations. Likes lookup endpoint can provide a list of accounts liked a given Tweet. That information would be valuable for understanding and mapping someone's preferences and ideas. The liking user's endpoints limits to a total of 100 liking accounts per tweet.

Follows Lookup

Following users is a fundamental action on Twitter for users to be connected and exchange ideas. The follows lookup endpoints enable user to explore relationships between users. There are two REST endpoints returning user objects, with the first representing who a specific user is following and the second reflecting who is following a specific user.

3.1.2 Rate Limitations

Since Twitter API is a widely used platform and thousands of daily requests need to be managed, Twitter has set up specific limitations for not overexploitation of the platform and even use, as per the user packages. The maximum number of requests allowed is based on a time interval. The most common limit interval is fifteen minutes. If for example an endpoint has a rate limit of 900 requests/15 minutes, only 900 requests over a 15-minute interval are allowed. The standard API limits are described in **Table 4** for GET endpoints, the endpoints used for extracting data from the source of Twitter. It is also important to note that user's rate limits are shared across all his applications they authorize, including Twitter application. For example, if a specific user likes 20 tweets on the Twitter mobile application and in parallel likes other 20 tweets on a third-party application within 24-hour period of time, 40 requests in total will be pulled out of the same user rate limit bucket. All request windows in the following table are 15 minutes in length.

Table 4: Tweeter API GET Endpoints

Endpoint	Requests / window per user	Requests / window per app
GET favorites/list	75	75
GET followers/ids	15	15
GET followers/list	15	15
GET friends/ids	15	15
GET friends/list	15	15
GET friendships/show	180	15
GET lists/list	15	15
GET lists/members	900	75
GET lists/members/show	15	15
GET lists/memberships	75	75
GET lists/ownerships	15	15
GET lists/statuses	900	900
GET lists/subscriptions	15	15
GET search/tweets	180	450
GET statuses/lookup	900	300
GET statuses/mentions_timeline	75	0
GET statuses/retweeters/ids	75	300
GET statuses/retweets_of_me	75	0
GET statuses/retweets/:id	75	300
GET statuses/show/:id	900	900

GET statuses/user_timeline	900	1500
GET users/lookup	900	300
GET users/search	900	0

Source : Twitter Web Developer Platform, 2020

3.2 Data Planning

The main goal of this thesis is to investigate any relations between controversy and diffusion of ideas by analyzing a social network. In order to analyze relations between controvert ideas expressed by users in Twitter, first we need to decide: a) what our network should look like? b) what are the assets that should be described in our graph? c) what are the measures we should use? d) what are the raw data we should use as an input for our analysis? To this end, this section focuses on answering the above-presented questions and determining the key elements that will be used for extracting a suitable dataset for the purposes of the current research.

3.2.1 Data Components

Since this thesis focuses on controvert ideas expressed in a social network, the main research interest should be limited to a specific event or topic of interest that rises a certain concern and at the same time, is the fruit for dichotomy to diffuse in the network. The main goal would be to pick a topic of interest with resonance in the network and take advantage of the hype at a specific spacetime on Twitter. Another factor that could point the direction on picking a topic for our research would be the anatomy of Twitter and how the majority of users communicate through it, taking into account that Twitter is a broadly used instrument for driving pressure in political and social aspects of our communities. Considering all the above-mentioned factors, an initial idea was formed, taking advantage of the fact that the “American elections”, a broad enough theme with high interest and spread participation from all Twitter community, was an event that was taking place at the time of implementing this thesis. However, the elections as a theme itself could be vague enough and create setbacks on mapping conversation contributors and filtering their text to identify voices for and against specific parties. For that reason, the topic of interest was decided to be restricted to “Trump candidacy”, in this way all our filters could be concentrated on one specific figure of interest, “Donald Trump”. Our population consists of all users in Twitter that have commented about Trump in relation to the elections from September 2021 until October 2021. After determining the topic of this research and the population, a second pillar in constructing the dataset for our research is understanding what diffusion is and how this movement of information could be captured in the early stages of data collection. For a manner of simplicity, this thesis defines **diffusion** as the effect of successive replies of users on additional tweets, published by other users. After identifying the elements that define our network of interest and consequently the dataset that should be used for the research, it was then easier to distinguish the components of our network. Those components are:

a) **users:** people or groups of people interacting within the network of Twitter

b) **tweets:** the text users are publishing in Twitter

c) **replies:** The replies of users in additional Tweets published

d) **retweets:** republishing Tweets of users

e) **followership:** the relationship between the Users who are following and being followed by other Users

f) **friendship:** the relationship between Users who are liking the content of other Users

g) **controversy:** the effect of a User replying to another one with negative polarity, more details will be shared on controversy in following chapters

h) **diffusion:** the effect of serial replies and retweets of Users under a published Tweet

3.3 Data Modeling

After all components of the dataset have been determined and before proceeding into extracting data, we should consider the restrictions, and the challenges for collecting all the information needed and, more importantly, the structure of the final dataset. Based on the endpoints available and the rate limitations already explained above, **Figure 12** depicts the dataset flowchart. The flowchart includes all components that are needed to be retrieved from Twitter API, their objects, and the relationships between components. It is also important to note that the flowchart is also reflecting the different tables of our dataset and how they are connected. The need for developing this flowchart derives from the limitations occurring from Twitter API. An important limitation is creating build in queries from scratch, or even the inability to modify current endpoints in a way of extracting interconnected information. For example, a user cannot retrieve with a single query information on how many followers a list of users has and who retweeted their published Tweets. In other words, users are obligated by the API to use specific endpoints for extracting a single type of information. Consequently, in order to connect the Tweets of a User's and the identities of people who replied on those, we would first need to use the endpoint "Lookup Tweets" and extract a list of tweets for a specific user and then use this information as an input to the endpoint "Follows Lookup" in order to find also who replied on those Tweets. For all the above reasons, the following flowchart has been created in order to identify the objects for all our valued components and where those objects can be used as inputs for retrieving information on other crucial components.

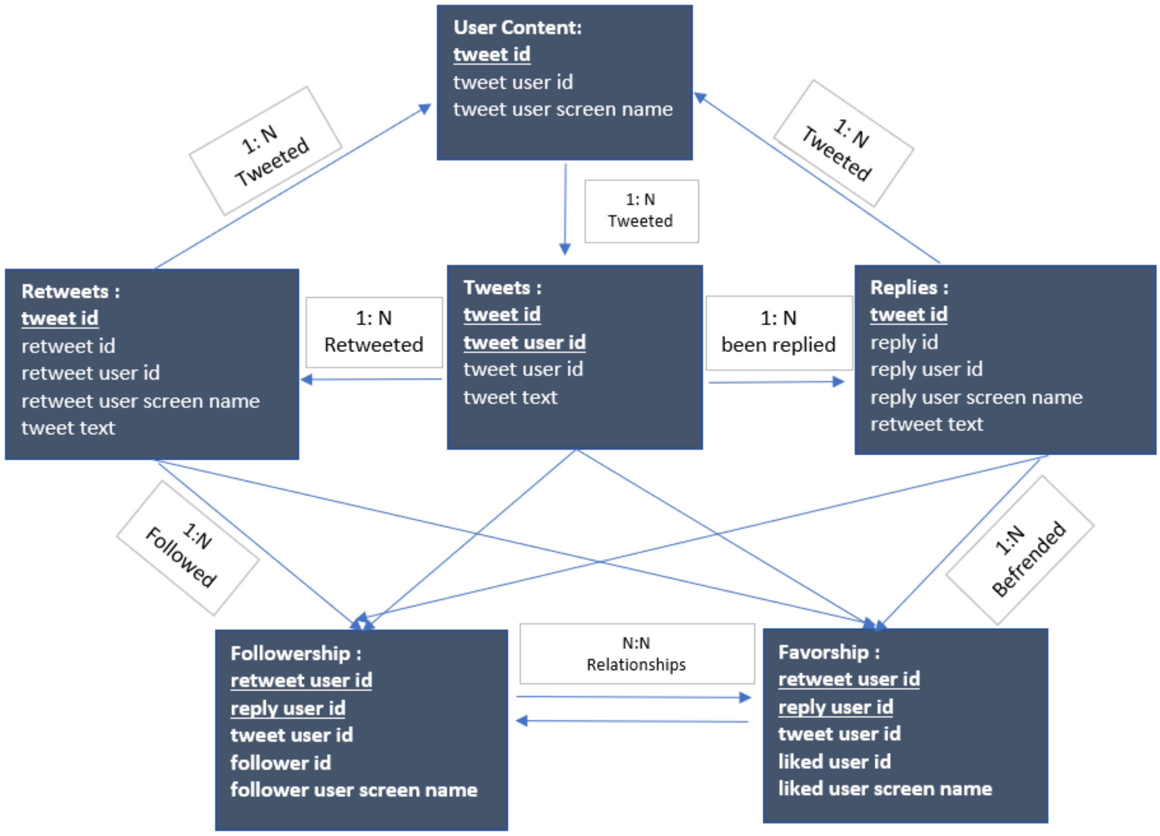

Figure 12: Dataset Flowchart created for this research purposes

3.4 Data Structure

All tables within the above-presented flowchart in **Figure 12** are connected with different types of relationships. Relationships 1:N (one to many) are the most common relationships between tables and describe the situation where one component can generate more than one of other components. For example, in our case, one Tweet can have many different Replies, as well as many different Retweets, but can only be generated by a single User. A second type of relationship between tables is N:N (many to many), with a typical example in our flowchart being the relationship between Followership and Favorship, as many Users can be followed by many others and can follow many others and swimmingly can befriended by many others and vice versa. Another important factor in our data structure are the **keys**, the objects that are being transferred from the parent table to the other generated tables. Those objects are the link from the one table to the other, in order to connect different objects of different tables. For example, table Retweets has borrowed the key object “tweet id” from table “Tweets”, meaning that we can search for all retweets of a single tweet in Tweets and relevant details about those retweets in this table. In this case, key objects are playing a double role, since there also used as the inputs for every single API endpoint to drive information for the child tables. A detailed review on data collection performed for each table, is presented below.

3.5 Data Extraction

3.5.1 Table Tweets & User

The first step of data collection, is the extraction of information about tweets and different relevant objects like text, tweet id, username etc. The extraction process starts with the table containing the more valuable key objects, which are needed as an input for the rest endpoints. Following, the Lookup Tweet endpoint was used, to search for tweets in English, independently of geolocation, using as main parameter one of the most popular hashtags at the time of the data collection, "#trump". A top ten of all related to USA election hashtags can be found in **Figure 13**. Another important filter for our query was also excluding retweets, as our first concern was to collect initial tweets with high exposure in the network and then collect all related retweets and replies upon those tweets. Towards the same direction, this specific query has been developed for providing the most popular tweets within a timeframe of past 2 weeks. From there we can assume that we collected the most retweeted, replied, liked, and disliked tweets within the timeframe available. In order to get the full text of the requested tweets, it was necessary to amend the tweet mode parameter to "extended". After having finalized our query, the next step was to clean our results and save our table in a csv file. Cleaning process is a time-consuming process that was implemented in different levels throughout our data processing. At this specific step, a cleaning process was performed in terms of excluding from the text some Twitter encodes and URLs. All the above-mentioned steps, including the code used, are more thoroughly described in **Figure 14** (*Best-Hashtags.Com*, 2021).

Figure 13: Top 10 USA elections Hashtags

Source : Best-Hashtags.Com, 2021

```python

search_words = "#trump"
date_since = "2020-08-24"


tweets = tw.Cursor(api.search,
              q='#trump-filter:retweets',
              lang="en",
              since=date_since, tweet_mode='extended').items(10000)
tw = []

users_locs = [[tweet.full_text.replace('\n',' ').encode('utf-8'), tweet.id, tweet.user.id, tweet

def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt
def clean_tweets(lst):
    # remove twitter Return handles (RT @xxx:)
    lst = np.vectorize(remove_pattern)(lst, "RT @[\w]*:")
    # remove twitter handles (@xxx)
    lst = np.vectorize(remove_pattern)(lst, "@[\w]*")
    # remove URL links (httpxxx)
    lst = np.vectorize(remove_pattern)(lst, "https?://[A-Za-z0-9./]*")
    # remove special characters, numbers, punctuations (except for #)
    lst = np.core.defchararray.replace(lst, "[^a-zA-Z#]", " ")

    return lst

tweet_text = pd.DataFrame(data=users_locs,
                    columns=["tweettext", "tweet ID","userID", "userscreenname", "location"])

tweet_text.to_csv(r'C:\Users\Chryssa\Documents\tweets10000.csv')
```

Figure 14: Extracting Tweets & Users

3.5.2 Table Replies

In the previous step, we extracted Tweets and User tables, by collecting details on tweet text, tweet id, user id, user screen name, and user location. Replies was the next table for retrieval, with this table being very important in order to acquire the feeling of others on the initial tweets and analyze dynamics of different dialogs that might occur. The overall aim was to find all possible replies, under the limitations applied, for the table of tweets that we had already preserved. For that reason, a list of all tweet IDs and user screen name was needed to be prepared in order to be used as an input to our query. Since there is no given endpoint specifically for retrieving data on replies, the endpoint Tweets Lookup was once again used, applying further filtering with “in_reply_to_status_id_str” attribute. It is worth mentioning that the specific available endpoint does not accept a list of inputs but only single values as inputs. If for example, we wish to find all replies for a list of initial tweets we cannot just insert that list of tweets in our query and get sub lists with replies for every tweet. To achieve this, a “for loop” was needed to be created and the query was needed to be executed for every single tweet on Tweets table. All the above-mentioned steps can be seen in better details on the code within **Figure 15** & **Figure 16**.

```python
import csv
import pandas as pd

#col_names = ["ID", "t.id", "t.user.id","t.text"]
df = pd.read_csv(r'C:\Users\Chryssa\Documents\retweetsall.csv', encoding='utf-8')

IDS = df.loc[:,"ID"].tolist()

user_id = df.loc[:,"t.user.id"].tolist()

print(IDS)
print(user_id)
```

Figure 15: Get the List of Tweet IDs & Screen_names

```python

with open(r'C:\Users\Chryssa\Documents\repliesall10000.csv','w',encoding='utf-8') as f1:
    writer=csv.writer(f1, delimiter='\t',lineterminator='\n')
    for ID, name  in itertools.zip_longest(tweet_id,list) :

        try:
            for tweet in tweepy.Cursor(api.search,q='to:'+name, result_type='recent',since_id=ID

                if hasattr(tweet, 'in_reply_to_status_id_str'):
                  replies = []
                  replies.append(tweet)
                  row = [[ID, tweet.user.id, tweet.user.screen_name.encode('utf-8'), tweet.id, t
                  writer.writerow(row)
                  print(row)
        except tweepy.error.TweepError:
                continue
        continue
```

Figure 16: Get Replies

3.5.3 Table Retweets

Seemingly with previous table for retrieving all possible retweets of our initially gathered tweets, we will need to create a “for loop” and search all retweets for every single tweet id. However, it has been used a different endpoint specifically searching for retweets on the network.

Like with previous Table, for retrieving all possible retweets of our initially gathered tweets, a “for loop” was also needed to be created in order to search all retweets for every single tweet id. However, a different approach was taken: a different endpoint was used for specifically searching retweets on the network. The code used is presented below in **Figure 17**:

```python
while i<6 :
    with open(r'C:\Users\Chryssa\Documents\retweetsall.csv','w',encoding='utf-8') as f1:
        writer=csv.writer(f1, delimiter='\t',lineterminator='\n')
        for ID in IDS :

            try:
                tweets = api.retweets(ID)
                retweets = [[ID, t.id, t.user.id, t.text]for t in tweets]
                writer.writerow(retweets)
                print(retweets)
            except tw.error.TweepError:
                continue
            continue
```

Figure 17: Get Retweets

3.5.4 Table Followership

In this table the main objective was to find the followers of each user that published a tweet. This way, we can spread our network and understand the relations between all users, as well as identifying how far or close they are in terms of connectivity. For this query, the endpoint Follows Lookup was used, again under a loop for searching followers for every single user. For every follower, two objects were extracted, namely the follower id and the follower screen name. One of the biggest challenges using this query, was the rate limitations within the available timeframes. Only 15 followers for a single user could be collected within 15 minutes. Since this was a very time-consuming process, requiring more than 50 days for acquiring all followers, only a part of followers was collected. The code can be found in **Figure 18**.

```
with open(r'C:\Users\Chryssa\Documents\followerstest7.csv','w',encoding='utf-8') as f1:
    writer=csv.writer(f1, delimiter='\t',lineterminator='\n')


    for screen_name in screen_names :
        try:
            for follower in tweepy.Cursor(api.followers, screen_name).items(5):
                followers = []
                followers.append(follower)
                print(follower.id, follower.screen_name)
                followerslist = [[screen_name, follower.id, follower.screen_name]for follower
                writer.writerow(followerslist)
        except tweepy.error.TweepError:
                continue
        continue
```

Figure 18: Get Followers

3.5.5 Table Favorship

The endpoint for favorites can provide another valuable information for introducing a polarity parameter in our network at this early stage of data collection. This query has been used in order to extract information about the users who liked tweets, retweets and replies. As it shown in **Figure 19**, again the query was executed within a loop, until the list of all combined tweets, replies and retweets was finished. Seemingly with previous tables, this query returned a list of different objects describing the users who liked the tweets. Even the fact that our input list was extended, since we combined 3 different tables containing tweet text, the rate limit for getting favorites is more flexible than followers. The rate limits extraction of data up to 75 users per 15 minutes.

```
with open(r'C:\Users\Chryssa\Documents\TEST_2_REST_GetLikes_4_all.csv','w',encoding='utf-8')
    writer=csv.writer(f1, delimiter='\t',lineterminator='\n')

    for screen_name in screen_names :
        try:
            for favorite in api.favorites(screen_name):
                favorites = []
                favorites.append(favorite)
                print(screen_name,favorite.user.screen_name)
                favoriteslist = [[screen_name,favorite.user.screen_name]for favorite in favor
                writer.writerow(favoriteslist)
        except tweepy.error.TweepError:
                continue
        continue
```

Figure 19: Get Favorites

3.6 Data Output

The data collection process described so far, resulted in extracting information for 10,000 Tweets, 98,306 Replies, 101 Retweets, 269,000 Followers, and 560,954 Likes. After further analysis, it appeared that only 5,628 users created the total volume of the activity captured. One of the biggest challenges regarding data collection, was extracting a number of followers for all Users. Due to above mentioned limitations, that was not possible and thus, the parameter of Followers was not used in our analysis. Since all our tables are related to key objects, the next step on data preparation was grouping all information available in order to create a first impression graph. All users who tweeted, replied, or retweeted were combined within a single list and duplicate values were excluded. All tables containing text were also combined, to create a graph. **Figure 20** reflects the total number of Users, nodes, and their connections. The connections are reflecting interactions between users, such as replying or retweeting on someone's tweet. The following chapters are focusing on shading some light on the issue of polarity of text and what would that mean for the interactions between users.

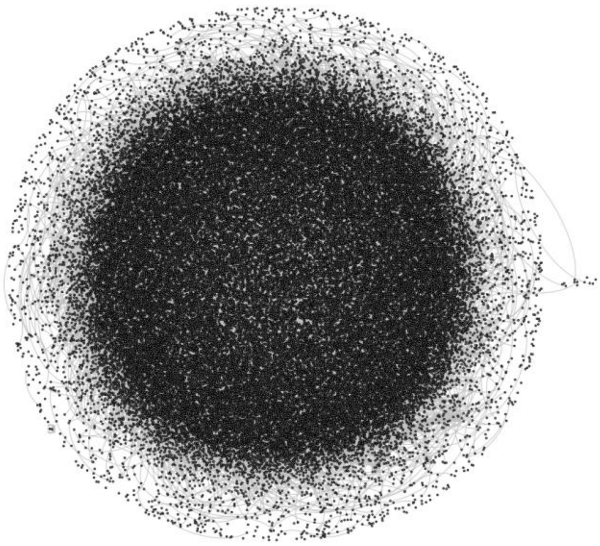

Figure 20: Tableau Results - Graph Visualization

To better understand the content of our tweets, two world cloud graphs were developed in Tableau software, for further investigation. The **Figure 21** is a display of the most used hashtags in our collected tweets, replies and retweets in total. At first reading, controversial hashtags can be seen to pop out, like #FacismisHere and #TrampWonTheArgument. On the other hand, another group of hashtags with more than one interpretation appears to be present. It could be argued that it is difficult to distinguish which argument outweighs in the majority of the networks, by just observing the volume of different hashtags.

Figure 21: Tableau Results - Word Cloud with the Most Commonly used Hashtags

From **Figure 22**, it is even more difficult to understand where the polarity of opinions bends to. Individual words cannot express an opinion about events without being in relation to another sequence of text. To this end, an in-depth analysis of the available text is needed in order to identify the sentiment of tweets and how it is distributed between the nodes of the network.

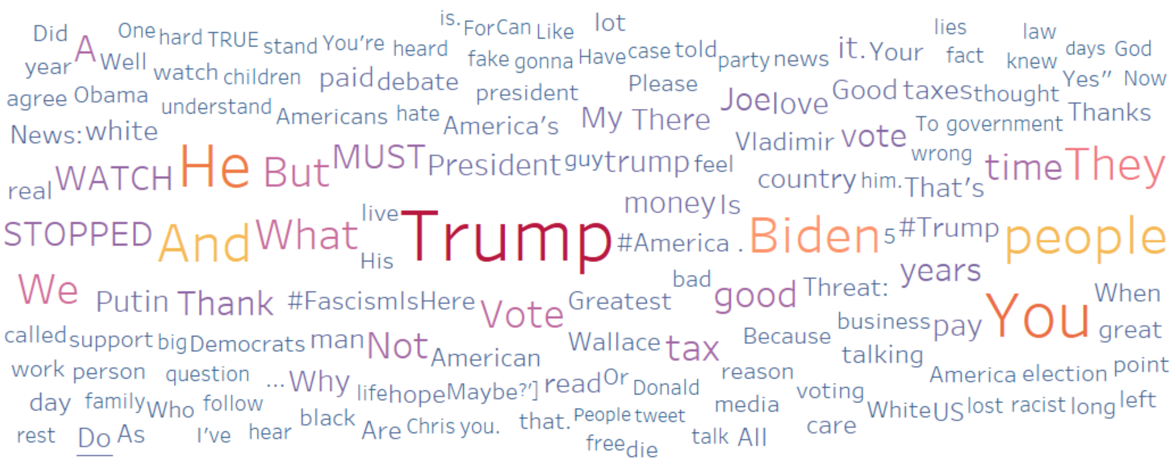

Figure 22: Tableau Results - Word Cloud with the Most commonly used text

CHAPTER 4: Analysis and Results

4.1 Introduction

In this chapter, the analysis of the collected data using sentiment analysis, is presented. Since the overall aim of this thesis is understanding the relation between two different phenomena evolving in social networks, controversy of opinions and how they affect the diffusion of information, a turn of the analysis towards measuring those phenomena was needed, in order for the latter to be reflected in our network of connections and users. After having collected a number of Tweets and their metadata around a bipolar subject like “Trump candidacy”, the next step would be to analyze text polarity and categorize users in two polars: “pro Trump candidacy” and “against Trump candidacy”. Another analysis was then performed on replies and their polarity towards the users who first published tweets. In this way, the edges of the graph were categorized as controvert or condescending connections. In order for the network diffusion to be measured, the dimension of time was split in five different sets of communications. Consequently, a last categorization on our graph edges was implemented, by placing every edge of communication to a specific timeslot. As a final step, a graph analysis using Gephi algorithms was performed, in order to gain a better understanding the phenomena of diffusion and controversy. The above-mentioned graphs, as well as an attempt of their interpretation, is presented in the last section of this chapter.

4.2 Challenges in Sentiment Analysis

The basic tool utilized for the analysis of the collected data, was sentiment analysis. In this section, the challenges that was needed to be tackled for achieving optimum results in sentiment analysis are explained and the sentiment methods used are thoroughly analyzed. Moreover, the sentiment analysis results are presented and a comparison between the different methods used is performed. Finally, the data preparation before proceeding with graph analysis, which is presented in the next chapter, is described.

Based on the pertinent literature, sentiment analysis can be a tricky process, since one of the parameters that can scale up the complexity of this analysis is the text itself. The form of the text, the environment where it has been procreated, as well as the tone or the restrictions under which the specific text has been organized, are all components that could possibly affect the final results of a sentiment analysis and they need to be taken into consideration in order to achieve qualitative results. For example, the language that is usually used in book reviews, or text coming from written surveys or books, differs from one case to another. As already mentioned, in this thesis the text analyzed in this thesis derives from a social media platform. People in social media are more loose with language, while the text in social media platforms is closer to verbal expression of communication than writing. The informality and in many cases the anonymity, provides users the privilege of having a sense of freedom on what they are sharing and how expressive they are. In this kind of base, people are doing vocabulary mistakes, are using short sentences by cutting phrases or implying others and they are using dogmatisms or slang language, when texting. All the above-mentioned characteristics are contributing to the complexity of sensing polarity or objectivity in text. Another important factor that should be taken into consideration, is the political content of our text. Politics, as a subject itself, can be very multidimensional and create tension between interlocutors. When talking about politics, people can be more polarized and intense comparing to other topics of interest. In addition, the bipolarity of our subject itself could also be considered another challenge for sensing true sentiment in the text. It is very difficult to understand the polarity of a sentence and identify a person as a defender or opposer, just from reading a sentence. Taking as an example the phrase “I believe Trump has destroyed America”, the sentiment would be utterly negative based on the word’s usage, although for a Trump opposer this sentence would mean to have a positive meaning. As a consequence, it is vital to identify the subject of interest within our text and then categorize users in different polars. For that reason, the option of using an aspect-based sentiment analysis was thoroughly explored and described within this chapter, with the overall aim of enhancing the possibility of capturing nuances about objects of interest. Another challenging decision was the choice of the categories for our sentiment analysis, while the most commonly used classification for text includes the positive, negative and neutral categories. For simplicity reasons, it was decided for the purposes of current thesis to be more beneficiary keeping only 2 categories, positive and negative.

4.3 Executing Sentiment Analysis

As already mentioned, different sentiment analysis algorithms have been applied and compared for the purposes of the current research, before concluding to the best performing algorithms the results of which were finally visualized, using graphs. At first, Naive Bayes & Stochastic Gradient Descent Classifier algorithms were used from scikits.learn. Sklearn is a software machine learning library for Python programming language, featuring various classification and regression algorithms like those used in this thesis.

4.3.1 Naïve Bayes Algorithm Execution

In order for the above-mentioned algorithm to be executed, finding and uploading a training dataset was required. The training dataset used, was extracted from Kaggle and contained 10,000 tweets, their sentiment, the user screen name, and text. The code used, is presented below in **Figure 23** :

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 24 21:46:39 2020

@author: Chryssa
"""

import pandas as pd
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn import naive_bayes
from sklearn.metrics import roc_auc_score
import numpy as np


df = pd.read_csv(r'C:\Users\Chryssa\Documents\My Thesis\DATASET\CSV_gEPHI\Training_Data_10k.csv'
                 names=['liked', 'user_name', 'txt'])

print(df)

```

Figure 23: Inserting Training Dataset

```
Atlanta.
9996        1   ...   @Beverleyknight i'm very VERY hungover.
Apart ...
9997        0   ...
Finally home
9998        0   ...   @Aphrosie It would still involve me
standing u...
9999        0   ...   Is sat around bored looking at expensive
cloth...

[10000 rows x 3 columns]
```

Figure 24: Training Dataset Print

Data cleaning was performed by using an available list of stopwords in sklearn library. Stopwords are considered to be a list of words that we wish to scrap from our text, since they do not reflect polarity and their existence only slowdowns the algorithm performance. The word "to" for example, does not express a sentiment and our algorithm would have passed it by for the next set of words. The next step was to vectorize our text. In order for the algorithm to estimate the probability of sentiment for all features within every tweet, it is vital to transform text into numbers. The Scikit-learn's Tfidftransformer converts a collection of raw documents to a matrix of TF-IDF features. TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency (Aizawa, 2003). It is a common algorithm for transforming text into a meaningful representation of numbers, which is then used to fit a machine algorithm for prediction. Practically all words in our dataset are being counted down and TF-IDF algorithm assigns to each word a specific identification number, based on the frequency of each word in the whole dataset. The process is described in **Figure 25**:

```
stopset = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(use_idf=True, lowercase=True, strip_accents='ascii',
                             stop_words=stopset)

y = df['liked']
x = vectorizer.fit_transform(df['txt'].values.astype('U'))
x = x.toarray()


x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42)
```

Figure 25: Vectorizing Text Data

As a next step, the dataset was distributed in train and test set. This is an important step on the procedure, since splitting the dataset can help detecting if a model suffers from underfitting or overfitting. The first occurs when a model is being unable to encapsulate relations between data and the second usually takes place when a model learns existing relations among data and noise.

In this specific example, as can be seen in **Figure 26**, the data are being distributed randomly in 2 subsets, train and test data. When the size of the test data is not applied, it is automatically set to 0.25. Moreover, the parameter "random_sate" controls the shuffling applied to the data before applying the split.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42)
```

Figure 26: Dataset Splitting

As shown in **Figure 27**, following step was fitting train and test data and evaluating the model by calculating the acuracy score and confusion matrix.

```
clf = naive_bayes.MultinomialNB()
clf.fit(x_train, y_train)


from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix


y_pred = clf.predict(x_test)

acc_score = accuracy_score(y_test, y_pred)
print(acc_score)


conmat = np.array(confusion_matrix(y_test, y_pred, labels=['0','1']))
confusion = pd.DataFrame(conmat, index=['negative', 'positive'],columns=['predicted_negative','p
print(confusion)

```

Figure 27: Executing Naïve Bayes Algorithm

4.3.2 Algorithm Evaluation

Theoretically, after the classification process, there were four possible categories that could describe our results. Based on the test set, True positives and False positives could be identified, when the actual positive and negative tweets were predicted to be positive by the model, respectively. Seemingly, False negatives were the tweets that were actually positive, but the model wrongly predicted them as negative, while True negatives, were the actually negative tweets that were correctly predicted to be negative by the model. On the other hand, Accuracy score is the fraction of samples predicted correctly. All the above-mentioned information, is presented in **Figure 28** :

<table>
<tr><td colspan="2" rowspan="2">Confusion Matrix</td><td colspan="2">Predicted</td></tr>
<tr><td>Negative</td><td>Positive</td></tr>
<tr><td rowspan="2">Actual</td><td>Negative</td><td>True Negative</td><td>False Positive</td></tr>
<tr><td>Positive</td><td>False Negative</td><td>True Positive</td></tr>
</table>

Figure 28: Confusing Matrix Example

Source : Science Direct, 2021

```
accurasy_score :
0.7516
confusion_matrix:
          predicted_negative  predicted_positive
negative                1870                   0
positive                 620                   9
['0']

In [61]:
```

Figure 29: Naïve Bayes Evaluation

Based on **Figure 29**, it could be argued that despite the relatively high accuracy score (higher than 60%), a significant inability of the model to predict positive tweets, can be identified. This conclusion can be easily drawn by taking a further look on False negative number of Tweets, since only 9 positive tweets were predicted correctly. This observation directed next steps towards optimizing the model, while a conclusion regarding the presence of model bias was derived.

4.3.3 Stochastic Gradient Descent Classifier Algorithm Execution

Moving on with the implementation of the algorithm for performing sentiment analysis, the first step was updating the training dataset with a new list of tweets form Kaggle. As already mentioned, the new dataset used for the specific method had equally distributed labels among the tweets, with 50% of the total tweets having positive polarity and the other 50% having negative polarity. Similarly, to the previous sentiment attempt, the next steps included vectorizing the dataset and splitting the data in train and test with random state. The code used for above purposes is presented in **Figure 30.**

```
file_path = r'C:\Users\Chryssa\Documents\My Thesis\DATASET\CSV_gEPHI\Train_Data_50&50.csv'
df = pd.read_csv(file_path, names = ['liked', 'user_name', 'txt'])

print(df)

from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer

token = RegexpTokenizer(r'[a-zA-Z0-9]+')
cv = CountVectorizer(lowercase=True,stop_words='english',ngram_range = (1,1),tokenizer = token.tokenize)
text_counts= cv.fit_transform(df['txt'].apply(lambda x: np.str_(x)))

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    text_counts, df['liked'], test_size=0.3, random_state=1)
```

Figure 30: Importing Training Dataset & Text Tokenization

As shown in **Figure 31**, 499,810 tweets were available for further processing, along with their sentiment and text:

```
        liked       user_name                                             txt
0           0      SnapItOut   Gaaah  So annoyed I can't go to West End Live
1           0     sososophie  Aye yi yi... Today starts my run of crazy work...
2           0        Siafu25  I don't have a phone yet and I'm  sorry you ha...
3           0       ragenpry  @immissworld I hope you had fun. Im kicking my...
4           0    Mrs_Dillard  What type of man won't give hIs child his addr...
...       ...            ...                                              ...
499805      1    Miss_Leerey               yup.tonight is the prom  im excited
499806      1     adamxpx182                    Fanboys was really really good
499807      1  vagueismyname  After a month of asking to get a new pair of j...
499808      1    MadisenHill  booked a commercial! pretty stokedd. writing a...
499809      1  OhNo_ItsAlice      Bank holiday mondaaaaaay   Exams tomorrow  D:

[499810 rows x 3 columns]
```

Figure 31: Training Dataset Print

A subsequent step was the fitting of the model with maximum passes of training data (aka epochs) 1,000 times. Furthermore, the criteria for the training to stop were set as follows: loss > best_loss – tol. By setting with_mean to False, we respectively set the mean μ=0 and the std.=1, assuming that the dataset features were coming from the normal Gaussian distribution. This implies that we are processing sparse matrices, which contain a significant number of zeros. All the above-mentioned parameters and the code used, are presented in the following **Figure 32.**

```
# Always scale the input. The most convenient way is to use a pipeline.
clf = make_pipeline(StandardScaler(with_mean=False),
                    SGDClassifier(max_iter=1000, tol=1e-3))
clf.fit(X_train, y_train)
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('sgdclassifier', SGDClassifier())])

movie_array = np.array(["b'Liberals are attacking #trump for misleading us when #COVID19 struck. In the recording he

movie_vector = cv.transform(movie_array)

print (clf.predict(movie_vector))
```

Figure 32: Executing SGD Algorithm

As a last step, the evaluation metrics were also calculated, including the accuracy score and the confusion matrix. The code used is presented in **Figure 33**. A comparison of this new model with the previous sentiment attempt was also conducted, in terms of the evaluation metrics calculated for both models.

```
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix

predicted= clf.predict(X_test)
print(predicted)
print(metrics.accuracy_score(y_test, predicted))

conf_mat = confusion_matrix(y_test,predicted)

print(conf_mat)

####################

conmat = np.array(confusion_matrix(y_test, predicted, labels=['0','1']))
confusion = pd.DataFrame(conmat, index=['negative', 'positive'],
                         columns=['predicted_negative','predicted_positive'])
print(confusion)
```

Figure 33: SGD Algorithm Evaluation

4.3.4 Algorithm Evaluation

Based on the model results, which are presented in **Figure 34**, the accuracy score of the SGD-Classifier was similar with the Naïve Bayes algorithm score. However, the confusion matrix revealed the superiority of the SGD-Classifier, since the 90% of tweets were predicted correctly (based on the predicted results of test data). Consequently, the SGD Classifier, along with the use of an updated dataset which had equally distributed labels among the tweets, outperformed the Naïve Bayes algorithm, obtaining more accurate results.

```
accuracy_score:
0.698418732451665
confusion_matrix:
          predicted_negative  predicted_positive
negative               53414               20746
positive               22920               51309

In [64]:
```

Figure 34: SGD Algorithm Evaluation Results

4.3.5 BERT Model Execution

Before proceeding with data preparation, some of the key libraries imported should be mentioned. TensorFlow is an open-source library for developing and training ML models. Another important library used is Keras, a high-level neural network API written in Python, designed to provide experimentation with deep neural networks for solving machine learning problems. The core data structures of Keras are layers and modules, and it can be run on TPU or GPU. As shown in the following **Figure 35**, both above mentioned libraries were used for running the model.

```
[ ] import os

    import numpy as np
    import pandas as pd

    import tensorflow as tf
    import tensorflow_hub as hub

    from keras.utils import np_utils

    import official.nlp.bert.bert_models
    import official.nlp.bert.configs
    import official.nlp.bert.run_classifier
    import official.nlp.bert.tokenization as tokenization

    from official.modeling import tf_utils
    from official import nlp
    from official.nlp import bert

    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder
```

Figure 35: Libraries Input

On the first attempt implementing BERT, a new dataset from Kaggle was used for training the classifier, since input requirements differ from previous algorithms. The dataset consisted of a total 1.6 million tweets and their target polarity of the tweet (values 0=negative or 1=positive), tweet id, date of tweet posted, flag representing the query, username, and text. As shown in **Figure 36** , the training dataset used was evenly distributed between positive and negative tweets, in order to avoid bias.

```
[ ] df.target.unique()

    classes = df.target.unique()
    counts = []

    for i in classes:
      count = len(df[df.target==i])
      counts.append(count)

    plt.bar(['negative', 'positive'], counts)
    plt.show()
```

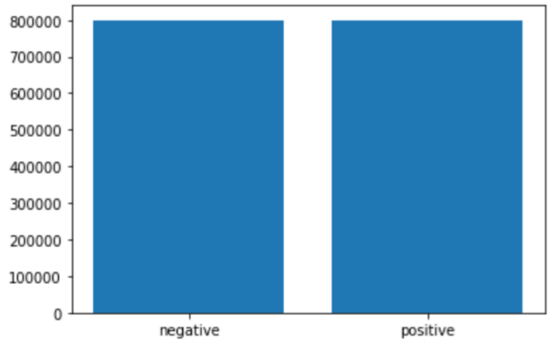

Figure 36: Training Dataset Print

In order to save time in the training process, only the 5% of the initial dataset was kept for further splitting in training set (80%) and test set (20%), as shown in **Figure 37**.

```
[ ] sample_size = int(len(df)*0.05)
    sampleDf = df.sample(sample_size, random_state=23)
    wkd = sampleDf.wkd.values
    x = sampleDf.text.values
    y = sampleDf.target.values
    wkd_train, wkd_test, x_train, x_test, y_train, y_test = train_test_split(wkd, x, y, test_size=0.20, random_state=32)
```

Figure 37: Training Dataset Split

As **Figure 38** shows, the categorical labels of each tweet under column “target” were encoded into numeric values between o-n_classes-1, where n is the number of distinct labels. When labels were repeated within the dataset, the same values were assigned.

```
encoder = LabelEncoder()
encoder.fit(y)
encoded_Y_test = encoder.transform(y_test)
encoded_Y_train = encoder.transform(y_train)


dummy_y_test = np_utils.to_categorical(encoded_Y_test)
dummy_y_train = np_utils.to_categorical(encoded_Y_train)
```

Figure 38: Encoding Training Dataset

In addition, the text data were tokenized, using functions borrowed from official.nlp.bert package. These functions are depicted in the following **Figure 39**.

```
[ ] vocab_file = bert_layer.resolved_object.vocab_file.asset_path.numpy()
    do_lower_case = bert_layer.resolved_object.do_lower_case.numpy()
    tokenizer = tokenization.FullTokenizer(vocab_file, do_lower_case)
```

Figure 39: Text Tokenization

Since BERT uses masks to replace tokens for the training procedure, it was important to also add tokens for ‘CLS’ and ‘SEP’ representations, as shown in **Figure 40**.

```
[ ] tokenizer.convert_tokens_to_ids(['[CLS]', '[SEP]'])
```

Figure 40: Converting Tokens into IDs

The next steps included the preparation of the inputs for BERT model, namely the tokens and the input mask while the input type was remade into functions for easier use. Moreover, a max_sequence_length was set for the inputs, as shown in **Figure 41**.

```
[ ] def encode_names(n, tokenizer):
       tokens = list(tokenizer.tokenize(n))
       tokens.append('[SEP]')
       return tokenizer.convert_tokens_to_ids(tokens)

    tweets = tf.ragged.constant([encode_names(n, tokenizer) for n in x_train])
    cls = [tokenizer.convert_tokens_to_ids(['[CLS]'])]*tweets.shape[0]
    input_word_ids = tf.concat([cls, tweets], axis=-1)

    lens = [len(i) for i in input_word_ids]
```

```
[ ] max_seq_length = max(lens)
    print('Max length is:', max_seq_length)
```

Figure 41: Preparing Bert Model Inputs

After setting the training parameters, the model ran through the selected dataset 3 times (as the number of epochs) and 177,854,979 neurons were used for solving the classification problem. Every epoch consumed more than 1 hour for training, which is explains why only the 5% percent of the total dataset was initially used. Based on the evaluation results presented in the following **Figure 42**, the utilization of BERT model raised the testing accuracy at 82.57%, providing better results comparing to the already used Naïve bayes and SGD algorithms.

```
[ ] history = model.fit(X_train,
                        dummy_y_train,
                        epochs=epochs,
                        batch_size=batch_size,
                        validation_data=(X_test, dummy_y_test),
                        verbose=1)

Epoch 1/3
4000/4000 [==============================] - 3837s 956ms/step - loss: 0.5498 - accuracy: 0.7085 - val_loss: 0.4057 - val_accuracy: 0.8167
Epoch 2/3
4000/4000 [==============================] - 3825s 956ms/step - loss: 0.3605 - accuracy: 0.8434 - val_loss: 0.4233 - val_accuracy: 0.8246
Epoch 3/3
4000/4000 [==============================] - 3826s 956ms/step - loss: 0.2682 - accuracy: 0.8919 - val_loss: 0.4727 - val_accuracy: 0.8257
```

```
[ ] loss, accuracy = model.evaluate(X_train, dummy_y_train, verbose=False)
    print("Training Accuracy: {:.4f}".format(accuracy))
    loss, accuracy = model.evaluate(X_test, dummy_y_test, verbose=False)
    print("Testing Accuracy:  {:.4f}".format(accuracy))

Training Accuracy: 0.9391
Testing Accuracy:  0.8257
```

Figure 42: Executing BERT Model

The predicted results can be seen in ***Figure 43***. The final step included the execution of the model for predicting the polarity of Replies. By observing the text against predicted polarity, the majority of sentiment predictions appeared to be correct. However, the results cannot be considered of a great value towards identifying the polarity against the users to whom a specific reply directs to. Identifying the polarity of a specific reply, does not necessarily lead to a safe conclusion that somebody agrees or disagrees with another user. For example, a specific reply could judge and criticize Trump having a negative sentiment, and at the same time, could agree with the initial user who tweeted a similar content. With the main objective of this research being the identification of controversy within a user's reply, it was important to captivate the sentiment of replies towards a specific subject, namely the user to whom a specific reply refers to. To this end, another sentiment approach was attempted, developing the already used BERT model within a feature-based framework. The transition from the conventional BERT model to the more sophisticated Aspect Based BERT model, is described in the following paragraphs:

```
inputs = bert_encode(string_list=list(text),
                     tokenizer=tokenizerSaved,
                     new_feature_class_count=7,
                     max_seq_length=240)
prediction = model.predict(inputs)
print (prediction)

[[0.01254539 0.98745465]
 [0.30557767 0.69442236]
 [0.0175003  0.9824997 ]
 ...
 [0.78935206 0.21064794]
 [0.30557752 0.6944225 ]
 [0.00656951 0.99343055]]
```

Figure 43: Predicting Sentiment for "Replies" Table

4.3.6 BERT Algorithm and Aspect Based Analysis

In this research aspect-based sentiment analysis is being implemented in our dataset with regard to find the sentiment user A develops about another user B. The second user is the aspect we need to identify within a Tweet in order to examine polarity. As it happens for all Tweet Replies to start with the user_screen_name someone directs his Tweet, an excel formula was executed in order to extract the aspect element from text (**Figure 44**). After labeling the aspect terms on the input dataset, APC task takes place.

tweet_user_r	reply_user_id	reply_user_s	reply_id"	text	left_text	aspect	**Target**	right_text
Bayernphan	"44461687"	Bayernphan	"1311003248	"b'@deng	"b'	@	denglercarlos	denglercarlo:
Bayernphan	"3060250335"	denglercarlos	"1310978283	"b"@Bay	"b"	@	Bayernphan	Bayernphan
kk131066	"119976813283	kk131066	"1310964512	"b'RT @O	"b'RT	@	OlderthanU70	OlderthanU7
kk131066	"53651738"	OlderthanU70	"1310963949	"b'@kk13	"b'	@	kk131066	kk131066 \xf
kk131066	"129500902341	LeftyLiteral	"1310963727	"b'@kk13	"b'	@	kk131066	kk131066 We
kk131066	"119976813283	kk131066	"1310954567	"b'RT @B	"b'RT	@	BettyGooch2	BettyGooch2
kk131066	"129655904088	BettyGooch2	"1310954330	"b'@kk13	"b'	@	kk131066	kk131066 #R
kk131066	"129764934923	LisaUshman	"1310949526	"b'@kk13	"b'	@	kk131066	kk131066 Th
kk131066	"119976813283	kk131066	"1310949141	"b'RT @Li	"b'RT	@	LisaUshman	LisaUshman:

Figure 44: Creating Aspect input in "Replies" Table

This thesis followed the LCF-ATEPC framework, described in the following **Figure 45**. This framework contains two differet pretrained BERT layers, the global contect feature generator (GCFG) and the local contect feature generator (LCFG). Feature interctive learning (FIL) layer combines BERT layers and predicts aspect polarity. In order for this process to be conducted, the input sequence is tokenized into tokens, where each token is annotated with ATE and APC labels. The firt indicates whether the token belongs to an aspect tern, while the APC label reveals the sentiment of an aspect term.

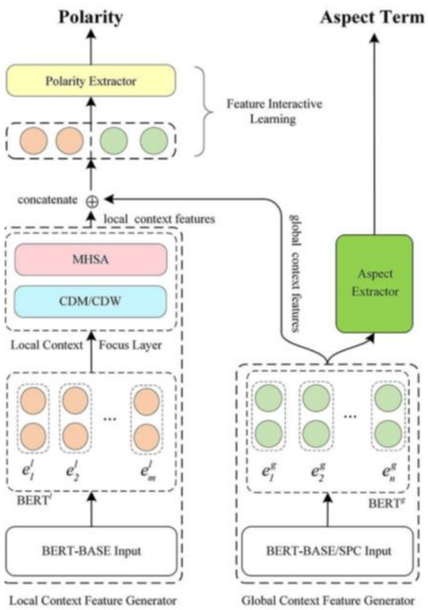

Figure 45: LCF-ATEPC Framework

Source : Villaseñor Rodríguez, 2014

To identify the sentiment tendency towards the "aspects", the Attentional Encoder Network (AEN) was again applied using the pretrained BERT model. The AEN BERT model consists of an embedding layer, an attentional encoder layer, a target-specific attention layer and an output layer. This specific methodology was applied for the purposes of the current thesis, using three different datasets: SemEval 2014 Task 4 (Pontiki et al., 2014) datasets, containing Restaurant reviews and Laptop reviews and ACL 14 Twitter dataset, gathered byDong et al. (2014). The same datasets were also used as training input and were labeled with three sentiment polarities, namely positive, neural and negative. For the natural language processing, the Pytorch library was used, with the source code having been borrowed by *Exploiting Fast LCF and BERT for Aspect-Based Sentiment Analysis (Official Implementation of LCF-BERT)*, 2020). The code was developed based on several surveys as per Qiu et al., (2020), Zhang & Corporation, (2011), and Young et al., (2018).

As shown in **Figure 46** , the model consisted of the -already explained- 3 different classes, namely BERT SPC, AEN BERT, and LCF BERT. Based on the number of epochs, the model ran through the training set for five times in total. The final score of the model was calculated for each training dataset, with the accuracy score of the LCF BERT for "Laptop" dataset being 81.03% and for "Restaurant" dataset 88.83%.

```
model_classes = {
    'bert_spc': BERT_SPC,
    'aen_bert': AEN_BERT,
    'lcf_bert': LCF_BERT,
    # default hyper-parameters for LCF-BERT model is as follws:
    # lr: 2e-5
    # l2: 1e-5
    # batch size: 16
    # num epochs: 5
```

Figure 46: Aspect Based – BERT Model Classes

The final step was applying the model on Replies table and identifying the sentiment of each reply towards the receivers of specific comments in the network. The given inputs, namely the aspect, the text_left and the text_right, were encoded and tokenized into text_bert_indices and bert_segments_ids, before running the model on our data. All the above-mentioned procedures and the code used, are presented in **Figure 47**.

```python
def prepare_data(text_left, aspect, text_right, tokenizer):
    text_left = text_left.lower().strip()
    text_right = text_right.lower().strip()
    aspect = aspect.lower().strip()

    text_raw_indices = tokenizer.text_to_sequence(text_left + " " + aspect + " " + text_right)
    aspect_indices = tokenizer.text_to_sequence(aspect)
    aspect_len = np.sum(aspect_indices != 0)
    text_bert_indices = tokenizer.text_to_sequence('[CLS] ' + text_left + " " + aspect + " " + text_right + ' [SEP] ' + aspect + " [SEP]")
    bert_segments_ids = np.asarray([0] * (np.sum(text_raw_indices != 0) + 2) + [1] * (aspect_len + 1))
    bert_segments_ids = pad_and_truncate(bert_segments_ids, tokenizer.max_seq_len)

    return text_bert_indices, bert_segments_ids
```

Figure 47: Data Preparation & Tokenization

As shown in the following **Figure 48**, the results printed were including the user_screen_name, the probability and the sentiment for all replies. It is important to notice that ABSA BERT sentiment results are different from BERT results obtained by the previous sentiment attempt. In conclusion, the aspect parameter can indeed have enormous diversions in the final classification of polarity. By observing the sentiment output against the text and the aspects provided, it is now more obvious that replies with negative polarity are showing controversy against the tweets in which those replies are targeting to.

```
loading model bert_spc ...
JamieAllingham1 ('t_probs = ', array([[0.1406798 , 0.39247784, 0.4668424 ]], dtype=float32)) ('aspect sentiment = ', array([1]))
Dolphfinn33 ('t_probs = ', array([[0.17067254, 0.27482536, 0.5545021 ]], dtype=float32)) ('aspect sentiment = ', array([1]))
JonnyDouglas ('t_probs = ', array([[0.13043939, 0.31879935, 0.5507612 ]], dtype=float32)) ('aspect sentiment = ', array([1]))
AximNYC ('t_probs = ', array([[0.40245888, 0.25892875, 0.33861235]], dtype=float32)) ('aspect sentiment = ', array([-1]))
robinsnewswire ('t_probs = ', array([[0.10387344, 0.3682964 , 0.5278302 ]], dtype=float32)) ('aspect sentiment = ', array([1]))
PaulGregory2173 ('t_probs = ', array([[0.46700993, 0.23541597, 0.29757416]], dtype=float32)) ('aspect sentiment = ', array([-1]))
ware_koko ('t_probs = ', array([[0.18599783, 0.27842385, 0.5355783 ]], dtype=float32)) ('aspect sentiment = ', array([1]))
```

Figure 48: Aspect-Based BERT Model Results

The same code was used for classifying sentiment of initial ten thousand tweets, using as an aspect input “Trump”. As shown in **Figure 49** , the most important change on the data input was replacing the aspect with the word “Trump” for all the series of rows in tweets table. As a result, the polarity of the initial tweets was finally predicted towards “Trump”. Since training and test set remained the same and no other variables in the model were triggered, the score results were also identical to those already presented in the previous example of ABSA BERT analysis for Replies table.

left_text	aspect	right_text	tweettext	userscree	Sentiment
b'Just some la	Trump	some law-ab	b'Just some l	Bayernphan	array([-1]))
b'@IStarshepp	Trump	arsheppygirl I	b'@IStarshe	kk131066	array([-1]))
b'@realDonal	Trump	rump We the	b'@realDona	JanettThinks	array([-1]))
b'Is this part o	Trump	his part of #T	b'Is this part	JamieAllingh	array([0]))
b"#200KDead/	Trump	didn't have t	b"#200KDea	thereseflana	array([-1]))
b'@realDonal	Trump	rump Drug ac	b'@realDona	Dolphfinn33	array([1]))
b'@NateSilver	Trump	eSilver538 Th	b'@NateSilve	DoodyGiulia	array([-1]))
b"@Deven_In	Trump	el I surprised	b"@Deven_I	Duktiamat	array([-1]))
b'Sure...#Fake	Trump	tempts to ma	b'Sure...#Fak	bindyb123	array([-1]))
b'Liberals are	Trump	tacking #Trun	b'Liberals are	graciela92pe	array([-1]))

Figure 49: Creating Aspect Input in “Tweets” Table

4.4 Data Preparation

Before proceeding with graph analysis, all the data collected from extraction and sentiment analysis results were processed and manipulated in a readable way for Gephi, the graph analysis tool used for sentiment analysis results visualization. All graphs developed were related to the main goals of the current research, including: a) to represent a social network echoing the polarity of tweets and their replies with subject "Trump's candidacy for USA elections in 2020", b) to analyze the communities created and identify the key leaders spreading influence within the network, c) to define controversy between opinions in the network and analyze the phenomenon and d) to identify any possible relations or coefficiency between controversy and diffusion of information in the network. **Figure 50** contains all tweet replies, the user_screen_name who posted the reply, the user_screen_name of the initial tweet other users replied to, the target_ user_screen_name to whom a reply refers to and the sentiment results from ABSA BERT analysis for every single reply. Two main tables were constructed as an input for Gephi analysis tool, nodes and edges table. For that reason, the rest columns, as well as the information in **Figure 50** were further transformed into attributes describing either nodes or edges behavior in the network.

text	tweet_user_nar	reply_user_id"	reply_user_scre	reply_id"	text	left_text	aspect	Target	right_text	Polarity	Sentiment				
"b'RT @poli	Bayernphan	"44461687"	Bayernphan	"1311003248	"b'@deng	"b'	@	denglercarlo	denglercarlo:	"1311003:	array([[0.2	0,161933	0.6033991	dtype=floa	array([1]))
"b'@kk131(	Bayernphan	"3060250335"	denglercarlos	"1310978283	"b"@Bay	"b"	@	Bayernphan	Bayernphan	"1310978:	array([[0.0	0,059374	0.8586946	dtype=floa	array([1]))
"b'RT @jess	kk131066	"119976813283	kk131066	"1310964512	"b'RT @O	"b'RT	@	OlderthanU7	OlderthanU7	"1310964:	array([[0.0	0,174383	0.7836520	dtype=floa	array([1]))
"b'RT @jess	kk131066	"53651738"	OlderthanU70	"1310963949	"b'@kk13	"b'	@	kk131066	kk131066 \xf	"1310963:	array([[0.0	0,074036	0.8759807	dtype=floa	array([1]))
"b'@kk131(	kk131066	"129500902341	LeftyLiteral	"1310963727	"b'@kk13	"b'	@	kk131066	kk131066 We	"1310963'	array([[0.0	0,078447	0.8900648	dtype=floa	array([1]))
"b'RT @Shie	kk131066	"119976813283	kk131066	"1310954567	"b'RT @B	"b'RT	@	BettyGooch2	BettyGooch2	"1310954:	array([[0.0	0,269412	0.6846761	dtype=floa	array([1]))
"b'@kk131(	kk131066	"129655904088	BettyGooch2	"1310954330	"b'@kk13	"b'	@	kk131066	kk131066 #R	"1310954:	array([[0.0	0,347429	0.6125118	dtype=floa	array([1]))
"b'RT @Me	kk131066	"129764934923	LisaUshman	"1310949526	"b'@kk13	"b'	@	kk131066	kk131066 Th	"1310949:	array([[0.0	0,11783	0.8493864	dtype=floa	array([1]))
"b'@kk131(	kk131066	"119976813283	kk131066	"1310949141	"b'RT @Li	"b'RT	@	LisaUshman	LisaUshman:	"1310949	array([[0.0	0,068526	0.9236215	dtype=floa	array([1]))
"b"IN CONC	kk131066	"129764934923	LisaUshman	"1310947758	"b'@kk13	"b'	@	kk131066	kk131066 Thi	"1310947'	array([[0.0	0,057206	0.9358041	dtype=floa	array([1]))
"b'@kk131(	kk131066	"119976813283	kk131066	"1310930825	"b'@polic	"b'	@	polidan_sha	polidan_shar	"1310930:	array([[0.7	0,089164	0.113658	dtype=floa	array([-1]))
"b'TO ADD	kk131066	"992399004116	polidan_sharo	"1310927849	"b'@kk13	"b'	@	kk131066	kk131066 Aw	"1310927:	array([[0.0	0,098935	0.8592566	dtype=floa	array([1]))

Figure 50: Including in "Replies Table" Aspect_Based BERT Results

In order for the Nodes Table in **Figure 51** to be created, all user_screen_names from columns tweet_user_screen_name, reply_user_screen_name, and target_user_screen_name were collected, duplicate values were removed and an id number in hierarchical order was given for each user. The Edges Table in **Figure 52** derived from using the V_lookup function in excel and assigning the ids from Table Nodes to reply_user_screen_names and target_user_screen_names. As a result, a network of nodes and edges was displayed. For the rest attributes to be created, a second step of encoding, shown in **Figure 53** was followed.

Label	Id	Sentiment	Controvert Node
Bayernphan	1	-1	FALSE
kk131066	2	-1	yes
JanettThinks	3	-1	yes
JamieAllingham1	4	1	yes
thereseflanagan	5	-1	yes
Dolphfinn33	6	1	FALSE

Figure 51: Nodes Table in Gephi

Source	Target	Relationship	STATUS ALL	Controvert Edge
1	5630	1	2	FALSE
5630	1	1	1	FALSE
2	5631	1	2	yes
5631	2	1	1	yes
5632	2	1	1	yes
2	5633	1	2	yes
5633	2	1	1	yes
5634	2	1	2	yes
2	5634	1	2	yes
5634	2	1	1	yes

Figure 52 : Edges Table in Gephi

In order to assign negative or positive polarity to every single node in the network, two different outputs from sentiment analysis chapter were used. Both ASBA BERT results for Replies and Tweets tables were concentrated in **Figure 53** and a hypothesis was adopted for every single node in the network. If a user replied with a positive sentiment to a tweet supporting Trump, then specific user was also considered to be pro Trump candidacy node. Likewise, it was also hypothesized that if a user replied with a negative sentiment to a tweet supporting Trump, then specific user was captured with a negative polarity towards Trump candidacy. In this way, an attribute in Nodes Table called "Sentiment" was created for every single node of the network, by notifying the polarity of a user based on his/her tweet about "Trump" or his/her reply to other tweets naming Trump.

One of the key elements of this research, is defining and visualizing opinion controversy. This element of communication demonstrates opposition against another user's post. As previously explained, this thesis defines controversy as "the effect of a User replying to another one with negative polarity". This effect can be described with the ABSA BERT sentiment results for Replies table. Controversy affects the relationship between two nodes of the network for a singular connection and thus, it is considered to be an attribute describing a single edge within the network. The attribute "Relationship" in the Edges Table, captures the existence of controversy between two nodes with -1 and the alignment of opinions with +1 values (non-controvert relationships). Taking as an example in **Figure 50** the first row of results, it can be easily interpreted that the user with screen_name "@Bayernphan" has replied to the target user with screen_name "@denglercarlo" with positive sentiment, as per ABSA_BERT sentiment analysis applied for Table Replies. This means that the first user agrees with the other user's argument and thus this relationship is characterized as non-controvert. Consequently, all users that replied with positive sentiment to other users Tweets consider to be agreeing with them, while all users replying with negative sentiment to others are disagreeing and thus their relationship demonstrates "controversy".

Twitter User	Twitter_User_ID	Reply_User	Reply_User ID	Tweet Sentiment	Reply sentimnet	Combined Sentiment	STATUS ALL
Bayernphan	Bayernphan	Bayernphan	Bayernphan	-1	1	-1	2
Bayernphan	Bayernphan	denglercarlos	denglercarlos	-1	1	-1	1
kk131066	kk131066	kk131066	kk131066	-1	1	-1	2
kk131066	kk131066	OlderthanU70	OlderthanU70	-1	1	-1	1
kk131066	kk131066	LeftyLiteral	LeftyLiteral	-1	1	-1	1
kk131066	kk131066	kk131066	kk131066	-1	1	-1	2
kk131066	kk131066	BettyGooch2	BettyGooch2	-1	1	-1	1
kk131066	kk131066	LisaUshman	LisaUshman	-1	1	-1	2
kk131066	kk131066	kk131066	kk131066	-1	1	-1	2
kk131066	kk131066	LisaUshman	LisaUshman	-1	1	-1	1

Figure 53: Encoding Attribute "Sentiment" & "Relationship"

Another important phenomenon the visualizations in Gephi attempted to capture, was the diffusion. For the purposes of the current thesis, diffusion is defined as "the effect of serial replies and retweets of Users under a published Tweet". Identifying the sequence of communications and more specifically the succession in such an enormous network, it is considered to be one of the biggest challenges in this thesis. Even by using the time related metadata from Twitter API, the concept of succession would still not be perceived, since the real question is not "when a user replied to a specific tweet" it rather is "in which sequence a user replied to a specific tweet". The word sequence simply denotes before and after which user a specific reply is placed in the scale of succession.

"b'@kk	tweet_i	Twittec	Source	Target	STATUS	STATUS :	New Statu	New Statu	New Statu	New Statu	STATUS A
"b'RT @p	Bayernphi	Bayernphan	Bayernphan	denglercarlos		2	FALSE	#REF!	#REF!	#REF!	1
"b'@kk13	Bayernphi	Bayernphan	denglercarlc	Bayernphan	1	FALSE	2	FALSE	#REF!	#REF!	2
"b'RT @je	kk131066	kk131066	kk131066	OlderthanU70		2	FALSE	FALSE	FALSE	#REF!	1
"b'RT @je	kk131066	kk131066	OlderthanU7	kk131066	1	FALSE	2	FALSE	FALSE	FALSE	2
"b'@kk13	kk131066	kk131066	LeftyLiteral	kk131066	1	FALSE	FALSE	FALSE	FALSE	FALSE	1
"b'RT @Sl	kk131066	kk131066	kk131066	BettyGooch2		2	FALSE	FALSE	FALSE	FALSE	1
"b'@kk13	kk131066	kk131066	BettyGooch	kk131066	1	FALSE	2	FALSE	FALSE	FALSE	2
"b'RT @M	kk131066	kk131066	LisaUshmar	kk131066		2	FALSE	FALSE	FALSE	FALSE	1
"b'@kk13	kk131066	kk131066	kk131066	LisaUshman		2	2	FALSE	FALSE	FALSE	2
"b"IN COI	kk131066	kk131066	LisaUshmar	kk131066	1	FALSE	2	FALSE	FALSE	FALSE	2
"b'@kk13	kk131066	kk131066	kk131066	polidan_sharon		2	FALSE	FALSE	FALSE	FALSE	1
"b'TO ADI	kk131066	kk131066	polidan_sha	kk131066		2	2	FALSE	FALSE	FALSE	2
"b"TO AD	kk131066	kk131066	kk131066	polidan_sharon		2	2	FALSE	FALSE	FALSE	2
"b"TO AD	kk131066	kk131066	polidan_sha	kk131066	1	FALSE	2	FALSE	4	FALSE	4

Figure 54: Encoding Attribute "Status"

To demonstrate this successive relationship between different nodes, all replies collected under an initial tweet were considered to be a conversation. Within the same conversation, users can reply to the initial tweet posted or to other users already replied to the first. All replies under one conversation were categorized based on the succession in different statuses. From data collection chapter, it is already known that Replies extracted from Twitter API are by default in hierarchical order by time of posting. Since the succession of events already exists, what is critical to identify before categorizing each connection are the subconversations created within a main conversation, under a specific tweet. As shown in **Figure 54** it is assumed that if at the same time two statements take place, source_user in position n is equal to target_user in position n-4 and source_user in position n-4 is equal to target_user in position n, then specific row in the excel table is representing a connection in status 5. That would mean that referring source and target users have exchanged within the same conversation 5 different texts. Seemingly, all connections are categorized in 5 different statuses following the same process. As it can be seen in **Figure 54** "Status All" is an attribute describing edges.

CHAPTER 5: Graph Analysis

5.1 Network Visualization

As already mentioned in the Literature Review section, the sentiment analysis results are usually visualized using appropriate graph analysis tools. For the purposes of the current research, sentiment results were further processed, and several graphs were developed, using the Gephi open-source software. The different graphs developed, are presented, and further discussed in this section.

In **Figure 55**, a network consisting of 62,800 different nodes connected with 98,306 edges is depicted and a general overview of the flow of communication for the entire network is illustrated. The graph is colored based on the “Sentiment” of nodes against our topic. Nodes in blue represent 67.37% being against Trump candidacy, while nodes in red represent the 32.63% being in favor of Trump candidacy.

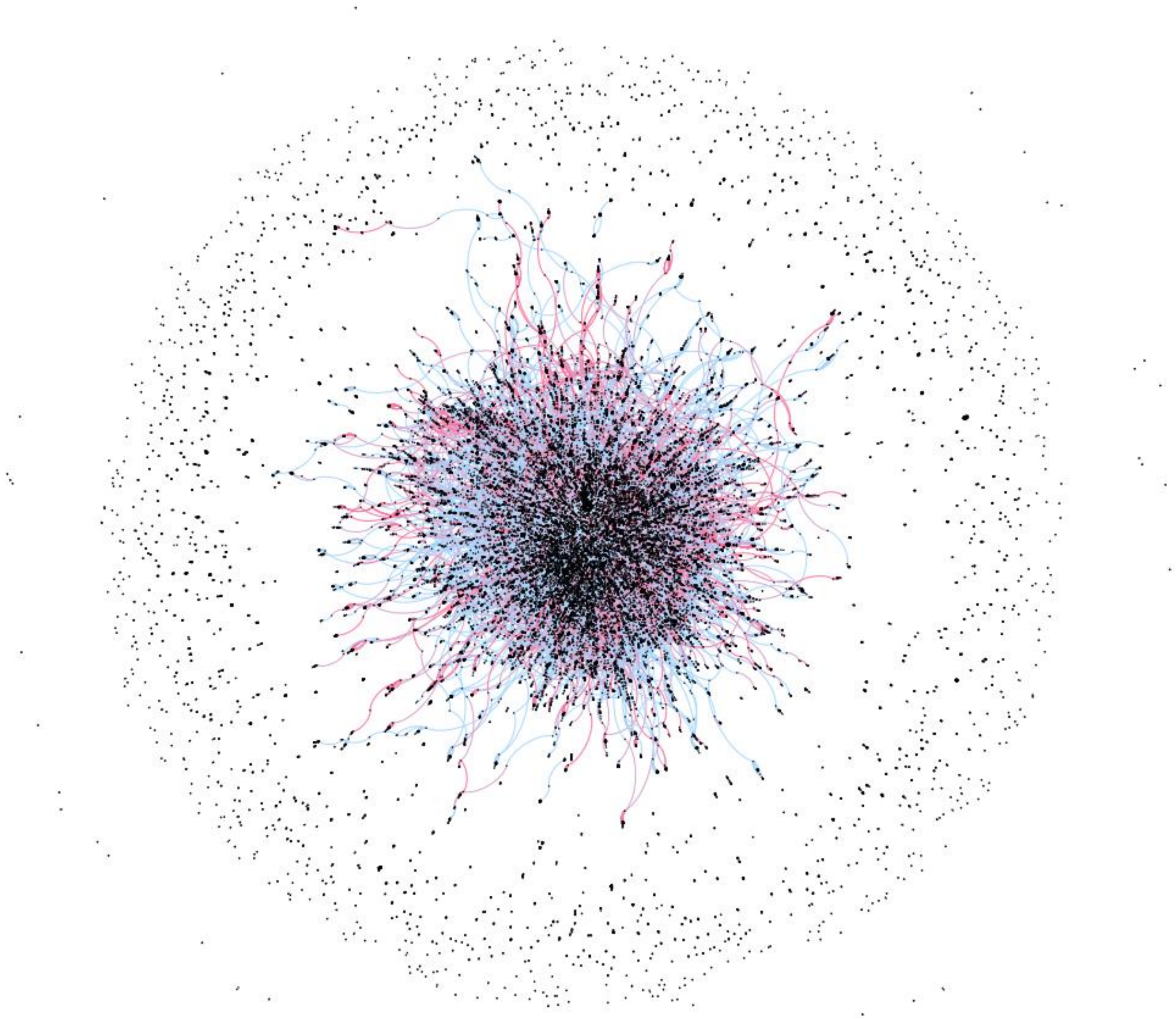

Figure 55: Graph Analysis Results - Network Visualization & Sentiment

5.2 Sub-communities in the Network

As the initial network is quite enormous for guiding into any results, the next step was filtering the network based on the modularity. Modularity measures the strength of division of a network in modules or else communities. High modularity in networks reveals dense connections between nodes within modules and sparse connections between nodes in different modules. As shown in **Figure 56**, the network has high modularity and consists of 3,442 unevenly sized subcommunities, each of which representing a different conversation. A further examination in **Figure 56** reveals that the majority of nodes within a subcommunity share the same polarity towards our topic, since the nodes participating within a conversation are seemingly colored. These findings are largely in line with those of previous studies mentioned in the literature review, showing that users strongly connected within a network are more likely to share the same knowledge, communicate more frequently, are influenced by the same forces, and share a similar code of ethics.

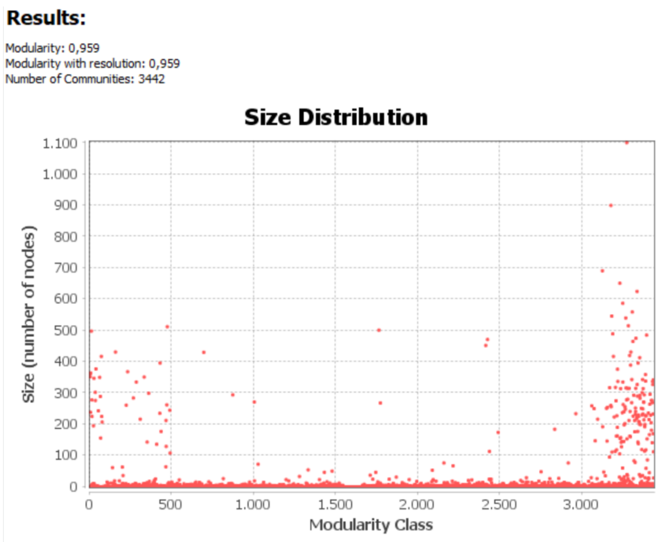

Figure 56: Gephi Output on Modularity Results

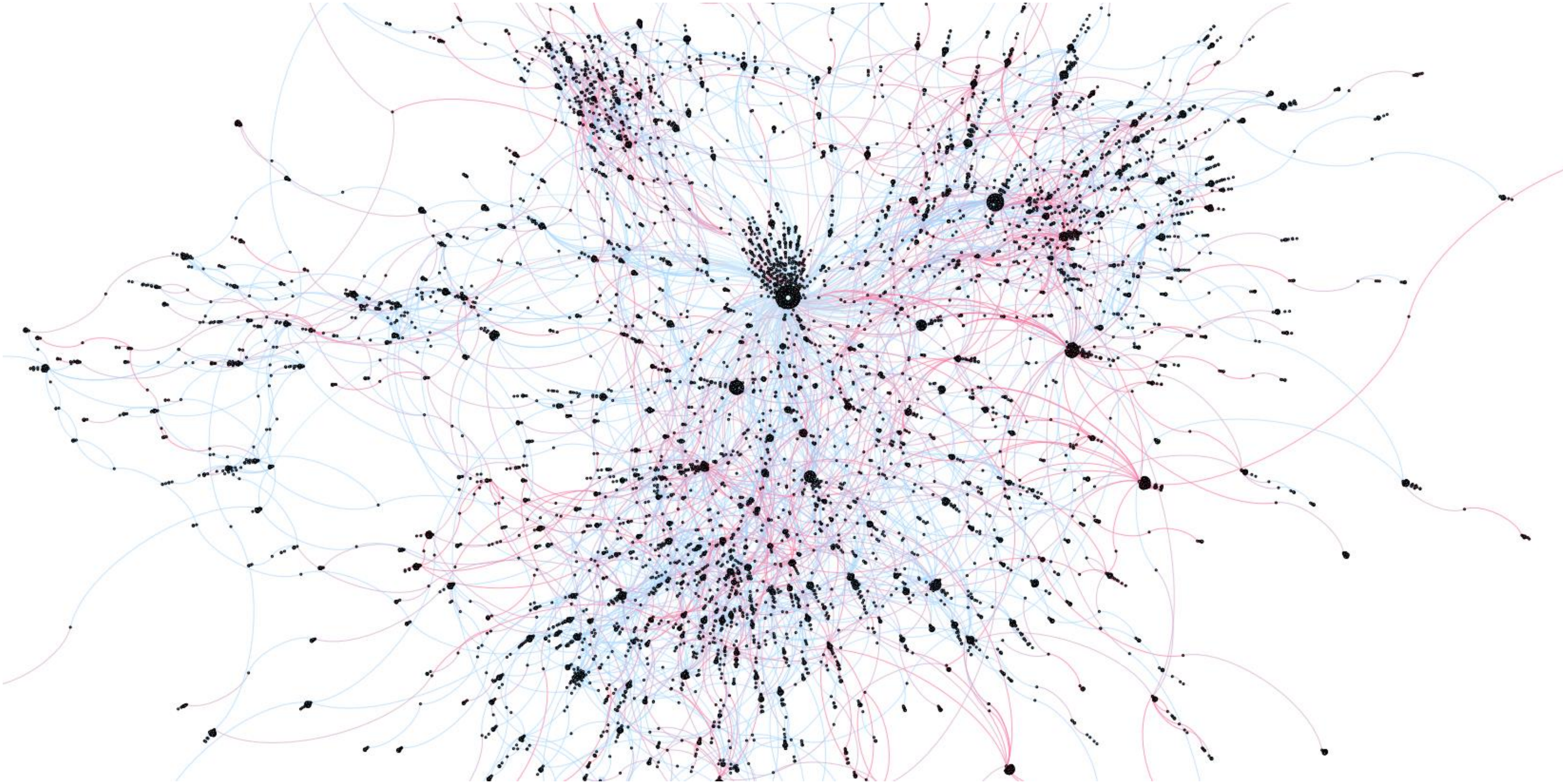

Figure 57: Graph Analysis Results - Subcommunities Visualization

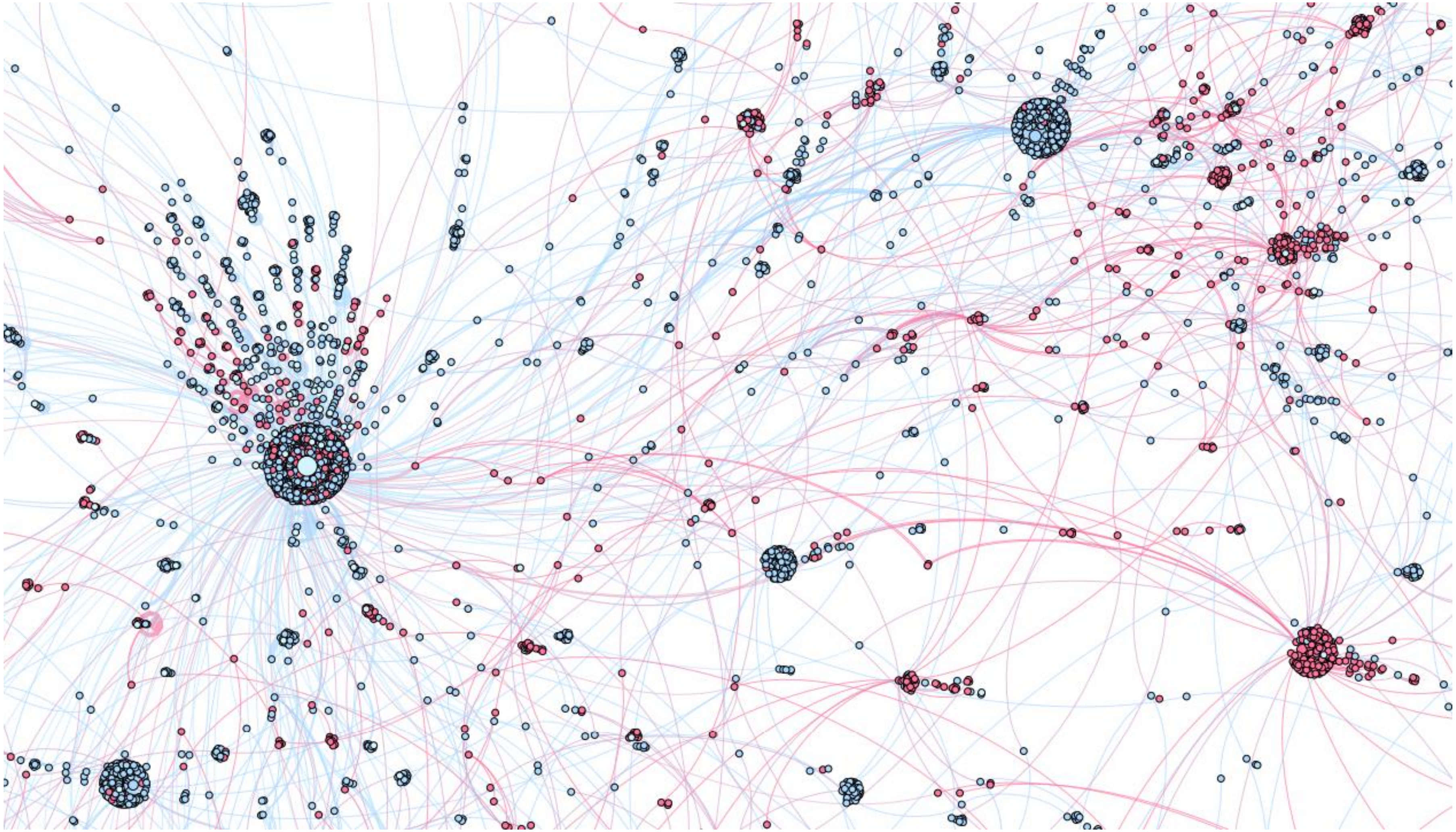

Figure 58: Graph Analysis Results - Closer Look in Subcommunities

5.3 Detecting Influencers

Detecting influencers in the network can be easily accomplished using another metric available in Gephi, called “In Degree”. Degree here refers to the number of replies that a user received from the rest of the network. Users with high connectivity through the network, who are followed by many users and have gained popularity, tend to receive more attention with several likes, retweets, and replies. The “In Degree” attribute has also been used in previous studies, including Manickavasagam & Vinayaga Sundaram, 2015, who attempted to detect gender-based influencers using Social Network Analysis, shows all nodes receiving more than 30 replies, with the bigger node in size receiving 620 replies. Moreover, this graph also reveals the connections between influencers among the network, although there is no specific pattern and thus, no further conclusions can be derived.

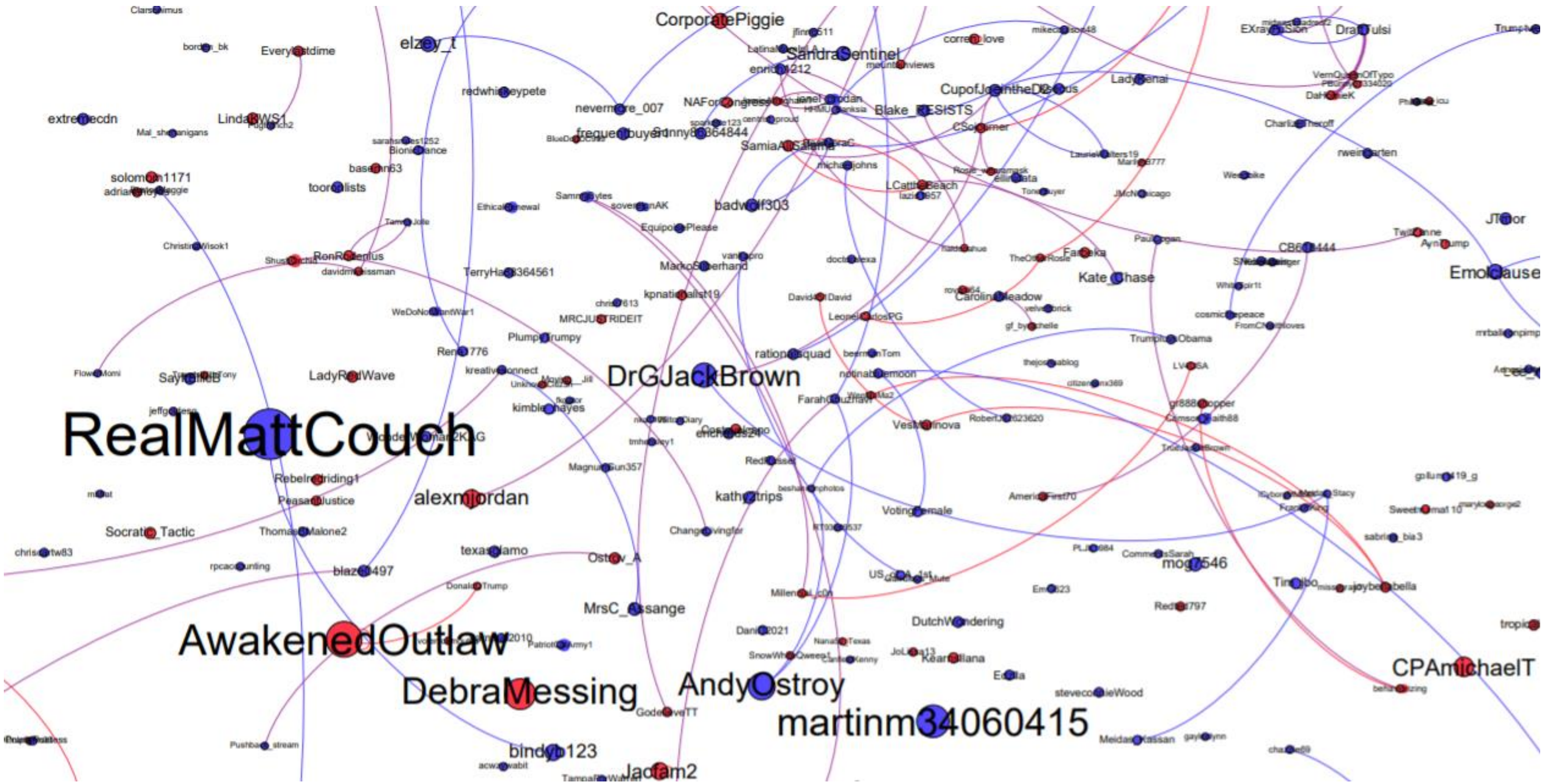

Figure 59: Graph Analysis Results - Influencers Visualization

5.4 Visualizing Controversy

So far, little effort has been done in terms of quantifying controversy as a phenomenon in social network analysis. In this thesis, controversy has been defined as the effect of a User replying to another one with negative polarity. This type of contravention between different voices in Twitter can form a mechanism for users to trigger dialogue and stimulate their curiosity for searching the "truth". Disagreeing in daily conversations is a characteristic for leveling up conversations and keeping the members active to participate and elaborate more on their ideas and believes. Challenge is one of the greatest feelings that can activate human behavior and enhance ingenuity. Higher activity and ingenuity can drive to conversations with substantial value for the network and its users and for that reason, measuring controversy can reveal new characteristics about a network and the relationships between users. To measure and visualize controversy, a filter was implemented on the attribute "Relationship" and controvert replies are colored in red. The total number of controvert replies in the graph are 5,019, almost the 7,36% of the total. That means only a small minority of all interactions in the network are opposing to the open dialogue spread about Trump candidacy. This is a striking result comparing to what is used in real life and everyday conversations. However, many explanations could be given, considering that most of users are interconnected and follow people in Twitter they already know and share similar ideas. As already described in the literature review section, dense communities with tight connections share many similar characteristics and the same code of values and believes. That could justify why the vast majority of user replies had positive polarity and agreeing with the initial tweets, since they probably reply to friends or people they already agree on different aspects as well. Another factor that could interpret the relative low percent of controversy, is the tendency of users to identify with other people they follow and try to show appreciation, sharing common ground with them and expressing acceptance to their idols. (Pinquart & Sörensen, 2000) proposed the Status Theory, where a relationship between two nodes within a network is highly related to the status specific users promote. For example, if user A shares some similarities with another user B, but the first one has more followers, receives more likes and in general holds a higher status in the network, user B will be competing in the network with other users for A's approval and acknowledgement. That makes users in second category to agree more easily with other user's ideas and promote same ideas for the sake of social acceptance.

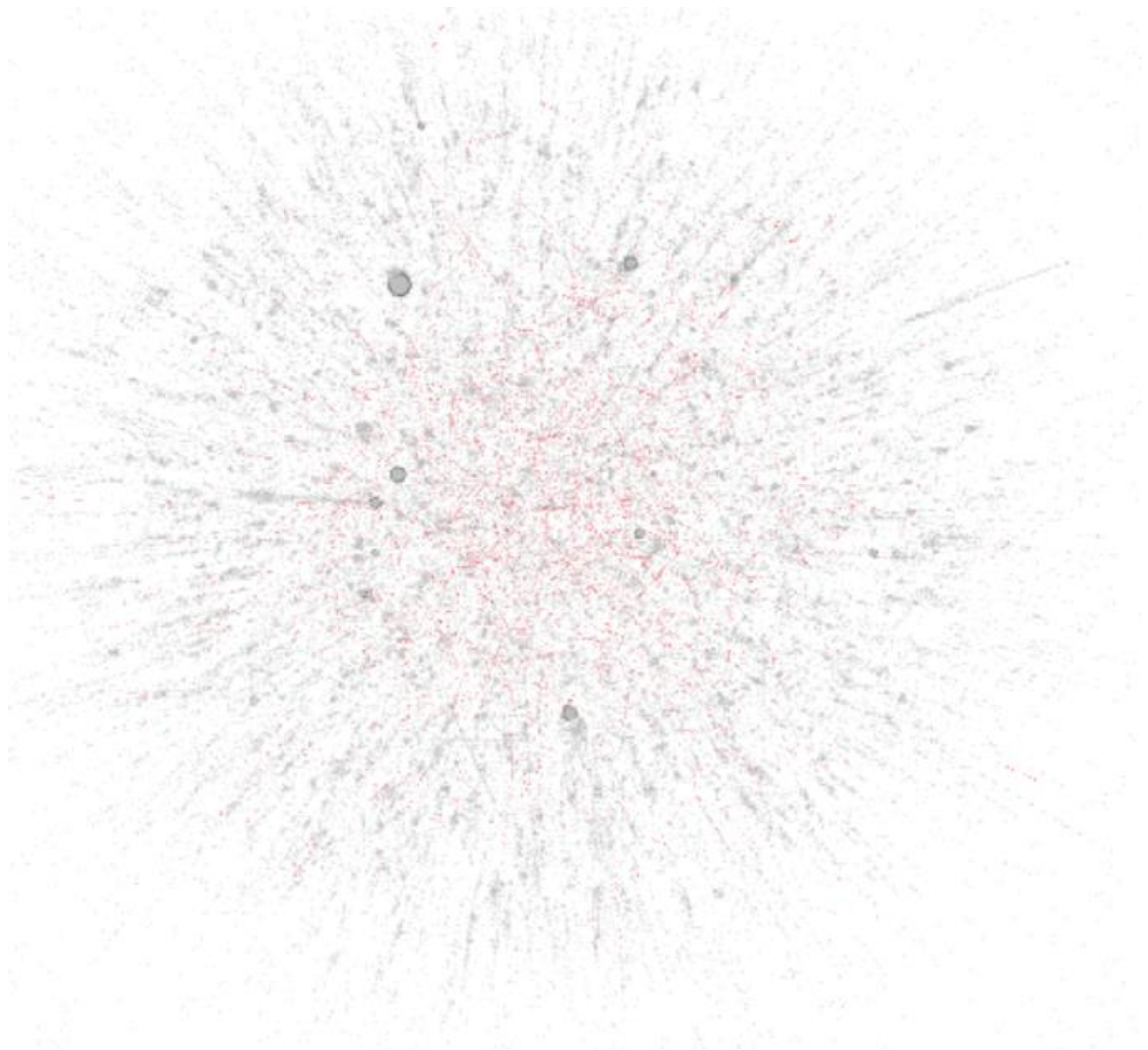

Figure 60: Graph Analysis Results - Network Visualization & Controversy Phenomenon

5.5 Controversy and different Polars

Digging deeper into analyzing controversy phenomenon and its aspects, the above graphs showcase for each polar all connections of vertices that replied with negative sentiment and represent controversy in the netwok. To this end, two different filters were applied for attributes “Sentiment” and “Relationship”, in order for all controvert nodes for each polar to be presented. The first graph consists of 42,029 vertices (66,03% of the total) and 40,611 nodes (58,74% of the total), while having 825 connections with negative polarity. **Figure 62** is respectively smaller than **Figure 61**, almost consisting of 32,63% nodes of the whole network and 25,91% edges and having even less connections with negative polarity, 353 in the number (0,51% of the total network). Even if th graphs do not consist the same number of nodes or connections and taking into consideration that contorversy is only covering a really small part of the netwrok itself, again the second graph representing nodes pro Trump candidacy contains three times less opposing connections than the first graph. Considering also that the low number of contorversy in second graph is not proportionate comparing to the size of the two graphs, it can be persumed that second graph is indeed less affected from contorversy. That result could be interpreted into characterizing those two groups of people and the way they preffer to communicate among their peers. It also reveals the tedancy for nodes in blue to challenge more their network and oppose their ideas against other users. Consequently, nodes in the second graph seem to have omophony, with a tedency to agree more with each other. Another interesting factor is the tedency in the network for nodes to desagree more with other nodes in different polar than with their own, **Figure 63** is showcasing this. This is very close to real life communication and how people interact with each other, when having opposing ideas about a subject.

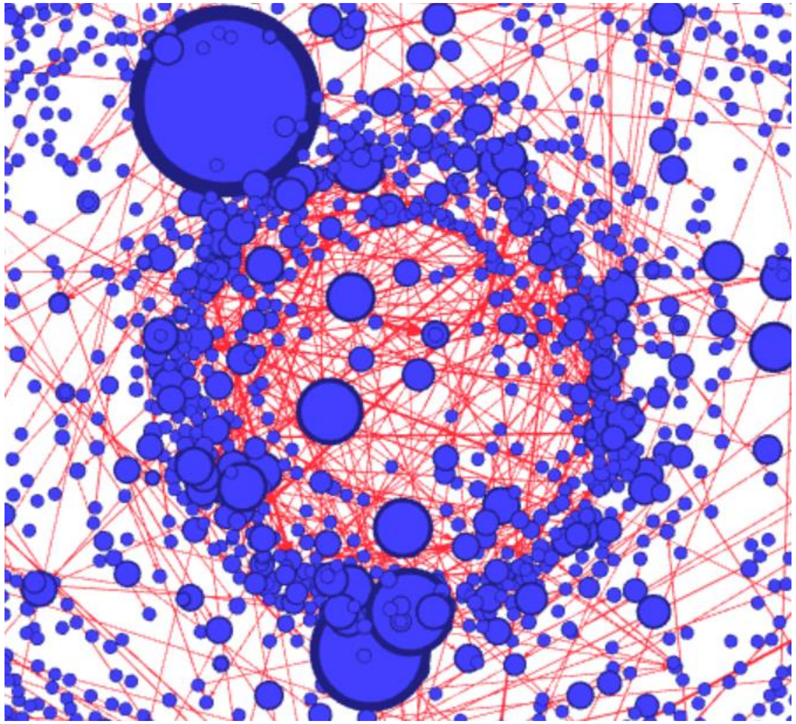

Figure 61: Graph Analysis Results - Blue Polar and Controversy Phenomenon

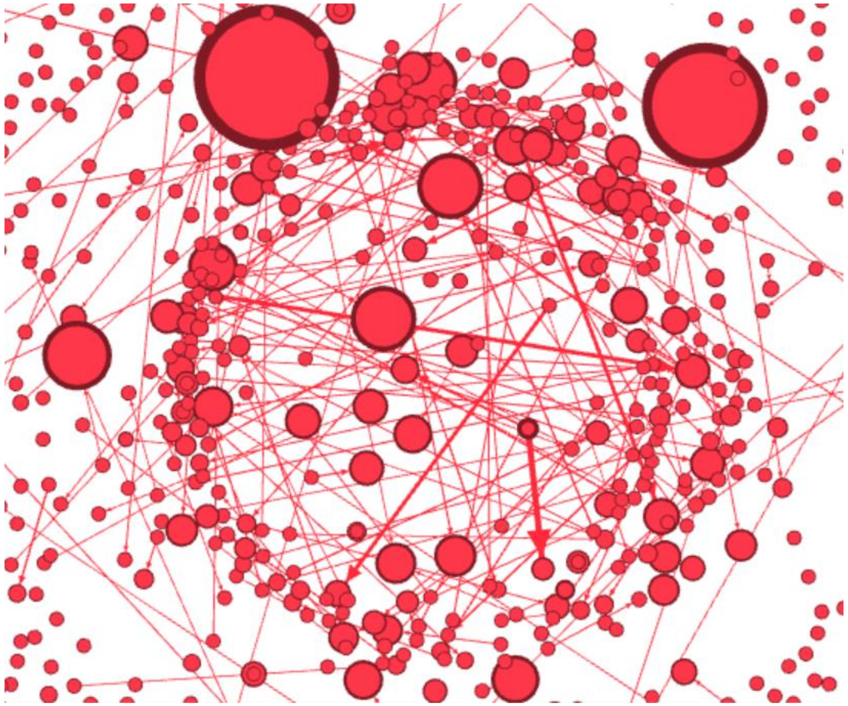

Figure 62: Graph Analysis Results - Red Polar and Controversy Phenomenon

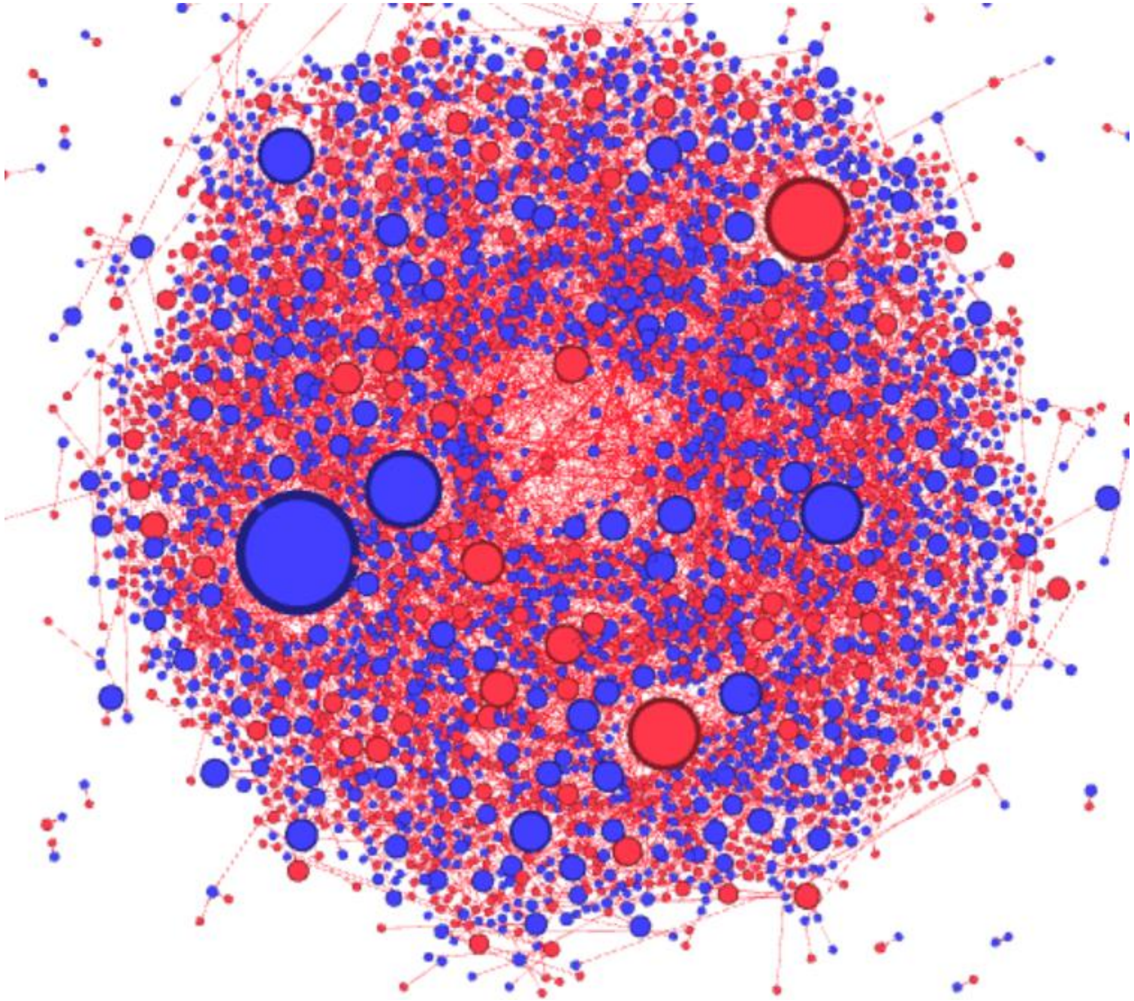

Figure 63: Graph Analysis Results - Controversy Phenomenon between the Polars of "Sentiment"

5.6 Controversy and Subcommunities

Figure 64, shows a more in-depth analysis on controversy between the more significant nodes within the system. For the graph development, two main filters were applied. First all nodes were filtered based on the eccentricity measure, capturing this way the distance between a node and the node which is furthest from it. The high eccentricity reveals that the furthest away node in the network is very distant, and a low eccentricity means that the furthest away node is quite close. A sub filter was also applied on betweenness centrality, in order to capture all subcommunities that are surrounding the main key players, who are interconnected and have been linking various parts of the network together. In **Figure 64** nodes with grey color represent users against Trump candidacy, while nodes in red showcase the ones in favor of, while edges in red are expressing controversy. Taking those parameters into consideration, it can be easily concluded that edges connecting homogenous subcommunities with the same sentiment towards Trump candidacy are grey in color, while edges stating controversy in red are connecting subcommunities with heterogeneous sentiment. This is a very useful insight, revealing that controversy is also present between diverse communities. In addition, it appears the key nodes expanding controversy in the main parts of the network are the influencers with high betweenness centrality and In Degree metrics. What would be even more challenging to assess, is understanding which communities have been created after controverting replies were posted in neighboring heterogenous communities. The sequence of those events could possibly prove that controversy is driving diffusion of information within the network.

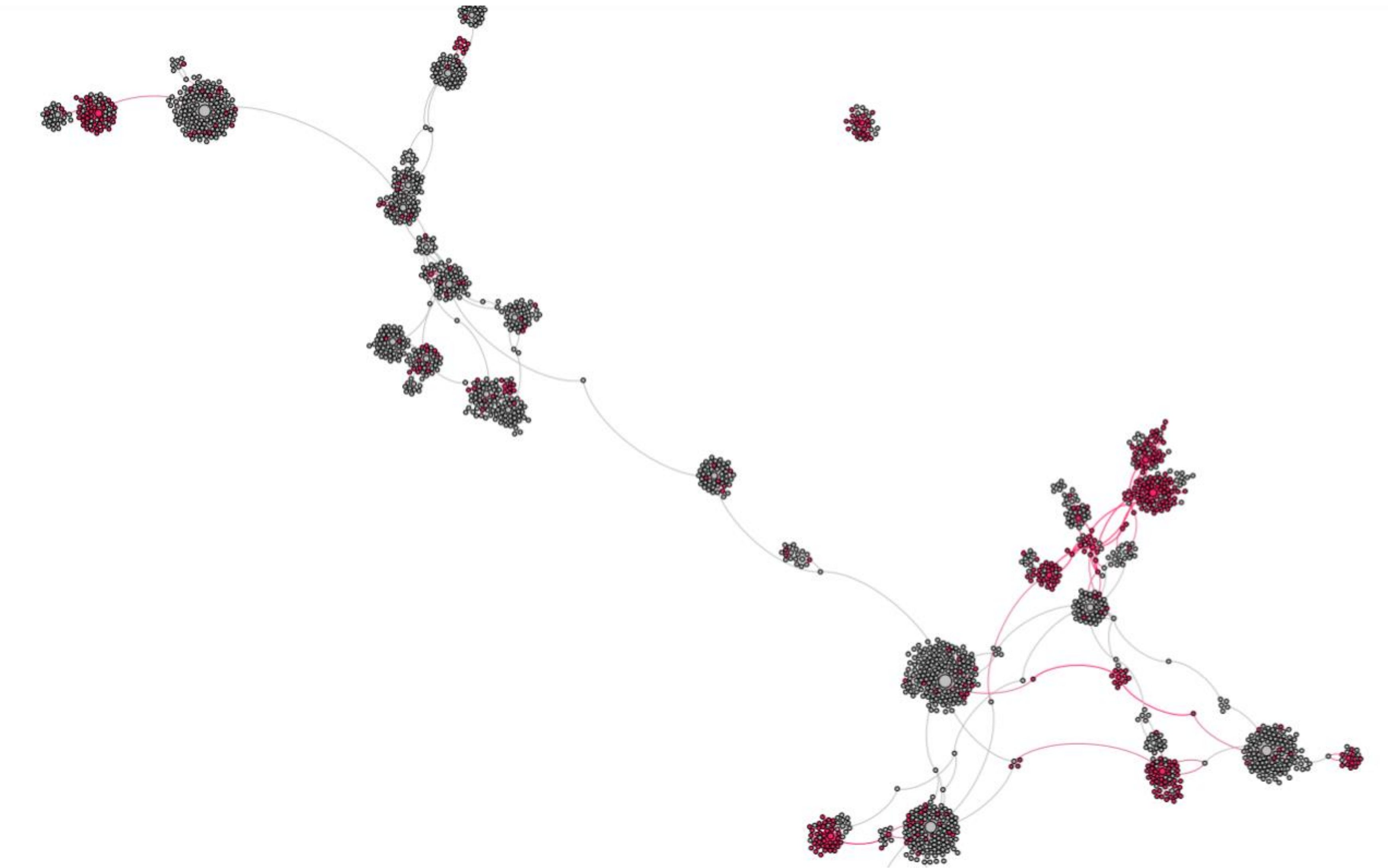

Figure 64: Graph Analysis Results - Controversy Phenomenon between Subcommunitie

5.7 Controversy & Diffusion

Another way of visualizing diffusion of the network in response to controversy spreading from one community to another, is filtering the graph based on the encoding already made for creating the attribute "Status". The showcases first the whole network and afterwards the network after applying the above-mentioned filter on the Status of every single edge in the system. Practically in second stage, all connections appeared in the network after a controvert reply were filtered out and the difference between the two stages shows the impact of the absence of 21.77% edges from the total connections. A striking realization is that almost 20% of the graph appears to exist after a controvert reply posted, triggering this way reaction. Those results can be seen also in **Table 5**. Consequently, it could be argued that controversy within the network has the ability as a phenomenon to initiate reaction from the vertices.

Table 5: Gephi Results after Status Filtering Applied

Nodes:	49720 (78,12% visible)
Edges:	54288 (78,23% visible)
Directed Graph	

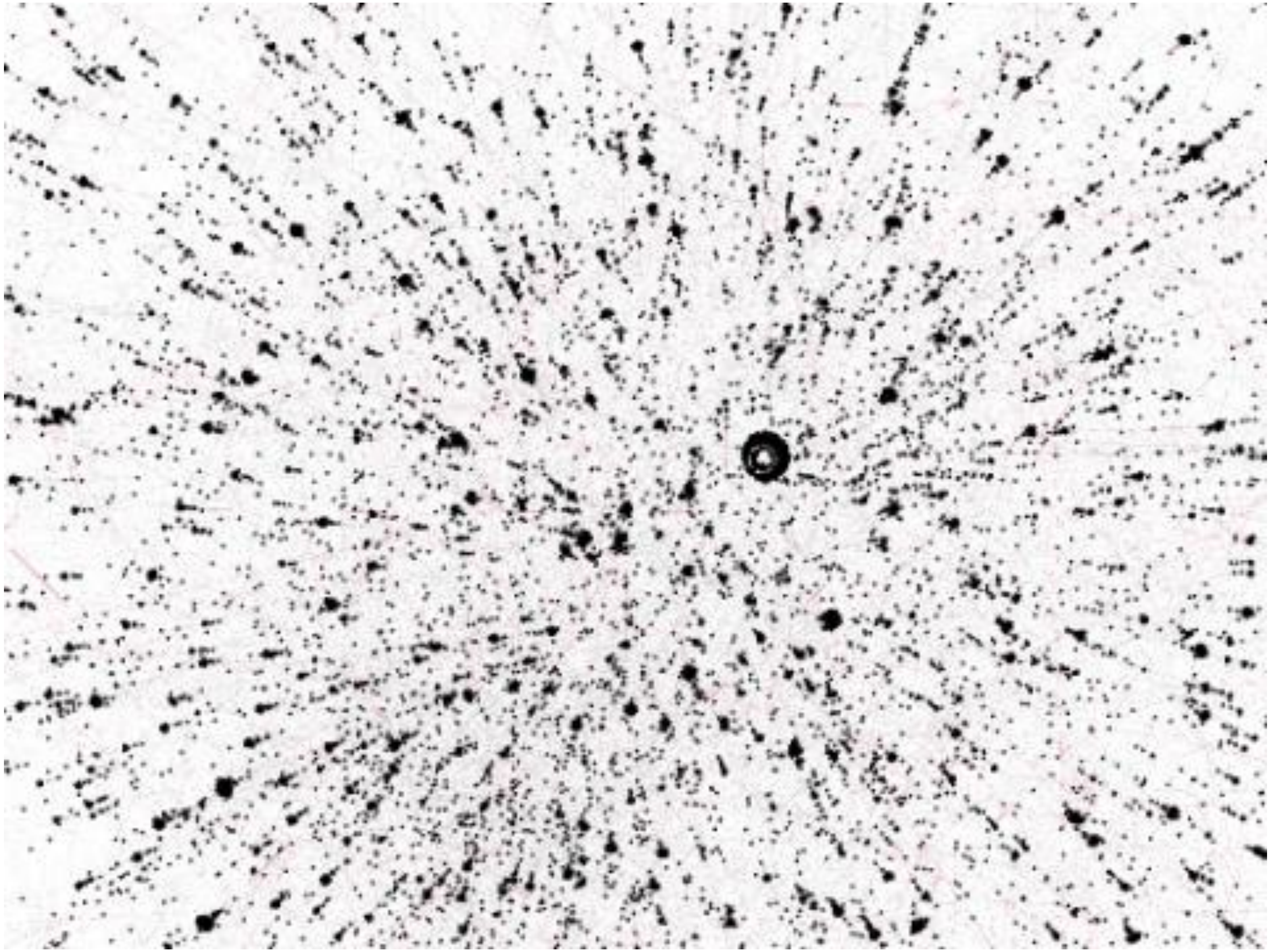

Figure 65: Graph Analysis Results - Controversy Phenomenon and Network Diffusion

CHAPTER 6: Conclusions

This thesis was concentrated into collecting Twitter data and developing a network of interactions characterized by sentiment towards a political event. Bipolarity and controversy phenomena were identified in the network and further analyzed along with their relations to diffusion of information.

A data collection framework was established in order to connect all users who replied, retweeted, liked or followed other users who made a post in relation to Trump candidacy. Using Twitter API gate, a number of metadata were also collected and used as different attributes characterizing network users or their relationships. The sentiment analysis process followed using BERT aspect-based model. This method was first used for classifying users in two polars based their sentiment against "Trump", and secondly for mapping the relationships among users as "controvert" or "affirming" based on their replies to other users. Final step of this thesis was graph analysis process using Gephi. Force Atlas was the main algorithm used for graph visualizations, while several graph attributes like In Degree, Modularity, or Eccentricity were used as filters. Following are the main results of this research and further topics of interest.

Aspect-based sentiment analysis prove to be the most accurate method for identifying sentiment towards a specific subject analyzing political text. By identifying the target subject within a post and analyzing text polarity towards this specific subject, it was managed to recognize user's opinion about Trump candidacy, as well as their opinion about other users' posts. To maximize method accuracy, it would be preferable to use as an input a training set containing political text, although the available datasets were only contemplating customer reviews.

The literature was supported by this thesis arguments concerning the homogeny of users, connected with strong tights, are reflecting in the network. According to graph visualizations the majority of subcommunities created in the network were polarized either in favor or against Trump candidacy, with a very small percentage of outliers defying the majority.

Another result reflected also on the literature review would be the prevalence of the most interconnected users in the network as the main influencers. When filtering the graph by Eccentricity attribute it was detected that the nodes with the bigger popularity, receiving the more tweets and replies, tend to be also the nodes connected with other subcommunities. From there it could be easily derived that the most popular users, receiving the biggest number of replies in the network, are very important for spreading information and providing different ideas from one community to another.

A finding also reported from graph analysis was the tendency of users polarized pro Trump candidacy to show homophony between their communications, while the other polar showed more controversy within users' opinions. This phenomenon could possibly characterize the first group more rigorous and catholic to their believes and how they communicate those, while users in the second group are more willing to challenge their neighbors and reinforce dialogue.

A striking result that came across also from this research was the realization that sub-communities representing different polars (pro and against Trump candidacy) are most probably to be connected with a negative sentiment. This is a very reasonable realization somebody would have thinking about real-life human relationships and communications. It also appears that social media communication enhances a similar behavior when it comes to opinion polarization between two groups.

The relation between controversy and diffusion of information is the strongest argument supported from current thesis. Almost 20% of the graph appears to exist after a controvert reply posted, triggering this way re-action. Controversy within the network has the ability as a phenomenon to initiate reaction from other vertices, enchasing this way network growth in number and value.

Further Research

By adding the variable of time, further research could be made towards understanding deeper the phenomenon of diffusion in relation to controversy. Could possibly entire communities be developed as a consequence of a controvert relation between two nodes? Considering the succession of relationships evolving within a system and the timeframe of subcommittees erupting, it would be interesting to investigate further if the first node sourcing others into a sub-community was interacting with controversy. That would mean that the phenomenon of controversy not only helps a system to diffuse more information, but also is developed to a key element for starting a group discussion. Meanwhile, future research is required into understanding the role of influencers in using controversy in their messages and how that affects their popularity among other users.

Moreover, for the purpose of predictive research, further efforts are needed applying machine learning techniques in the graphs constructed over this thesis. Applying a graph neural network model, by using sentiment results and other attributes like the proximity of users, new graphs can arise predicting new nodes, relationships, and their sentiment. Predicting the sentiment users might develop towards other users' comments, or even manipulating the message of key players for achieving maximum diffusion within a system, could be a useful mechanism for marketing and advertising sector.

References

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the Web. *Social Networks*, *25*(3), 211–230. https://doi.org/10.1016/S0378-8733(03)00009-1

Adler, P. S., & Kwon, S. W. (2002). Social capital: Prospects for a new concept. *Academy of Management Review*, *27*(1), 17–40. https://doi.org/10.5465/AMR.2002.5922314

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, *39*(1), 45–65. https://doi.org/10.1016/S0306-4573(02)00021-3

Arafat, N. A., & Bressan, S. (2017). Hypergraph drawing by force-directed placement. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10439 LNCS*(March), 387–394. https://doi.org/10.1007/978-3-319-64471-4_31

Best-hashtags.com. (2021). http://best-hashtags.com/hashtag/election/

Borgatti, S. P., & Halgin, D. S. (2011). On network theory. *Organization Science*, *22*(5), 1168–1181. https://doi.org/10.1287/orsc.1100.0641

Brass, D. J., Butterfield, K. D., & Skaggs, B. C. (1998). Relationships and Unethical Behavior: A Social Network Perspective. *The Academy of Management Review*, *23*(1), 14. https://doi.org/10.2307/259097

Bullard, S. B. (2014). Book Review: Twitter: Social Communication in the Twitter Age , by Dhiraj Murthy . *Journalism & Mass Communication Quarterly*, *91*(4), 861–862. https://doi.org/10.1177/1077699014554765p

Choi, S. S., Cha, S. H., & Tappert, C. C. (2009). A survey of binary similarity and distance measures. *WMSCI 2009 - The 13th World Multi-Conference on Systemics, Cybernetics and Informatics, Jointly with the 15th International Conference on Information Systems Analysis and Synthesis, ISAS 2009 - Proc.*, *3*, 80–85.

Cruz Mata, F., Troyano Jiménez, J., Enríquez de Salamanca Ros, F., & Ortega Rodríguez, F. (2008). Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento de Lenguaje Natural*, *41*(41), 73–80.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Di Caro, L., & Grella, M. (2013). Sentiment analysis via dependency parsing. *Computer Standards and Interfaces*, *35*(5), 442–453. https://doi.org/10.1016/j.csi.2012.10.005

Exploiting Fast LCF and BERT for Aspect-based Sentiment Analysis (Official Implementation of LCF-BERT). (2020). https://github.com/yangheng95/PyABSA

Eytan Bakshy, Jake M. Hofman, Winter A. Mason, Duncan J. Watts. (2011). *Everyone's an influencer: quantifying influence on twitter*.

Fahrni, A., & Klenner, M. (2008). *University of Zurich Old wine or warm beer : target-specific sentiment analysis of adjectives Old Wine or Warm Beer : Target-Specific Sentiment Analysis of Adjectives*. *April*, 60–63.

Fleck, R. J., & Calvert, A. T. (2015). Geochemistry, Geophysics, Geosystems. *Geochemistry Geophysics Geosystems*, *18*(1–2), 1541–1576. https://doi.org/10.1002/2015GC005807.Received

Ghosh, R., & Lerman, K. (2010). *Predicting Influential Users in Online Social Networks*. http://arxiv.org/abs/1005.4882

Granovetter, M. (1973). The Strength of Weak Ties Author (s): Mark S . Granovetter Published by : The University of Chicago Press Stable URL : http://www.jstor.com/stable/2776392 The Strength of Weak Ties '. *American Journal of Sociology*, *78*(6), 1360–1380.

Grosz, B. J. (1982). Natural language processing. *Artificial Intelligence*, *19*(2), 131–136. https://doi.org/10.1016/0004-3702(82)90032-7

Hatzivassiloglou, V., & McKeown, K. R. (1997). *Predicting the semantic orientation of adjectives*. 174–181. https://doi.org/10.3115/979617.979640

He, Y., & Zhou, D. (2011). Self-training from labeled features for sentiment analysis. *Information Processing and Management*, *47*(4), 606–616. https://doi.org/10.1016/j.ipm.2010.11.003

Hernández, C., Del Sagrado, J., Rodríguez, F., Moreno, J. C., & Sánchez, J. A. (2015). Modeling of energy demand of a high-tech greenhouse in warm climate based on Bayesian networks. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/201646

Hoang, M., Bihorac, O. A., & Rouces, J. (2019). Aspect-Based Sentiment Analysis using {BERT}. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 187–196. https://www.aclweb.org/anthology/W19-6120

Hu, H., Li, J., Wang, H., Daggard, G., & Shi, M. (2006). A maximally diversified multiple decision tree algorithm for microarray data classification. *Workshop on Intelligent Systems for Bioinformatics*, *73*, 35–38. http://portal.acm.org/citation.cfm?id=1274178

Huaxia Rui, Yizao Liu,Andrew B. Whinston. (2010). *CHATTER MATTERS: HOW TWITTER CAN OPEN THE BLACK BOX OF ONLINE WORD-OF-MOUTH*.

Jacomy, M., Venturini, T., Heymann, S., & Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS ONE*, *9*(6), 1–12. https://doi.org/10.1371/journal.pone.0098679

Jagtap, V. S., & Pawar, K. (2013). Analysis of different approaches to Sentence-Level Sentiment Classification. *International Journal of Scientific Engineering and Technology*, *2*(3), 164–170. http://ijset.com/ijset/publication/v2s3/paper11.pdf

Jurkowitz, B. Y. M., Hitlin, P., Mitchell, A., Santhanam, L., Adams, S., & Vogt, N. (2013). *Mark Jurkowitz, Paul Hitlin, Amy Mitchell, Laura Santhanam, Steve Adams, Monica Anderson and Nancy Vogt*.

Kaufmann, M. (2012). JMaxAlign: {A} Maximum Entropy Parallel Sentence Alignment Tool. *{COLING} 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Demonstration Papers, 8-15 December 2012, Mumbai, India*, *December 2012*, 277–288. http://aclweb.org/anthology/C/C12/C12-3035.pdf

Keith A. Quesenberry. (2021). *Are You Simply Treating Symptoms? For better digital marketing results ensure that you are addressing the root cause*. https://www.postcontrolmarketing.com/

Ko, Y., & Seo, J. (2000). *Automatic text categorization by unsupervised learning*. 453–459. https://doi.org/10.3115/990820.990886

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). (2010). What is Twitter, a social network or a news media? Proceedings of the 19th International Conference on World Wide Web - WWW ’10. doi:10.1145/1772690.1772751. *Archivos de Zootecnia*, *60*(230), 297–300. https://doi.org/10.4321/S0004-05922011000200015

Lane, P. C. R., Clarke, D., & Hender, P. (2012). On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data. *Decision Support Systems*, *53*(4), 712–718. https://doi.org/10.1016/j.dss.2012.05.028

Leena A Dhande, G. K. P. (2014). *Analyzing sentiment of movie review data using Naive Bayes neural classifier*.

Manickavasagam, S., & Vinayaga Sundaram, B. (2015). Exploring gender based influencers using Social Network Analysis. *6th International Conference on Advanced Computing, ICoAC 2014*, 224–228. https://doi.org/10.1109/ICoAC.2014.7229715

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Min, H. J., & Park, J. C. (2009). Toward finer-grained sentiment identification in product reviews through linguistic and ontological analyses. *ACL-IJCNLP 2009 - Joint Conf. of the 47th Annual Meeting of the Association for Computational Linguistics and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Proceedings of the Conf.*, *August*, 169–172. https://doi.org/10.3115/1667583.1667635

Mischaud, E. (2007). Twitter: Expressions of the Whole Self - An investigation into user appropriation of a web-based communications platform. *London Media Lse Retrieved May*, 53. http://www.lse.ac.uk/collections/media@lse/mediaWorkingPapers/MScDissertationSeries/Mishaud_Final.pdf

Mohammadpour, A., Arbatani, T. R., Gholipour, T. H., Farzianpour, F., & Hosseini, S. (2014). A Survey of the Effect of Social Media Marketing on Online Shopping of Customers by Mediating Variables. *Journal of Service Science and Management*, *07*(05), 368–376. https://doi.org/10.4236/jssm.2014.75034

Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. https://doi.org/10.1016/j.eswa.2012.07.059

Moreno, J. L. (1934). *Who shall survive?: A new approach to the problem of human interrelations*.

Navlani, A. (2018). *Datacamp*. https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

Newman, M. (2010). *Networks*.

Noack, A. (2003). *An Energy Model for Visual Graph Clustering*.

Palmer, A., & Koenig-Lewis, N. (2009). An experiential, social network-based approach to direct marketing. *Direct Marketing: An International Journal*, *3*(3), 162–176. https://doi.org/10.1108/17505930910985116

Pinquart, M., & Sörensen, S. (2000). Influences of socioeconomic status, social network, and competence on subjective well-being in later life: A meta-analysis. *Psychology and Aging*, *15*(2), 187–224. https://doi.org/10.1037/0882-7974.15.2.187

Qiu, X. P., Sun, T. X., Xu, Y. G., Shao, Y. F., Dai, N., & Huang, X. J. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*(10), 1872–

1897. https://doi.org/10.1007/s11431-020-1647-3

Quinlan., J. R. (1964). Programs for machine learning. Part II. *Information and Control*, *7*(1), 55–77. https://doi.org/10.1016/S0019-9958(64)90259-1

Rost, K. (2011). The strength of strong ties in the creation of innovation. *Research Policy*, *40*(4), 588–604. https://doi.org/10.1016/j.respol.2010.12.001

Science Direct. (2021). https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019

Sharma, R., Nigam, S., & Jain, R. (2014). Opinion Mining of Movie Reviews At Document Level. *International Journal on Information Theory*, *3*(3), 13–21. https://doi.org/10.5121/ijit.2014.3302

Singh, V., & Dubey, S. K. (2014). Opinion mining and analysis: A literature review. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, 232–239. https://doi.org/10.1109/CONFLUENCE.2014.6949318

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2019). *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. 1–16. http://arxiv.org/abs/1908.08530

Taboada, M., Brooke, J., & Voll, K. (2011). *Lexicon-Based Methods for Sentiment Analysis*. *August 2010*.

Tan, A.-H. (1999). Text Mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, *8*, 65–70. https://doi.org/10.1.1.38.7672

Tribhuvan, P. P., Bhirud, S. G., & Tribhuvan, A. P. (2014). A Peer Review of Feature Based Opinion Mining and Summarization. *International Journal of Computer Science and Information Technologies (IJCSIT)*, *5*(1), 247–250.

Twitter Web Developer Platform. (2020). https://developer.twitter.com/en/docs/twitter-api/enterprise/account-activity-api/overview

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Villaseñor Rodríguez, I. (2014). Publicaciones españolas para la fundamentación de un marco teórico sobre los estudios de usuarios de información. *Investigación Bibliotecológica: Archivonomía, Bibliotecología e Información*, *28*(63), 223–257. https://doi.org/10.1016/s0187-358x(14)72581-x

Walker, M. A., Mehl, M. R., Moore, R. K., & Mairesse, F. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of Artificial Intelligence Research*, *30*(Xxxx), 457–500. https://www.jair.org/index.php/jair/article/view/10520%0Ahttps://pdfs.semanticscholar.org/7a5e/a7c513aac77880d37bd98527f4d24dda44f8.pdf

Wang, J. B. and J. (2016). *Improving malware detection using multi-viewensemble learning*. https://onlinelibrary.wiley.com/doi/epdf/10.1002/sec.1600

Wang, P., Jia, Z., & Zhao, H. (2014). Grammatical error detection and correction using a single maximum entropy model. *CoNLL 2014 - 18th Conference on Computational Natural Language Learning, Proceedings of the Shared Task*, *13511500200*, 74–82. https://doi.org/10.3115/v1/w14-1710

Weng, J., & Lim, E.-P. (2010). *NoTwitterRank: finding topic-sensitive influential twitterers*.

Xianghua, F., Guo, L., Yanyan, G., & Zhiqiang, W. (2013). Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, *37*, 186–195. https://doi.org/10.1016/j.knosys.2012.08.003

Xu, K., Liao, S. S., Li, J., & Song, Y. (2011). Mining comparative opinions from customer reviews for Competitive Intelligence. *Decision Support Systems*, *50*(4), 743–754. https://doi.org/10.1016/j.dss.2010.08.021

Yang, J., & Counts, S. (2010). Predicting the speed, scale, and range of information diffusion in Twitter. *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 355–358.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhang, L., & Corporation, L. (2011). RAAF Richmond. *A + U-Architecture and Urbanism*, *487*, 121.

Zhou, F., Jiao, J. R., Yang, X. J., & Lei, B. (2017). Augmenting feature model through customer preference mining by hybrid sentiment analysis. *Expert Systems with Applications*, *89*, 306–317. https://doi.org/10.1016/j.eswa.2017.07.021

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the Web. *Social Networks*, *25*(3), 211–230. https://doi.org/10.1016/S0378-8733(03)00009-1

Adler, P. S., & Kwon, S. W. (2002). Social capital: Prospects for a new concept. *Academy of Management Review*, *27*(1), 17–40. https://doi.org/10.5465/AMR.2002.5922314

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, *39*(1), 45–65. https://doi.org/10.1016/S0306-4573(02)00021-3

Arafat, N. A., & Bressan, S. (2017). Hypergraph drawing by force-directed placement. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10439 LNCS*(March), 387–394. https://doi.org/10.1007/978-3-319-64471-4_31

Best-hashtags.com. (2021). http://best-hashtags.com/hashtag/election/

Borgatti, S. P., & Halgin, D. S. (2011). On network theory. *Organization Science*, *22*(5), 1168–1181. https://doi.org/10.1287/orsc.1100.0641

Brass, D. J., Butterfield, K. D., & Skaggs, B. C. (1998). Relationships and Unethical Behavior: A Social Network Perspective. *The Academy of Management Review*, *23*(1), 14. https://doi.org/10.2307/259097

Bullard, S. B. (2014). Book Review: Twitter: Social Communication in the Twitter Age , by Dhiraj Murthy . *Journalism & Mass Communication Quarterly*, *91*(4), 861–862. https://doi.org/10.1177/1077699014554765p

Choi, S. S., Cha, S. H., & Tappert, C. C. (2009). A survey of binary similarity and distance measures. *WMSCI 2009 - The 13th World Multi-Conference on Systemics, Cybernetics and Informatics,* *Jointly with the 15th International Conference on Information Systems Analysis and Synthesis, ISAS 2009 - Proc.*, *3*, 80–85.

Cruz Mata, F., Troyano Jiménez, J., Enríquez de Salamanca Ros, F., & Ortega Rodríguez, F. (2008). Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento de Lenguaje Natural*, *41*(41), 73–80.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Di Caro, L., & Grella, M. (2013). Sentiment analysis via dependency parsing. *Computer Standards and Interfaces*, *35*(5), 442–453. https://doi.org/10.1016/j.csi.2012.10.005

Exploiting Fast LCF and BERT for Aspect-based Sentiment Analysis (Official Implementation of LCF-BERT). (2020). https://github.com/yangheng95/PyABSA

Eytan Bakshy, Jake M. Hofman, Winter A. Mason, Duncan J. Watts. (2011). *Everyone's an influencer: quantifying influence on twitter*.

Fahrni, A., & Klenner, M. (2008). *University of Zurich Old wine or warm beer : target-specific sentiment analysis of adjectives Old Wine or Warm Beer : Target-Specific Sentiment Analysis of Adjectives*. *April*, 60–63.

Fleck, R. J., & Calvert, A. T. (2015). Geochemistry, Geophysics, Geosystems. *Geochemistry Geophysics Geosystems*, *18*(1–2), 1541–1576. https://doi.org/10.1002/2015GC005807.Received

Ghosh, R., & Lerman, K. (2010). *Predicting Influential Users in Online Social Networks*. http://arxiv.org/abs/1005.4882

Granovetter, M. (1973). The Strength of Weak Ties Author (s): Mark S . Granovetter Published by : The University of Chicago Press Stable URL : http://www.jstor.com/stable/2776392 The Strength of Weak Ties '. *American Journal of Sociology*, *78*(6), 1360–1380.

Grosz, B. J. (1982). Natural language processing. *Artificial Intelligence*, *19*(2), 131–136. https://doi.org/10.1016/0004-3702(82)90032-7

Hatzivassiloglou, V., & McKeown, K. R. (1997). *Predicting the semantic orientation of adjectives*. 174–181. https://doi.org/10.3115/979617.979640

He, Y., & Zhou, D. (2011). Self-training from labeled features for sentiment analysis. *Information Processing and Management*, *47*(4), 606–616. https://doi.org/10.1016/j.ipm.2010.11.003

Hernández, C., Del Sagrado, J., Rodríguez, F., Moreno, J. C., & Sánchez, J. A. (2015). Modeling of energy demand of a high-tech greenhouse in warm climate based on Bayesian networks. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/201646

Hoang, M., Bihorac, O. A., & Rouces, J. (2019). Aspect-Based Sentiment Analysis using {BERT}. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 187–196. https://www.aclweb.org/anthology/W19-6120

Hu, H., Li, J., Wang, H., Daggard, G., & Shi, M. (2006). A maximally diversified multiple decision tree algorithm for microarray data classification. *Workshop on Intelligent Systems for Bioinformatics*, *73*, 35–38. http://portal.acm.org/citation.cfm?id=1274178

Huaxia Rui, Yizao Liu,Andrew B. Whinston. (2010). *CHATTER MATTERS: HOW TWITTER CAN OPEN THE BLACK BOX OF ONLINE WORD-OF-MOUTH*.

Jacomy, M., Venturini, T., Heymann, S., & Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS ONE*, *9*(6), 1–12. https://doi.org/10.1371/journal.pone.0098679

Jagtap, V. S., & Pawar, K. (2013). Analysis of different approaches to Sentence-Level Sentiment Classification. *International Journal of Scientific Engineering and Technology*, *2*(3), 164–170. http://ijset.com/ijset/publication/v2s3/paper11.pdf

Jurkowitz, B. Y. M., Hitlin, P., Mitchell, A., Santhanam, L., Adams, S., & Vogt, N. (2013). *Mark Jurkowitz, Paul Hitlin, Amy Mitchell, Laura Santhanam, Steve Adams, Monica Anderson and Nancy Vogt*.

Kaufmann, M. (2012). JMaxAlign: {A} Maximum Entropy Parallel Sentence Alignment Tool. *{COLING} 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Demonstration Papers, 8-15 December 2012, Mumbai, India*, *December 2012*, 277–288. http://aclweb.org/anthology/C/C12/C12-3035.pdf

Keith A. Quesenberry. (2021). *Are You Simply Treating Symptoms? For better digital marketing results ensure that you are addressing the root cause*. https://www.postcontrolmarketing.com/

Ko, Y., & Seo, J. (2000). *Automatic text categorization by unsupervised learning*. 453–459. https://doi.org/10.3115/990820.990886

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). (2010). What is Twitter, a social network or a news media? Proceedings of the 19th International Conference on World Wide Web - WWW '10. doi:10.1145/1772690.1772751. *Archivos de Zootecnia*, *60*(230), 297–300. https://doi.org/10.4321/S0004-05922011000200015

Lane, P. C. R., Clarke, D., & Hender, P. (2012). On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data. *Decision Support Systems*, *53*(4), 712–718. https://doi.org/10.1016/j.dss.2012.05.028

Leena A Dhande, G. K. P. (2014). *Analyzing sentiment of movie review data using Naive Bayes neural classifier*.

Manickavasagam, S., & Vinayaga Sundaram, B. (2015). Exploring gender based influencers using Social Network Analysis. *6th International Conference on Advanced Computing, ICoAC 2014*, 224–228. https://doi.org/10.1109/ICoAC.2014.7229715

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Min, H. J., & Park, J. C. (2009). Toward finer-grained sentiment identification in product reviews through linguistic and ontological analyses. *ACL-IJCNLP 2009 - Joint Conf. of the 47th Annual Meeting of the Association for Computational Linguistics and 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Proceedings of the Conf.*, *August*, 169–172. https://doi.org/10.3115/1667583.1667635

Mischaud, E. (2007). Twitter: Expressions of the Whole Self - An investigation into user appropriation of a web-based communications platform. *London Media Lse Retrieved May*, 53. http://www.lse.ac.uk/collections/media@lse/mediaWorkingPapers/MScDissertationSeries/Mishaud_Final.pdf

Mohammadpour, A., Arbatani, T. R., Gholipour, T. H., Farzianpour, F., & Hosseini, S. (2014). A Survey of the Effect of Social Media Marketing on Online Shopping of Customers by Mediating Variables. *Journal of Service Science and Management*, *07*(05), 368–376.

https://doi.org/10.4236/jssm.2014.75034

Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. https://doi.org/10.1016/j.eswa.2012.07.059

Moreno, J. L. (1934). *Who shall survive?: A new approach to the problem of human interrelations*.

Navlani, A. (2018). *Datacamp*. https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

Newman, M. (2010). *Networks*.

Noack, A. (2003). *An Energy Model for Visual Graph Clustering*.

Palmer, A., & Koenig-Lewis, N. (2009). An experiential, social network-based approach to direct marketing. *Direct Marketing: An International Journal*, *3*(3), 162–176. https://doi.org/10.1108/17505930910985116

Pinquart, M., & Sörensen, S. (2000). Influences of socioeconomic status, social network, and competence on subjective well-being in later life: A meta-analysis. *Psychology and Aging*, *15*(2), 187–224. https://doi.org/10.1037/0882-7974.15.2.187

Qiu, X. P., Sun, T. X., Xu, Y. G., Shao, Y. F., Dai, N., & Huang, X. J. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*(10), 1872–1897. https://doi.org/10.1007/s11431-020-1647-3

Quinlan., J. R. (1964). Programs for machine learning. Part II. *Information and Control*, *7*(1), 55–77. https://doi.org/10.1016/S0019-9958(64)90259-1

Rost, K. (2011). The strength of strong ties in the creation of innovation. *Research Policy*, *40*(4), 588–604. https://doi.org/10.1016/j.respol.2010.12.001

Science Direct. (2021). https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019

Sharma, R., Nigam, S., & Jain, R. (2014). Opinion Mining of Movie Reviews At Document Level. *International Journal on Information Theory*, *3*(3), 13–21. https://doi.org/10.5121/ijit.2014.3302

Singh, V., & Dubey, S. K. (2014). Opinion mining and analysis: A literature review. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, 232–239. https://doi.org/10.1109/CONFLUENCE.2014.6949318

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2019). *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. 1–16. http://arxiv.org/abs/1908.08530

Taboada, M., Brooke, J., & Voll, K. (2011). *Lexicon-Based Methods for Sentiment Analysis*. *August 2010*.

Tan, A.-H. (1999). Text Mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, *8*, 65–70. https://doi.org/10.1.1.38.7672

Tribhuvan, P. P., Bhirud, S. G., & Tribhuvan, A. P. (2014). A Peer Review of Feature Based Opinion

Mining and Summarization. *International Journal of Computer Science and Information Technologies (IJCSIT)*, *5*(1), 247–250.

Twitter Web Developer Platform. (2020). https://developer.twitter.com/en/docs/twitter-api/enterprise/account-activity-api/overview

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Villaseñor Rodríguez, I. (2014). Publicaciones españolas para la fundamentación de un marco teórico sobre los estudios de usuarios de información. *Investigación Bibliotecológica: Archivonomía, Bibliotecología e Información*, *28*(63), 223–257. https://doi.org/10.1016/s0187-358x(14)72581-x

Walker, M. A., Mehl, M. R., Moore, R. K., & Mairesse, F. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of Artificial Intelligence Research*, *30*(Xxxx), 457–500. https://www.jair.org/index.php/jair/article/view/10520%0Ahttps://pdfs.semanticscholar.org/7a5e/a7c513aac77880d37bd98527f4d24dda44f8.pdf

Wang, J. B. and J. (2016). *Improving malware detection using multi-viewensemble learning*. https://onlinelibrary.wiley.com/doi/epdf/10.1002/sec.1600

Wang, P., Jia, Z., & Zhao, H. (2014). Grammatical error detection and correction using a single maximum entropy model. *CoNLL 2014 - 18th Conference on Computational Natural Language Learning, Proceedings of the Shared Task*, *13511500200*, 74–82. https://doi.org/10.3115/v1/w14-1710

Weng, J., & Lim, E.-P. (2010). *NoTwitterRank: finding topic-sensitive influential twitterers*.

Xianghua, F., Guo, L., Yanyan, G., & Zhiqiang, W. (2013). Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, *37*, 186–195. https://doi.org/10.1016/j.knosys.2012.08.003

Xu, K., Liao, S. S., Li, J., & Song, Y. (2011). Mining comparative opinions from customer reviews for Competitive Intelligence. *Decision Support Systems*, *50*(4), 743–754. https://doi.org/10.1016/j.dss.2010.08.021

Yang, J., & Counts, S. (2010). Predicting the speed, scale, and range of information diffusion in Twitter. *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 355–358.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhang, L., & Corporation, L. (2011). RAAF Richmond. *A + U-Architecture and Urbanism*, *487*, 121.

Zhou, F., Jiao, J. R., Yang, X. J., & Lei, B. (2017). Augmenting feature model through customer preference mining by hybrid sentiment analysis. *Expert Systems with Applications*, *89*, 306–317. https://doi.org/10.1016/j.eswa.2017.07.021