

Social Network Analysis of Twitter Interactions around Technology Conferences

Xiao Cui

I Abstract:

Twitter is one of the most popular social network services. Also, it can be a valuable data source for estimating the impact of trends by analyzing social network interactions. This thesis will analyze the data from Twitter around cloud computing related technology conferences to find the “hot topics” and relevant multipliers. By using the Twitter stream and graph database to record and visualize the data. Then using a word cloud to present the "hot topics" and evaluating the correlated influencers based on their influences. Sentiment analyzer will be applied for finding their overall sentiments towards the topic. The analysis solution will demonstrate its operational functioning on two particular case studies("DockerCon" and “KubeCon + Cloud Native Con Europe”). Finally, the research shows that the number of followers cannot be the only standard to identify the users’ influence and the critical discussion of threats on validity and technical misuse will be discussed.

[Key words] Python; Data Stream; Twitter; Influence; Natural Language Processing; Social Networks Analysis; Graph Database

II Table of Contents

Contents

1	Introduction.....	4
1.1	Motivation.....	4
1.2	Goal and Organization.....	4
2	Background Information.....	6
2.1	Natural Language Processing.....	6
2.2	Sentiment analysis.....	7
2.3	Social Network Analysis.....	9
3	Solution and Comparison.....	10
3.1	Python and Important Packages.....	10
3.2	Nature Language Processing and Sentiment Analysis Libraries.....	10
3.3	Graph Database.....	15
4	Implementation.....	19
4.1	Twitter stream recording.....	19
4.2	Find the hot topic.....	21
4.3	Storing tweets into the graph database.....	23
4.4	Search and process data by Python.....	24
4.5	Find the multiplier.....	25
4.6	Sentiment analysis on tweets.....	28
5	Result of Two Use Cases.....	29

5.1	Use case 1: DockerCon 2019.....	29
5.2	Use case 2: KubeCon+ CloudNativeCon 2019.....	39
6	Threats on validity and technical misuse.....	51
6.1	The limitation to the number of tweets recorded from Twitter Streaming API.....	51
6.2	Limitations due to Twitter User Protection Terms and Ethical Considerations.....	51
6.3	Limitation of the accuracy of sentiment analysis.....	51
6.4	Limitation of the evaluation of the influence.....	51
6.5	Technical misuse.....	52
7	Conclusion & Outlook.....	53
7.1	Conclusion.....	53
7.2	Outlook.....	53
8	Acknowledgements.....	54
	Bibliography.....	55
	Appendix.....	59
	Appendix A – List of Figures.....	59
	Appendix B – List of Tables.....	60
	Appendix C – List of Equations.....	61

1 Introduction

1.1 Motivation

Since the launch of Twitter in the United States in 2006 as a social networking site by Evan William, microblogging applications have become especially popular in people's daily lives. The arrival of social media has made the communication evolve from "church style" to "market style", which enables everyone can access or build their interested contents through various media, rather than receive the contents generated and selected by publishers passively. As the introduction directly from Twitter said: "Twitter is what's happening in the world and what people are talking about right now." [1], people can state or forward ideas and participate in the discussion of different topics all over the world. Also, a large amount of social network interaction causes businesses to begin to investigate the engagement of social media. For companies and enterprises, the real voice from Twitter can help companies quickly touch consumer psychology, the feelings of the products, and the latest needs to obtain the market or detect the precursor to a dynamic and even public relations crisis.

Industrial technology conferences are used to demonstrate and report on the latest trends in software, systems, gadgets, frameworks, services, solutions, and more. To estimate the future impact of these trends corresponding triggered social network interactions might be a viable data source for evaluation.

The interactions on Twitter will raise the audiences' interests and generate more influence. For a better understanding of users, it is vital to find what users are talking about and the user who play essential roles in generating interactions like distributing their ideas or forwards contents towards some specific topics. Identifying a trend will help companies to attract more audience properly. By finding and building relationships with these users with influence can help companies to reach more target audiences because these users have the ability to propagate brands to their followers in a more convincing way.

1.2 Goal and Organization

This thesis will trace tweets on two industrial technology conferences from Twitter to find out the "hot topics" and relevant multipliers. By processing the users' tweets via the natural language processing tools will help us to find the "hot topics", then we will analysis their behaviors with social network analysis and sentiment analysis tools to identify the relevant multipliers and their sentiments toward the topic. Chapter 2 includes the background information of natural language processing, sentiment analysis and social network analysis (graphs). Then we will talk about the solution to the tasks by making comparisons of NLP, sentiment analysis libraries and graph databases for selecting tools in

Chapter 3. Chapter 4 describes the process of implementation of identifying trends and correlated multipliers. In Chapter 5, we will use specific examples to illustrate the feasibility of the method. The threats on validity and technical misuse will be discussed in the next Chapter. In chapter 7 conclusion will be made, along with the perspective to the future.

2 Background Information

To finish the task of finding the multiplier and the trends, we collected Tweets and analyzed twitter users' behavior and sentiments. The tools include natural language processing libraries, sentiment analyzers, and graph database. For a better understanding of the functions provided by these tools. In this chapter, we will introduce the fundamental information of natural language processing, sentiment analysis, and social network analysis.

2.1 Natural Language Processing

Computers are ideal for handling standardized and structured data such as database tables and financial records, and they can process faster than humans. However, humans communicate with words and sentences which are considered as unstructured data. When writing programs, syntax, and structure are needed to be used carefully. While when humans are talking with each other, they can have many options. Like making short sentences, making longer sentences or layering extra meaning in sentences. No standardized techniques can process these unstructured data for computers so that it's not an easy task for teaching machines to understand how humans communicate.

Natural Language Processing (NLP) is a branch of artificial intelligence and a computer science field in which computers derive meaning from human language and input as a way to interact with the real world [2]. It transforms human language into a machine-understandable, structurally complete semantic representation. The goal is to let computers understand and generate human language. Natural language processing challenges often involve recognition of speech, understanding natural language, and generating natural language. With NLP, human-readable tweets can be transferred to machine-readable formatted data, and then these data can be used for machine to process and analyze the sentiments.

Terminology: NLU vs. NLP vs. ASR

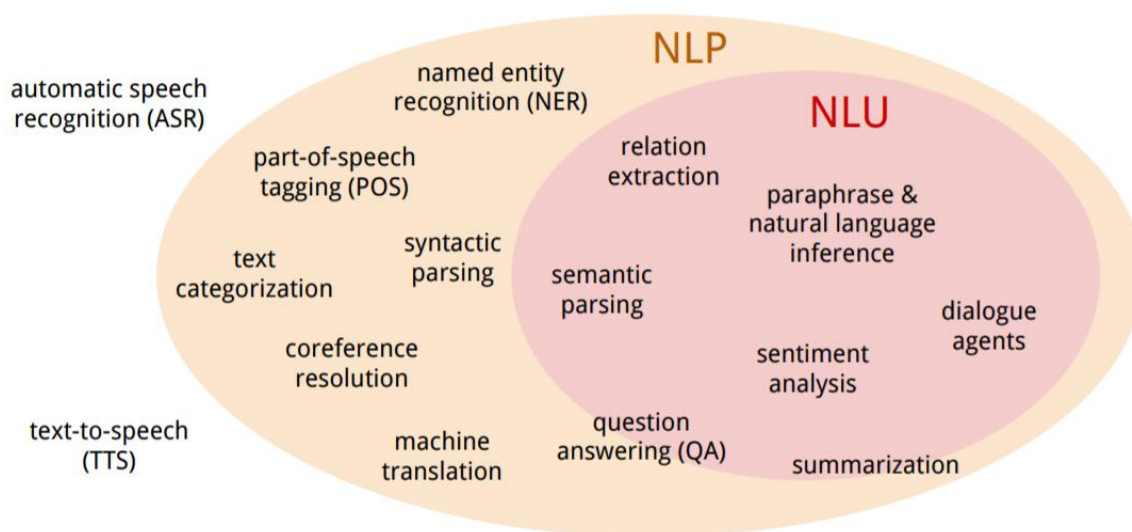


Figure 1: Comparison between NLP and NLU [3]

Common terminology in the area of machine learning is easy to confuse people. As Figure 1 shows, in addition to Natural Language Understanding (NLU), NLP also includes an understanding of the previous processing stages and the application phase after the understanding. That is to say, NLU is a subset of NLP. They are not a union, or equal.

When analyzing tweets recorded from Twitter, the machine needs particular methods to transfer unstructured tweets into structured data for it to understand. Then the sentiment analysis will be applied to analyze the sentiment from these structured tweets. The whole process can be considered as natural language processing.

2.2 Sentiment analysis

Human communication not only contains words and their explicit meanings.

Sentiment analysis is based on the context of text, then identifies and analyzes the information of each user from the source material. It will help a business to understand the social sentiment of their brand, product or service [4]. In the area of social media monitoring, using sentiment analysis is extremely useful to gain public opinion on specific topics. By extracting insights from social media helps us to detect public emotions towards two conferences in cases studies. The first is the container industry conference “DockerCon”. Second is the conference “kubernetes + Cloud Native Con Europe” mainly about open source and cloud native communities.

The sentiment is people's attitude or experience towards objective things. It is a subjective term and has a relationship with the characteristic of a human's mind so that it can not be observed and verified in an objective way [5].

The requirements towards machine are to do calculations on the sentiment which is also considered as the ability to understand and generate human's emotion.

To obtain emotion and have communication with the human, the machine needs to obtain resource of human's emotion which are very helpful for the machine to understand and learn how to recognize or generate emotion. Social media is one of the effective ways to observe human's emotion, it includes various social activities such as shopping, chatting, community, news, lifestyles. Not only text resources but also emotional resources exist are included because these activities are all revealing people's emotion on several aspects. For example, the comments made from Yelp contain emotion on products or services. These comments are made on these medias every day. The sentiment calculation on text from social media depends on not only text but also on users and group's information. Then text emotions are analyzed, processed, and summarized so that sentiment analysis has better pertinence and precision.

The difficulty of Sentiment Analysis with inappropriate English

Informal language refers to the use of colloquialisms and slang in communication, using spoken language conventions [6]. Such as “can not” as “can’t”. Some systems are not able to analyze the sentiments from the inappropriate English words for that it will disturb the process of decision making.

An emoticon is short for "emotion icon" [7], which can represent a facial expression. By using characters like punctuation marks, numbers, and letters. It expresses a person's feelings or mood, or as a time-saving method. As humans often turn to emoticons to adequately express what they cannot put into words [8]. For example, The emoticon '☺' indicates the mind of happiness. While machine maybe does not have a related database to recognize and extract the emotion from these icons. It will cause the loss of the sentiment from the whole sentence which contains emoticons.

Short-formed words are also used in short message service (SMS) widely. Because of the limitation of the length of Twitter, some users prefer using short-formed words to make their tweets satisfy the requirement of length limitation. For example, ‘ROFL’ refers to ‘Rolling on floor laughing’ and ‘LOL’ means ‘Laughing Out Loud’.

As marketing becomes more focused on a daily basis, automated classification of users into cohorts can simplify the marketer's life. Marketers can monitor and classify users based on how they discuss a product or brand online. The classifier can be trained for identifying promoters or detractors and let this group of people better serve the brand.

On Twitter, users write tweets to express their ideas or opinions on some topics which are related to their daily lives. Moreover, the data generated by an increasing number of users can be the source for the task of sentiment analysis and opinion mining [9]. The public's sentiments represent the positive or negative or indifferent attitude feedback from the audience, and it brings the interaction that the traditional media can't do. Collecting this feedback to a certain amount can have overall representativeness although in the case of small errors, the data of feedback and user portraits can be obtained.

2.3 Social Network Analysis

Social network analysis (SNA) is the strategy of researching social structures by using networks and graph theory [10].

A network can be represented as a graph containing two basic elements node and edge. Edges can be directional. If the edge is directional, then the graph is called a directed graph. Otherwise, it is called an undirected graph. The graph reflects a certain correlation between points, which is represented by edges.

In Twitter's network, users and tweets can be nodes of the network. The user generates the tweet to the network, then the relationship of "posts" is generated between the user and the tweet. If someone wants to mention others in his tweet, he can use "@" then with the Twitter user name. And they can include "#" with some keywords in their tweets which are also called hashtags to participate some discussion on specific topics. For the action of retweeting and quoting, the former means the user repost other's tweet without his own words and the later tweet will include both other's and user's words like user commenting to other's idea. For the action of replying, user can just reply to someone's tweet.

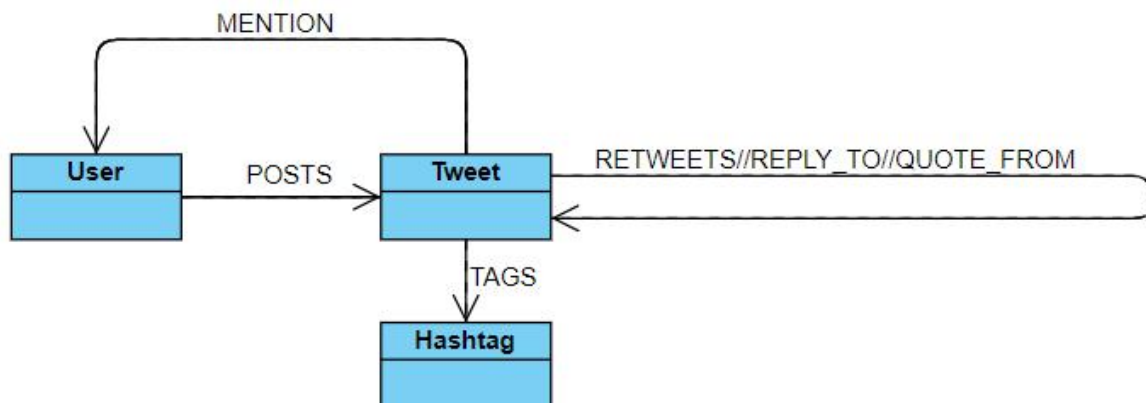


Figure 2: Example of nodes and relationships on Twitter

Figure 2 shows the example chart of showing the relationships between nodes. First, if a user wants to post, retweet, or quote a tweet. A tweet will be posted by the user and then other relationship will be attached to the tweet like "retweets" or "quote_from". User can be mentioned in every tweet and hashtags are also able to be attached to each tweet. Every tweet can be retweeted, replied, quoted by other tweets posted by users.

While from twitter stream API, we can only get the JSON formatted tweets. "JSON (JavaScript Object Notation) is a lightweight data-interchange format. [47]" For better analyzing of the social network, a tool which is able to extract the information from JSON formatted tweet data and realize the visualization of relationships and nodes in the social network is needed.

3 Solution and Comparison

To achieve the goal of social network analysis, necessary tools are required to collect and analyze the data. This chapter will make comparisons of existing NLP, sentiment analysis libraries, and graph databases for testing the usability and necessity for the study.

3.1 Python and Important Packages

Python is a dynamic, interpreted, and famous programming language which focuses on readability of the code. Python's syntax helps programmers code less than Java or C++ steps. It was created by Guido van Rossum, and released in 1991 [11]. Also, it can be written once and run on many platforms. Besides thousands of third-party modules which are available from the Python Package Index, plenty of documentation, guides, tutorials, and active community can give support to developers includes beginners.

Python is also efficient and useful for both the physical sciences and data sciences. NumPy and SciPy packages in Python are of considerable significance to the field of scientific computing and analysis [13]. NumPy is the essential package for scientific computing with Python, which can be used as an efficient multi-dimensional container of generic data. Arbitrary data types are able to be defined, which allows NumPy to integrate with a wide variety of databases seamlessly and quickly [14]. For analyzing part, NumPy will be essential to make the process easier with its flexible array objects. The package called Matplotlib which can be used for presenting the result of the analysis. It is an easy-use plotting library which can run with Python programming language, and NumPy can be the extension for its numerical mathematics.

As mentioned above, Python provides many useful and scientific packages coming from academia and industry. In this thesis, we will use Numpy, NLP libraries and Matplotlib for data processing, analyzing, and visualization. Also, some other drivers connected to other systems are also provided by Python.

3.2 Nature Language Processing and Sentiment Analysis Libraries

Nowadays, some already written sentiment library can be used. Their main purpose is to simplify text preprocessing so that researchers can focus on building machine learning models and modifying parameters. Here are some popular and helpful natural language processing libraries to solve the problem of sentiment analysis:

- NLTK (Natural Language Toolkit)
- Spacy [48]
- Scikit-learn [49]
- Gensim [50]
- Polyglot [51]
- TextBlob

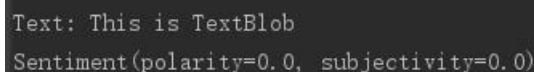
Compared with NLTK, Spacy lacks flexibility. Although both of Scikit-learn and Genism provide a wide variety of algorithms for solving NLP problems, mostly they will be used with machine learning and requires other libraries to build a chain of independent modules in which each one taking as an input the output of the module before it. It seems not to satisfy the requirement of light-weight and efficiency. Polyglot supports many languages while it is not as popular as NLTK and lacks community support, knowledge with high quality generated by users in the community may be hard to find when encountering problems [15]. As the most well-known and full NLP library, NLTK can be seen as a powerful and comprehensive library with over 50 corpora and lexicons and dozens of algorithms to choose from [16]. Compared with NLTK, TextBlob provides more functions. And it is built on NLTK while is not as vast as NLTK [17]. So it can be used for fast-prototyping or building applications that don't require highly optimized performance. Due to the consideration of time and operational difficulty, this thesis will use TextBlob as the tool of the sentiment analysis.

In TextBlob, three ways of sentiment analysis are provided. It provides two pretrained analyzers: PatternAnalyzer and NaiveBayesAnalyzer, and the option to build a custom sentiment analyzer.

Using TextBlob's default analyzer (PatternAnalyzer) is a simple and fast way [18].

```
1. from textblob import TextBlob
2. sentence = TextBlob("This is TextBlob")
3. print("Text: This is TextBolb")
4. print(sentence.sentiment)
```

This analyzer will return the result of "popularity" and "subjectivity". The factor "popularity" which ranges from -1 to 1, positive means the popularity is above 0, negative means below 0, neutral is 0. "subjectivity" will shows subjective degree of the sentence. Figure 3 is the result of the analysis of the sentence "This is TextBlob".



```
Text: This is TextBlob
Sentiment(polarity=0.0, subjectivity=0.0)
```

Figure 3: Example result of TextBlob API

Also, it can also use the other pre-trained analyzer(NaiveBayesAnalyzer) from the library [19].

```
1. from textblob import TextBlob
2. from textBlob.sentiments import NaiveBayesAnalyzer
3. print("Text: This is NaiveBayesAnalyzer")
4. sentence = TextBlob("This is NaiveBayesAnalyzer", analyzer = NaiveBayesAnalyzer)
5. print(sentence.sentiment)
```

As shown in Figure 4, NaiveBayesAnalyzer has "classification" to show the sentiment, but it is also determined by "p_pos", which is range from 0 to 1. If "p_pos" exceeds 0.5, it will be positive. Negative means "p_pos" is below 0.5, "neutral" is 0.5.

```
Text: This is NaiveBayesAnalyzer
Sentiment(classification='neg', p_pos=0.49578281497100696, p_neg=0.5042171850289932)
```

Figure 4: Example result of NaiveBayesAnalyzer

Not only TextBlob provides a simple and easy way to do analyzing work, but also the custom analyzer which needs a data source for training [20].

```
1. from textblob.classifiers import NaiveBayesClassifier
2. with open('train_set1.csv','r',encoding="utf-8") as fp:
3.     cl = NaiveBayesClassifier(fp, format="csv")
4.     print("Text: This is NaiveBayesClassifier")
5.     print(cl.classify("This is NaiveBayesClassifier"))
```

NaiveBayesClassifier needs datasets contains texts and labels then it will classify the input with the trained classifier. Finally, it will match the input to the label as a result shown in Figure 5.

```
Text: This is NaiveBayesClassifier
positive
```

Figure 5: Example result of NaiveBayesClassifier

Three useful tools are mentioned from above, so a comparison among PatternAnalyzer, NaiveBayesAnalyzer, and NaiveBayesClassifier will be made.

First part is the choosing of the sentiment dataset because NaiveBayesClassifier needs a dataset for training. Two datasets are selected to fulfill the task are shown in Table 1.

	4A English	Sentiment 140
Description	A compilation of annotated sentiment datasets for SemEval-2017 task 4	Created by Computer Science graduate students at Stanford University, it helps researchers to analyze the sentiment of the topic on Twitter.
Source url	http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools	http://help.sentiment140.com/for-students

Table 1: Description of two datasets

Taking consideration of CPU loads, 2000 tweet data will be divided randomly for the training of the classifier and 20000 tweet data will be used as the test set.

The second part is to calculate the accuracy of these tools. Because of the dataset Sentiment Analysis 140 does not have neutral tweets so the "neutral" tweets identified by each tool will take part in the calculation of the accuracy. We will call 4A English as dataset1 and Sentiment 140 as dataset2.

The first step is to “clean“ all the tweets by removing all the url, mention lable, hashtag, and punctuation.

```

1. def clean_words(data):
2.     data = re.sub(r'(https|http)?://(\w|\.|\/|\?|\=|\&|\%)*\b', '', data)
3.     data = re.sub('@[\w]*', '', data)
4.     data = re.sub('#[\w]*', '', data)
5.     data = re.sub("[\s+\.!\/_,$%^*(+\"'\']+|[+—!,. ?\~@#¥%.....&*
   () ]+", " ", data)
6.     return data

```

Then the dataset are needed to be transferred to the formatted csv.file for classifier training. So we need to see the structure of two datasets.

Table 2 shows example tweets of dataset 1 and It has all the sentiments positive, negative, and neutral.

Table 2: Example tweets from dataset1

Id	Sentiment	Text
621149890161307648	positive	K-Rod is just trying to set up Ryan Braun for a walk off grand slam in the 9th. He's all about making his teammates look good. #selfless
621160187680944128	neutral	Ryan Braun leadoff triple to start the 9th. Casual. @BrewerNation
621160246220824576	neutral	AL 6-2 NL (B9) Ryan Braun the last batter to be used by the NL and he squeezes it up the 1st base line for a triple! Surely not...
621233092997578752	negative	"Road accident is the 2nd highest source of violent death after Boko Haram in Nigeria.Yet,no data to warn citizens @BudgITng #DATARevolution"

Table 3 shows the example tweets from dataset2, as the official website said that 0 means negative and 4 means positive. While no neutral sentiments included.

Polarity	Id	Date	Query	Username	Text
0	2329205574	Thu Jun 25 10:28:30 PDT 2009	NO_QUERY	davidlmlulder	@Eric_Urbane Sounds like a rival is flagging your ads. Not much you can do about that though
0	2329205794	Thu Jun 25 10:28:31 PDT 2009	NO_QUERY	tpchandler	has to resit exams over summer... wishes he worked harder at first year of uni...
4	1467822272	Mon Apr 06 22:22:45 PDT 2009	NO_QUERY	ersle	I LOVE @Health4UandPets u guys r the best!!
4	1467822273	Mon Apr 06 22:22:45 PDT 2009	NO_QUERY	becca210	im meeting up with one of my besties tonight! Cant wait!! - GIRL TALK!!

Table 3: Example tweets from dataset2

After extracting the sentiment and text and saved as formatted csv files, two train sets and test sets are ready for use. We can start using 3 analyzers mentioned above for counting the accuracy of each analyzer. The accuracy will be calculated as the result of the number of all the tweets where the sentiments analyzed by the analyzer match the sentiment provided by the dataset divided by the number of all the tweets for test.

Table 4 is the result of accuracy:

Accuracy Result	Dataset 1	Dataset 2
NaiveBayes Analyzer	0.53	0.54
PatternAnalyzer	0.67	0.68
NaiveBayesClassifier (trained with dataset 1)	0.58	0.57
NaiveBayesClassifier (trained with dataset 2)	0.67	0.68

Table 4: Result set of accuracy

As can be seen from Table 4, NaiveBayesAnalyzer seems to work not very well. NaiveBayesClassifier works well when dataset2 is the training set. While its performance goes down with the dataset1 that

means the performance of the classifier mainly depends on the training dataset. Also, it takes time and causes CPU consumption during the training step. The Pattern Analyzer works well with both datasets so it can be a promising and lightweight tool for sentiment analysis.

3.3 Graph Database

Graph Database origins from Euler Graph theory and it is called the graph-oriented database. The basic meaning of the graph database is using the graph as the data structure. It stores nodes of data and the edges to represent the relationship between the nodes [21]. The advantage of the graph database is to quickly solve complex relationship problems.

Although the SQL database is a very useful tool, after 15 years the dominance is about to be broken. This is only a matter of time: people were being forced to use a relational database, but eventually found that the situation can not adapt to the needs in some situation.

Complete trend, starting with January 2013

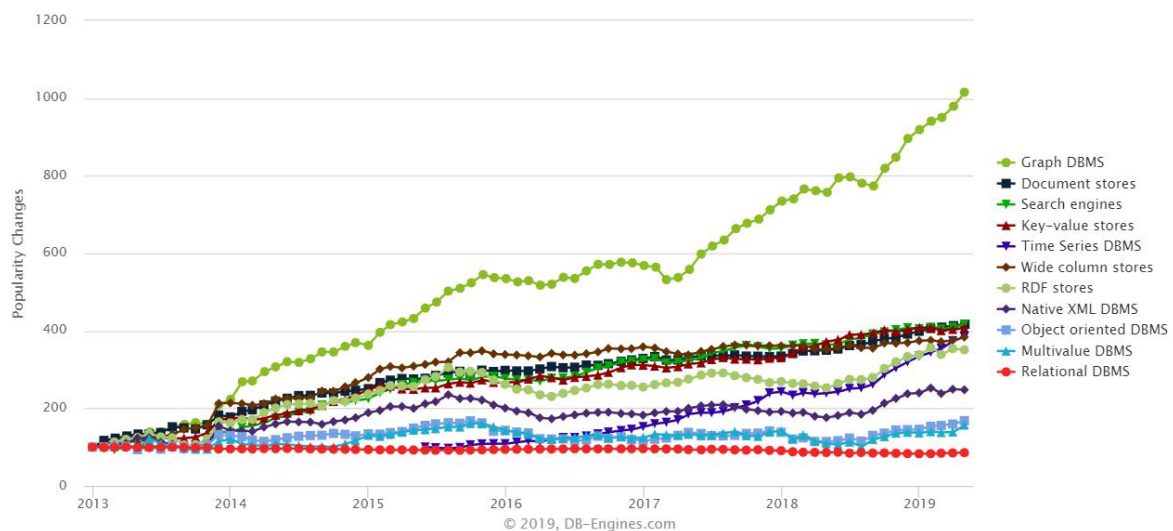


Figure 6: Trend of database [22]

The trend of the graph database is growing annually, as Figure 6 from db-engines.com which is the website provides lists of DBMS ranked by their current popularity.

3.3.1 Comparison with SQL

The relational database represented by Mysql has existed for a long time. It has always been the driving force of the database field. Highly structured data was stored in a two-dimensional table and must be strictly operated according to relevant conventions (such as outside Key constraint).

However, relational databases need to formulate relevant agreements before they are built. Tables and tables have mutual constraints and mutual references. As the Figure 7 shows, if the database continues to grow, the relationship of mutual constraints will increase, and the number of operations that perform search matching will increase exponentially, which in turn consumes a large number of resources [21].

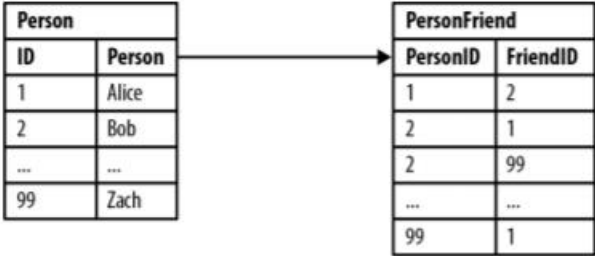


Figure 7: Example of the model friends and friends-of-friends in a relational database[21]

For example, when you want to query "Friends of Jake", the relational layer database will involve some expensive indirect layers, such as querying with an index table. However, if asked "Friends of friends of Jake's friends..." the depth is increasing, and each additional layer is added with an index table so that the indirect layer is increased. Queries are getting slower and slower, and the memory overhead required is getting bigger and bigger [21].

In contrast, the graph database has a unique advantage, querying relationships within a graph database is faster because they are already stored in the database itself. If asked "Friends of friends of Jake's friends...", even the depth of searching increases while only nodes and relationships are added. Also, the structure of the existing network and original data will not be distributed [21]. So that even a query has complex connections, the millisecond level can be reached. In the Twitter social network, multiple nodes are connected by a limited number of relationships. For faster storing and searching these nodes with specific relationships, using graph database will be a suitable solution.

3.3.2 Comparison of Graph databases

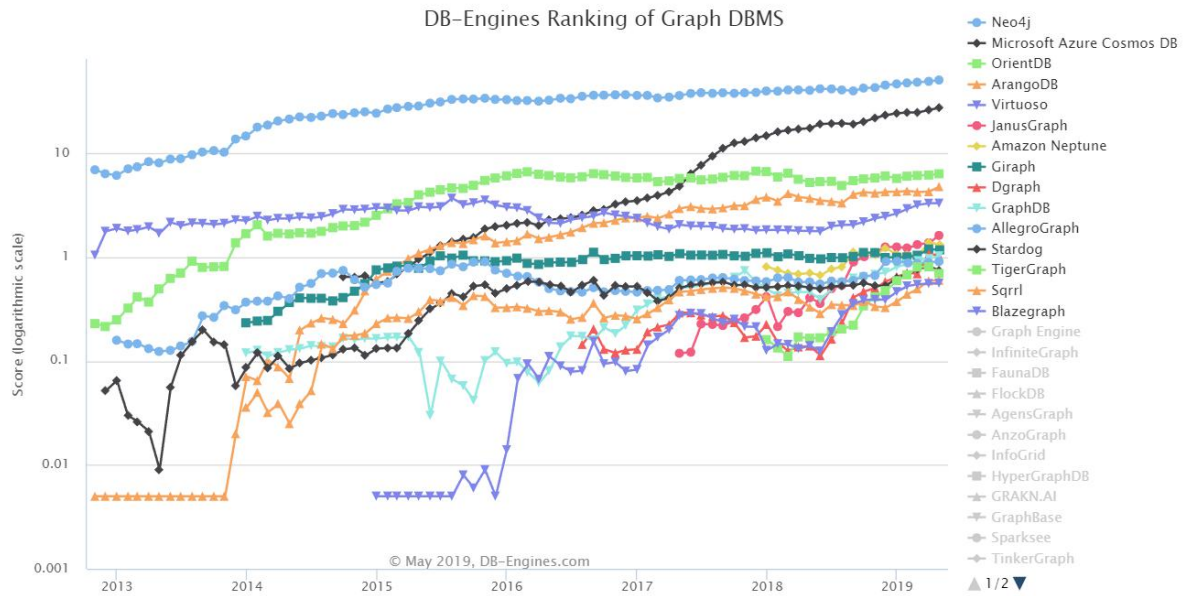


Figure 8: Ranking of the graph database [23]

Figure 8 shows the ranking of graph database until May 2019, Neo4j ranks number one among the graph databases. Then comes with the Microsoft Azure Cosmos DB, OrientDB, ArangoDB. Taking the consideration of open source, Microsoft Azure Cosmos DB will be excluded.

According to Table 5’s brief overview of these databases, building a social network needs the traversal of the graph. Neo4j’s index-free adjacency can satisfy the requirement and calculate a graph of nodes which do have a large number of relationships. Other databases consume time for querying and computing relationships through JOIN operations, Neo4j stores connections alongside the data in the model. It is consistent with the characteristics of the Twitter network which has a massive amount of nodes but few relationships.

Table 5: Comparison of three most top-ranked open source graph databases [24]

Name	ArangoDB	OrientDB	Neo4J
Description	multi-model DBMS	multi-model DBMS	graph database
Primary database model	Document store, Graph DBMS, Key-value store	Document store, Graph DBMS, Key-value store	Graph DBMS
Server operating systems	Linux, OS X, Raspbian, Solaris, Windows	All OS with a Java JDK (\geq JDK 6)	Linux, OS X, Solaris, Windows
Transaction concepts	ACID	ACID	ACID
Foreign keys	No	Yes	Yes

3.3.3 Introduction of Neo4j

Neo4j is an open-source, NoSQL, native graph database which stores structured data on the network instead of in the table. For beginners, Neo4j has already encapsulated many algorithms so that they do not have to study many difficult algorithms. Also, it provides Cypher which is a declarative query language similar to SQL and drivers for popular programming languages includes Python [25].

Neo4j has broad application prospects in social, retail, financial, credit, IT management and other fields. Because of its graph-oriented theory, It has significant advantages in dealing with social networks, logistics and transportation, recommendation systems, fraud detection, and relationship analysis. For example, Wal-Mart used Neo4j to implement retail recommendations of products [27].

As the comparison made above, we can use Python as programming language. It makes programming more efficient with some useful scientific packages like Numpy for computing and Matplotlib for data visualization. Also, It has the library TextBlob and the toolkit Py2neo for working Neo4j [57]. So TextBlob and its PatternAnalyzer can be used for natural language processing and sentiment analysis. Then using Neo4j for storing and searching Twitter network via its graph database.

4 Implementation

This chapter will demonstrate the whole process of implementation of finding the “hot topic” and the “multiplier” on specific topics.

4.1 Twitter stream recording

First, tweets are needed to be recorded from Twitter streaming. While a massive amount of tweets is being generated in Twitter every second, some methods needed to be applied for narrowing down the range of the search to trace the trend of the specific topic and recording these tweets locally. Twitter provides a lot of useful functions, and developers can use them by calling application programming interfaces(API). With API, applications and developers have the ability to access a set of routines based on a piece of software or hardware without having to access the source code or understand internal work and the details of the mechanism.

The first step is to gain access to Twitter’s API by registering and creating an application to obtain the token keys for the API. Once the app has been created, a Consumer Key, Consumer Secret, Access Token, and Access Token Secret can be obtained.

Once the application has been created and approved, the next step is to connect to the Twitter API. Tweepy is a Python library for accessing the Twitter API and makes it easier to use the Twitter Streaming API by handling authentication, connection, and other things [28].

Now we need to find and record tweets on the specific topic from the Twitter Network.

According to the official introduction:

“A hashtag—written with a # symbol—is used to index keywords or topics on Twitter. This function was created on Twitter, and allows people to follow topics they are interested in easily.” [29]

Using hashtag is a better choice for that users can use this as a keyword for categorizing and searching a specific topic or event easily and allowing them to locate and contribute to the discussion [30].

Twitter provides a lot of useful APIs for developers to use to record the tweets. So with the hashtag, this API can be used “POST statuses/filter” for filtering tweets with specific hashtags. Also, the hashtag is not case-sensitive.

From these conferences' official websites we can find its official Twitter account. The hashtag mentioned in the twitter account's description will act as an official hashtag for users to use. With this hashtag, most related tweets about this topic can be traced. But some associated tweets with other hashtags have the possibility of being excluded, so these hashtags are supposed to be considered.

To predict these hashtags, the first solution is to check the website which can find related hashtags with one inputted hashtag. One website called Hashtagify (hashtagify.me) provides the service and 10 related hashtags will be returned for free.

And then we can compare these results with the hashtags counted from the Tweets recorded with the “official” hashtag, the amount of tweets is around 800 to 1000.

Then same hashtags will be found between both two result sets and use them for recording. And with these hashtags, if related tweets on specific topics are received from TwitterStreamingAPI. The “listener()” will be used for handling these tweets and we will talk it in detail later.

```
1. while True:
2.     try:
3.         print("start")
4.         auth = OAuthHandler(consumer_keys, consumer_secret_keys)
5.         auth.set_access_token(access_token, access_token_secret)
6.         twitterStream = Stream(auth, listener())
7.         twitterStream.filter(track=["#official hashtags"],
8.                               languages = ["en"])
9.     except:
10.        continue
```

Meanwhile, network fluctuation will interrupt the recording process by sending an error message from the Twitter server. This problem can be solved by adding a try and except block as the codes shown above, the program will catch the error message and then continue to recording tweets.

Finally, with the codes of class “listener()” shown below. Every tweet will be recorded and saved as JSON files which each file contains 500 tweets.

```
1. class listener(StreamListener):
2.     def on_data(self, data):
3.         global index, tweets, file_index
4.         tweet = json.loads(data)
5.         tweets.append(tweet)
6.         tweets_per_file = 500
7.         print(index)
8.         index = index + 1
9.         if(index==tweets_per_file):
10.            work_dir = os.path.dirname(os.path.abspath(__file__))
11.            json_path = os.path.join(work_dir, 'twitter_stream')
12.            with open(json_path+str(file_index)+".json", 'a') as my_file:
13.                my_file.write(json.dumps(tweets))
14.                index = 0
15.                file_index = file_index + 1
16.                tweets = []
17.        return True
18.     def on_error(self, status_code):
19.         print(status_code)
```

Now, all the related tweets can be saved for later use.

Meanwhile, we need to consider the running environment of the program. According to the Windows 10 End User License Agreement (EULA), Windows Update can't be readily disabled in Windows 10 Home and Professional version because of the license terms that all users must agree to allow Microsoft to install updates automatically [43]. If the computer was running in Windows OS and not properly set, automatically update will restart the computer. It will terminate the program when recording tweets from Twitter API and cause a loss of data of tweets for analyzing. Switch Wi-Fi settings to a metered connection is a proper way to prevent Windows downloading updates automatically.

4.2 Find the hot topic

Word cloud is a novelty data visualization technique used to represent text data. The size of the word indicates its frequency and importance. A word cloud can be used to highlight important text data points in a simple, easy, and impressive way. They are widely used to analyze data from social networking websites.

In Python, the Package word cloud was developed by Andreas Mueller [31]. Figure 9 is an example of the word cloud:

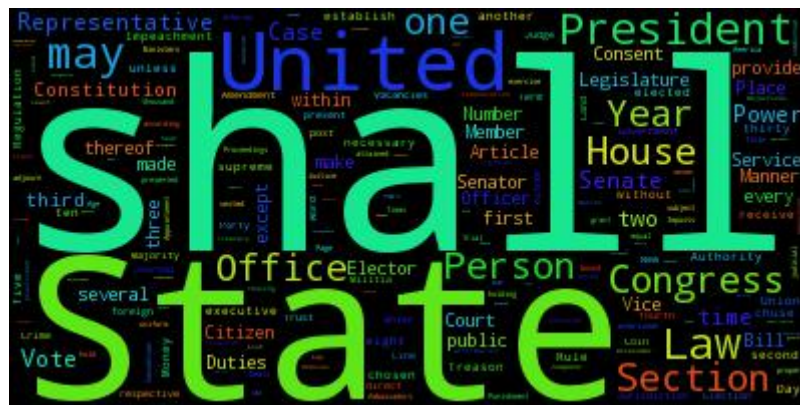


Figure 9: Example of a word cloud[21]

With word cloud, the hot topic from a large number of tweets can be demonstrated clearly because it shows the high-frequency words on the topic.

To make a word cloud, clean the row tweets is very essential.

According to the tutorial from a blog written by Stephen Hsu[32]. First, we need to clean all the tweets, all URL links, special characters, unicodes, and extra white spaces must be removed from each tweet.

```
1. raw_string = ''.join(tweet_list)
2. raw_string = re.sub(r'http\S+', '', raw_string)
3. raw_string = re.sub(r"\\[a-z][a-z]?[0-9]+", '', raw_string)
4. raw_string = re.sub('[^A-Za-z ]+', '', raw_string)
```

Then split the tweets into separated words and make all the letters become lower case.

Also, the word which contains less than 3 characters can be the meaningless word or discourse marker. These words are also supposed to be removed before processing [32].

Later, the stop words are needed to be removed. In computing area, stop words are the words that are filtered out before or after natural language data is processed. Like the words “the” or “and”, they help to link the context of the text while they do not carry the meanings by themselves [33]. Since stop words with little meaning that show up in the content and serve just a syntactic capacity. They will not demonstrate the subject and have a high frequency of appearing in the text which will interference the counting of related words.

```
1. words = raw_string.split(" ")
2. words = [word.lower() for word in words]
3. words = [word for word in words if len(word) > 2]
4. words = [word for word in words if word not in STOPWORDS]
```

Then, counting every word’s frequency of appearing is needed. Python library Collection provides the function Counter() for solving the problem like this, and we can transfer the result object to dictionary for later counting.

```
1. wordCountDict = dict(Counter(words))
2. print(wordCountDict)
```

For better visualization of key words which imply the hot topic. The word whose frequency above the average will be selected for later visualization:

```
3. def find_hot_words(wordCountDict):
4.     value_key_list = []
5.     for key,value in wordCountDict.items():
6.         value_key_list.append((value,key))
7.     value_key_list.sort(reverse=True)
8.     hot_words = []
9.     count = []
10.    for value,key in value_key_list:
11.        count.append(value)
12.    count = np.array(count).astype(np.int)
13.    average = np.average(count)
14.    print(average)
15.    for val,key in value_key_list:
16.        if(int(val)>average):
17.            hot_words.append(key)
18.    return hot_words
```

Now, it is the time to generate the word cloud to see key words of the hot topic:

```
1. hot_words = find_hot_words(wcdict)
2. wordcloud = WordCloud(
3.     background_color='black', max_words=200,
4.     max_font_size=50, scale=3,
5.     repeat= False,
6. ).generate(str(hot_words))
7. plt.axis('off')
8. plt.imshow(wordcloud,interpolation="bilinear")
9. plt.show()
```

4.3 Storing tweets into the graph database

Figure 10: Structure of Tweet's JSON

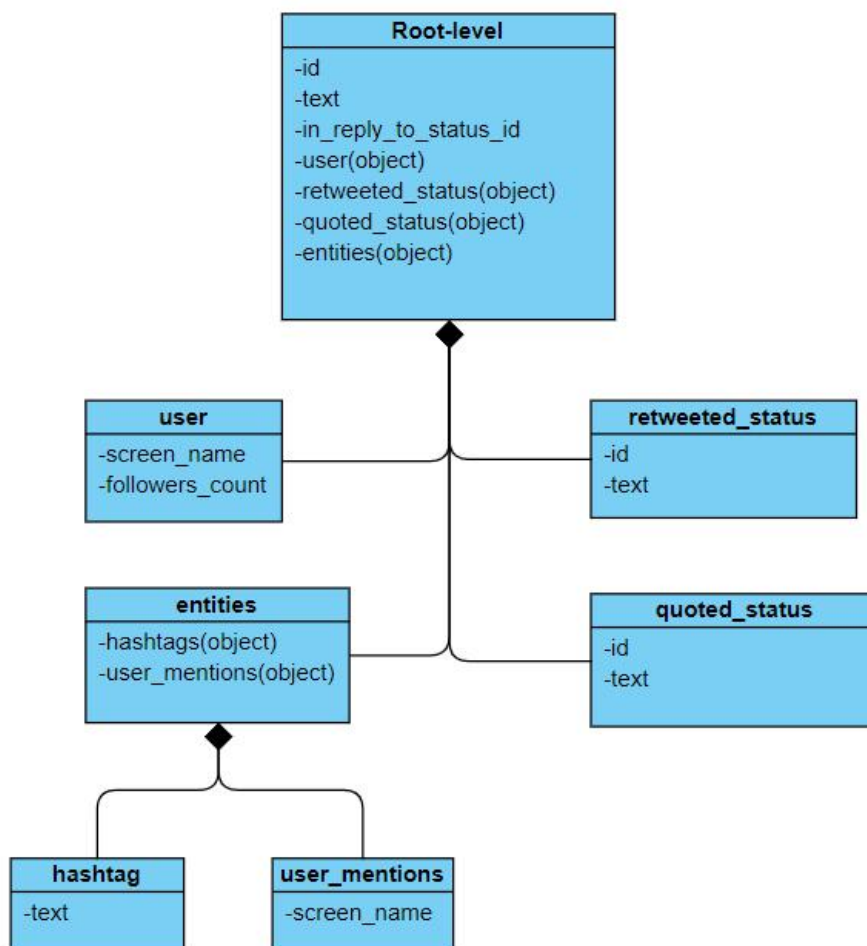


Figure 10 shows the structure of Tweet objects according to the introduction from Twitter websites [34]. The poster's information was stored on the 'root-level' of the JSON. If a Tweet created as a

retweet or quote or reply of other tweets, the information of 'original' tweet will be provided in the "retweeted_status/quoted_status/in_reply_to_status". The "entities" object contains "mentions" and "hashtags".

With the JSON files which contain recorded tweets, Neo4j can be applied for storing the data and visualize it in its graph database. According to the data structure of the tweet, the nodes are "User", "Hashtag" and "Tweet", with relationship "POSTS", "RETWEETS", "QUOTE_TO", "TAGS", "REPLY_TO" and "MENTIONS".

These entities can be extracted from JSON formatted data from Twitter Streaming. Original codes are referenced from the user of Github "nicolewhite" [35], and some improvements were applied to solve the problem of the overwriting the nodes' information and searching to fulfill the information of mentioned users.

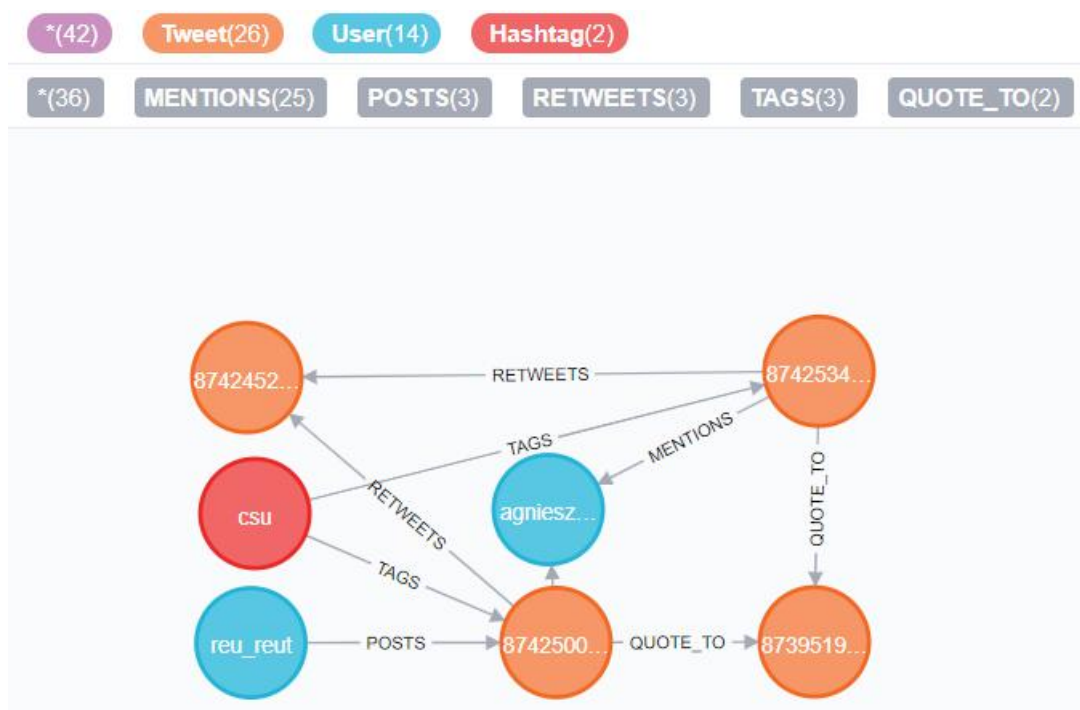


Figure 11: Overview of the part of data from Neo4j

Figure 11 shows the example overview of the graph database after storing.

4.4 Search and process data by Python

With Neo4j, the JSON formatted tweets can be stored as nodes and relationships in the graph database. While with Cypher, a human-readable query language for Neo4j, searching and counting the relationship between nodes will be more easily. By using the Python module "Py2neo", results can be returned to python from Neo4j with properly Cypher.

First, we need to connect to the graph database. And the setting of connection is very simple:


```

1. from py2neo import Graph,Node,Relationship,NodeMatcher
2. graph = Graph(url,username=username,password=password)

```

Then, we can use the graph for running the query language Cypher for searching. For example, if someone wants to find the Tweets posted by a specific user by search the user's username like "Tom". The Cypher could be :

```

1. Match(n:User{username:"Tom"})-[:POSTS]->(t:Tweet)
2. Return n,t

```

In the Python, the code will be :

```

1. def find():
2.     query = graph.run("""\
3.     Match(n:User{username:"Tom"})-[:POSTS]->(t:Tweet)
4.     Return n,t
5.     """)
6.     return query

```

This will return the user and all the tweets he posted and the results can be set to four formats: Graph, Table, Code, and Text. Python can use formatted results to analysis and find the solution to the problems. With proper Cyphers, the later analyzing work will be easier.

4.5 Find the multiplier

The typical way is used to identify the multiplier is to find the user with the most followers, while a user with most followers may not has the most times of being mentioned or retweeting tweets[36]. Like considering a famous pop star who retweets a tweet about a technical conference then he was considered as this conference's multiplier, but most of his followers will not pay attention to this or participate in this topic, which will not generate the interactions with other users in the Twitter social network based on the topic. So other factors need to be considered, as the number of retweeted tweets and the times of being mentioned in others' tweets on the topic. Both of them indicate the influence of the user [36]. The action of retweeting tweets means users acknowledge the value of the tweet and be willing to expose this to their network so that the original tweet can be exposed to more people. A user calls the attention of another user by mentioning others in a tweet. This means the ability of a user's influence to participate in a conversation.

So, first we need to find the multiplier by evaluating their ability to spread the topic. The ability can be calculated with a simple formula as Equation 1 :

$$\text{Spreading Score} = \text{Followers} * \text{Retweeting tweets}$$

Equation 1: Calculation of spreading score

It shows the user's ability to spread topic tweets by retweeting others' tweets and exposes them to their followers.

The time of user being mentioned and the number of their tweets being retweeted can be used as criteria for evaluating a user's influence. Then the average value of these two factors needs to be calculated. Because we need to find someone above it.

For better visualization, the number will be displayed in logarithm form.

Figure 12 is the example chart with the factors of spread score, mention times and retweeted times.

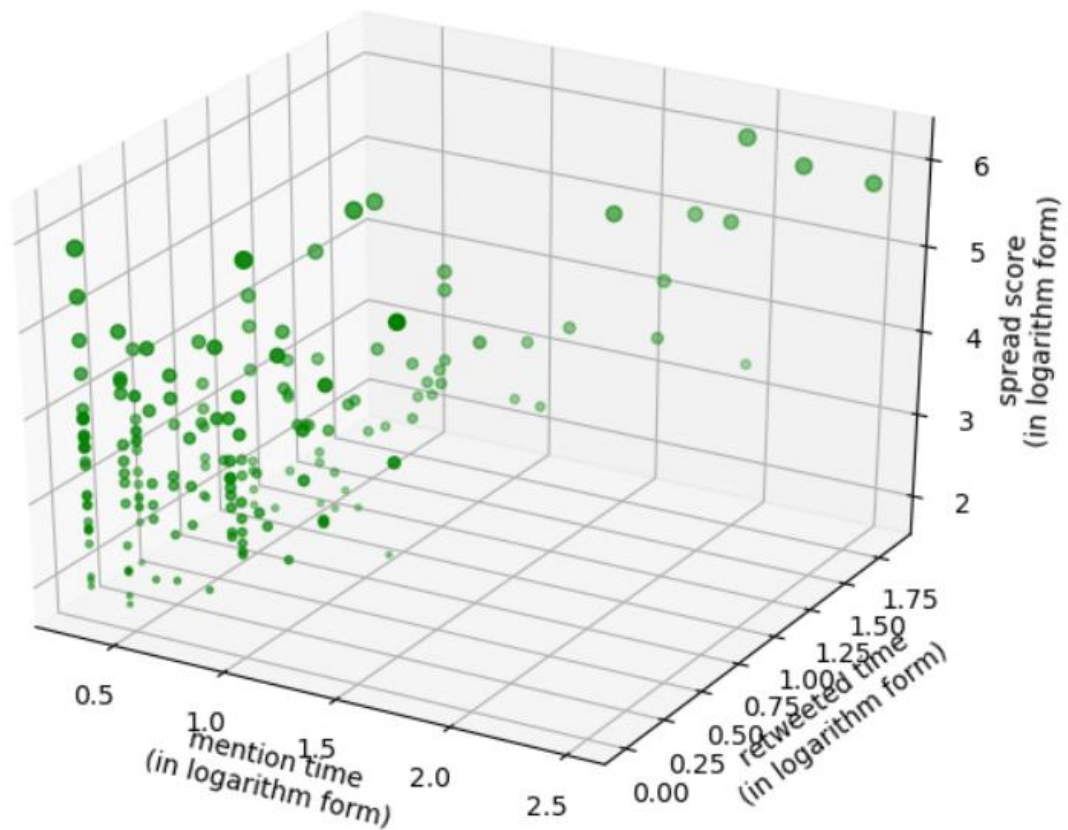


Figure 12: Example 3d scatter chart with the factors of spread score, mention times and retweeted times

Figure 13 is the example 2D scatter chart with the factors of the spread score and retweeted time. Besides, one horizontal and one vertical line will indicate the average value. Also, the chart has relationship with mention time will be displayed in same way.

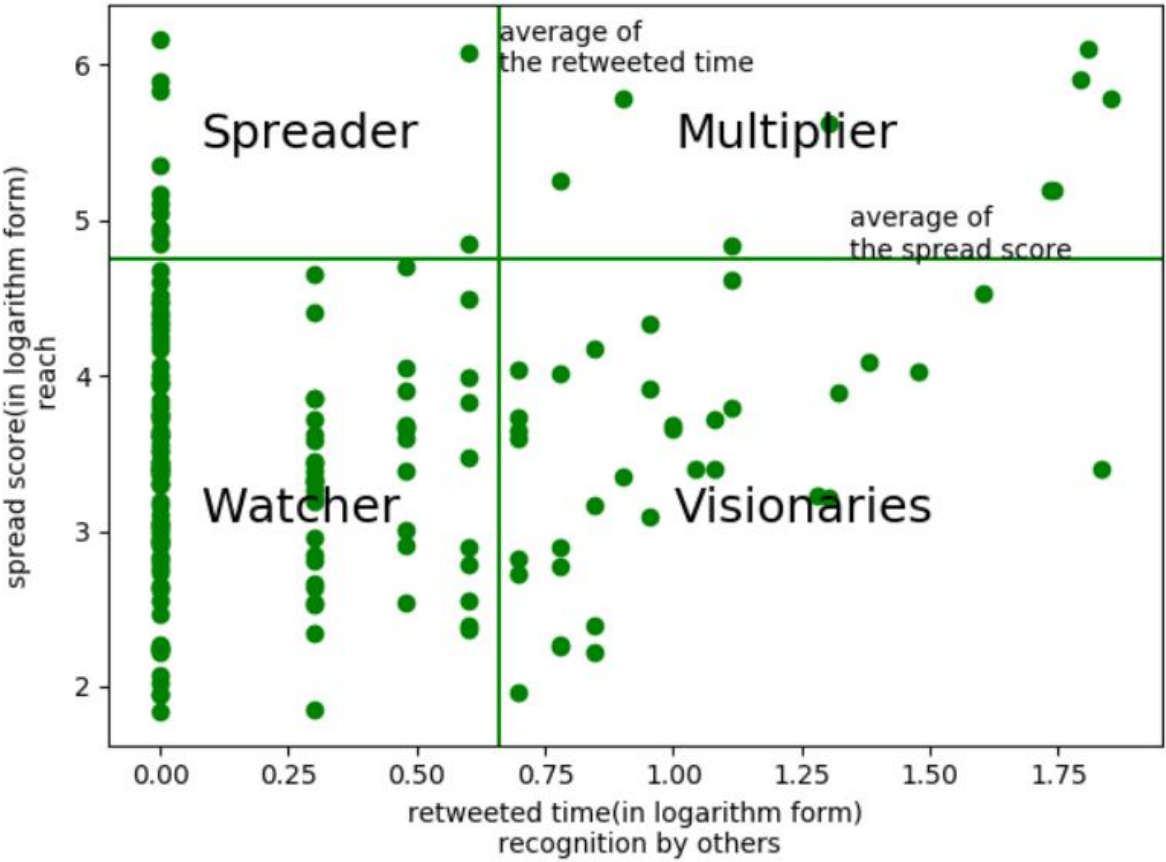


Figure 13: Example chart with the factors of the spread score and retweeted time

After referencing the research done by Gartner, we can use the way as the Gartner does in its "Magic Quadrant" researches to present the data in 4 regions divided by these 2 average lines [56].

We define the users who above both the average lines will be the multiplier with adequate influence.

For the group of users which exceed the average number of retweeted time and being mentioned can be seen as the visionaries because they have an understanding of the topic and the knowledge in that field while lacks the motion to spread others' ideas or words.

The users who are willing to retweet ideas, comments, discussion on the topic can be considered as the spreader, while it does not demonstrate that they have a vision of the topic or field.

For users who both below the average lines, they are considered as the watcher, they participate in the topic while they are not willing to generate interactions with others actively.

4.6 Sentiment analysis on tweets

Now, the multipliers are confirmed. It is of great significance to analyze their sentiment because their emotion will have an influence on their followers and the topic itself.

As mentioned above, the PatternAnalyzer in TextBlob will be applied to analyze these multipliers' tweets. In the Neo4j graph database, if a user wants to post, reply, quote, or retweets other tweets. The first step he needs to do is to post a tweet. Then different relationships will be attached to the nodes according to the situation. So the simple and properly Cypher is:

```
1. Match(u:User{username:"the name of the multiplier"})-[:POSTS]->(t1:Tweet)
2. Return t1.text
```

Then the tweets from the user in the list will be analyzed by TextBlob, the outcome will be positive, negative or neutral. Then the resulting chart will show the positive sentiments in the color blue, negative sentiments in the color yellow, and neutral sentiments in green with usernames in Figure 14:

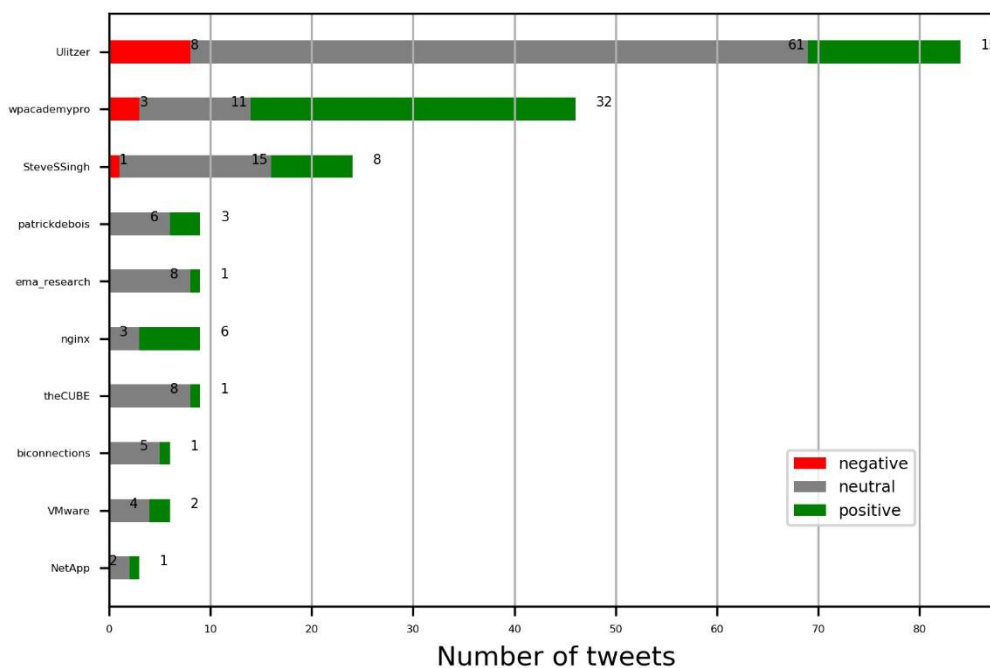


Figure 14: Example result chart of sentiment analysis

The result charts will contains both the users with the most retweeted tweets and users with the most time of being mentioned. This chart will help us to see the overall sentiments on the topic from these influencers. Also, a pie chart will be made to show the overall sentiments of all the multipliers.

5 Result of Two Use Cases

5.1 Use case 1: DockerCon 2019

According to the DockerCon’s official website, “DockerCon 2019 is a 3 day technology conference, where customers and community come to learn, share and connect with each other”. It started from April 29 to May 2 at the Moscone Center in San Francisco, California, 2019 [37].

Firstly, the related hashtags need to be found to get filtered Tweets from the Twitter API. From DockerCon’s official Twitter account’s description, we found the official hashtag is “#DockerCon”. Then we searched the website “Hashtagify” with this hashtag, and the website returned 10 related hashtags [38].

Also, we recorded 960 tweets from March 18 to March 19 which contains the official hashtag. We counted all the hashtags and sorted them from high to low.

Then Table 6 shows the comparison of the hashtags from two sources.

Table 6: Hashtag comparison of two sources I

Top10 mentioned hashtag(not case-sensitive)	
Source: https://hashtagify.me/hashtag/dockercon	Source: 960 tweets recorded from API with #DockerCon(at 19/03/2019)
#docker	#docker
#containers	#devops
#devops	#cloud
#thecube	#bigdata
#opensource	#hybridcloud
#container	#cio
#emc	#iot
#meetup	#kubernetes
#dockerselfie	#analytics
#kubemetes	#automated

As we can see from above, the hashtags appear from both sources are “#docker”, “devops”, “Kubernetes”. Taking the consideration of limited time and source to collect tweets by oneself, and compared with those gathered by a commercial website, self-collected samples may have less accuracy. From DockerCon’s website description, we can know that DockerCon is the conference focused on the container industry. The hashtag “#container” and “#containers” will also be added to the filter.

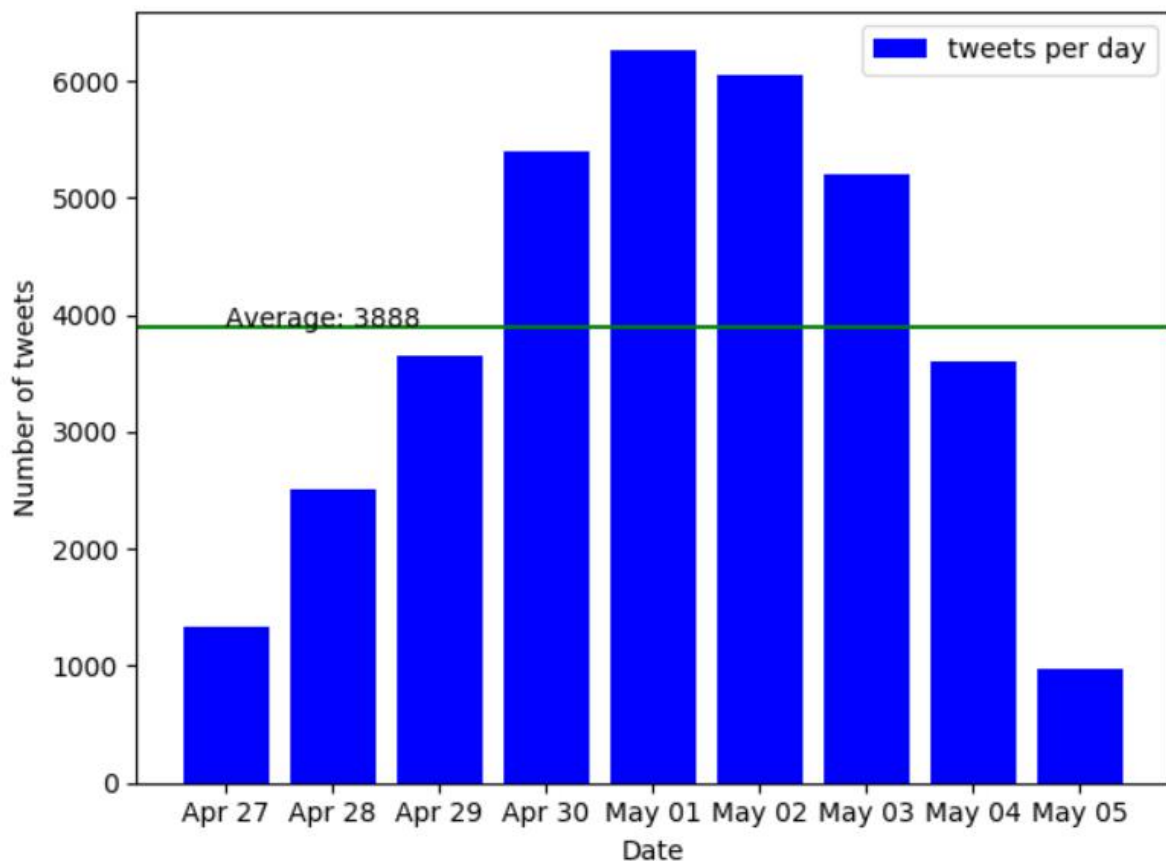
Then these hashtags can be used for recording related tweets on DockerCon.

```
1. twitterStream.filter(track=[ "#DockerCon", "#docker",  
2.                               "#devops", "#kubernetes",  
3.                               "#container", "#containers"],  
4.                               languages=["en"])
```

Now, recording tweets can start. Taking the consideration of the relevant discussion around the DockerCon had already started and lasted longer than when it closed. So, the range of recording time we chose is from 2019-04-27 to 2019-05-05 which can cover the whole meeting time(2019-04-29 to 2019-05-02).

After recording, 35000 tweets were collected. Figure 15 shows the overview activity of tweets:

Figure 15: Overview of activities I



As the chart shows, the number of tweets is increasing from the start date of the recording by degrees. And during the meeting time (04-29 to 05-02), almost the number exceeds the average line marked in blue every day. After reaching the top on 1.May, the number decreases gradually. While after the closing time, the number on 3.May is still above the average which shows the discussion on the topic remains ongoing.

Figure 18 is the chart with the factor of spread score and mention times:

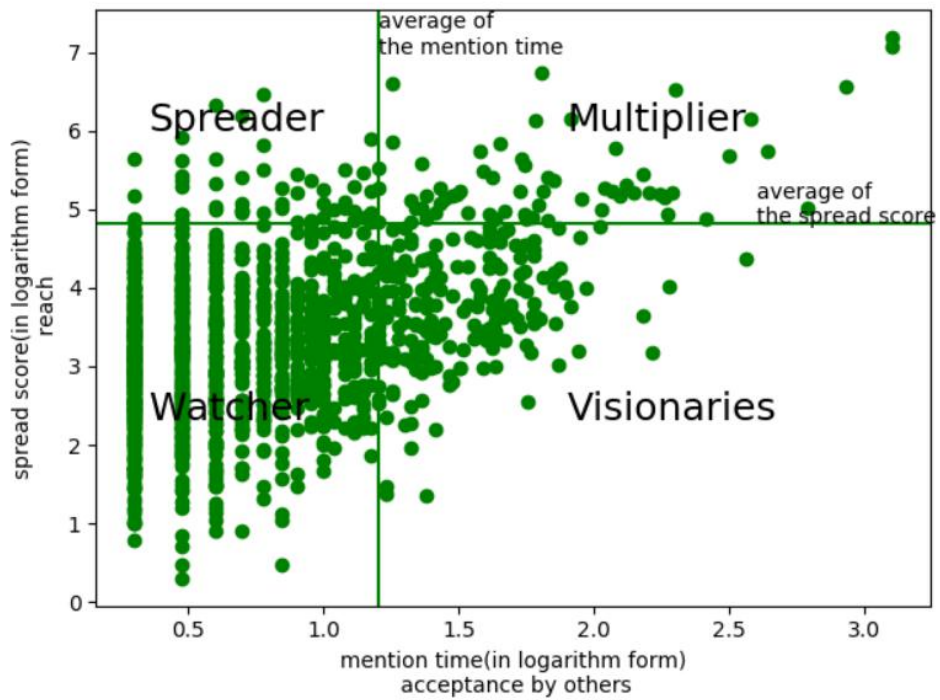
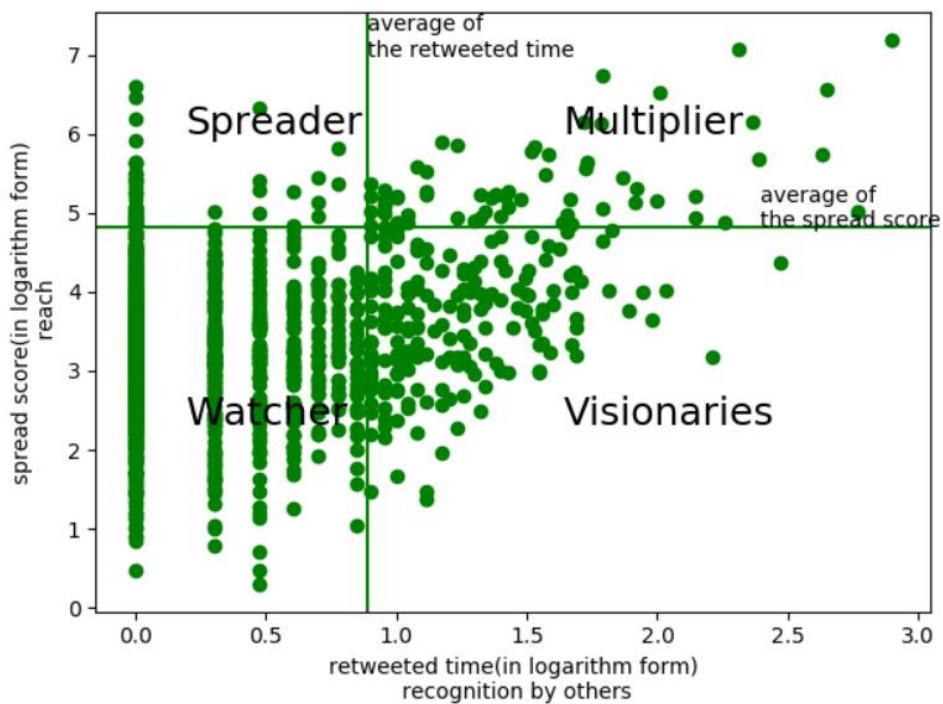


Figure 18: Overview with the factor of spread score and mention times I

Figure 19 is the chart with the factors of the spread score and retweeted time:

Figure 19: Overview with the factor of spread score and retweeted times I



Now, we can confirm the multipliers on this topic and print them in Table 7 which is sorted by spread_score:

Table 7: List of multipliers I

username	spread score	mention time	retweeted time	follower number
CloudExpo	15394313	1255	787	73657
Docker	11902135	1256	203	340061
BigDataExpo	5631161	63	61	29177
SantchiWeb	4023157	17	0	4973
DockerCon	3646448	854	447	35062
DevOpsSummit	3436722	198	101	18477
TechNative	1460232	81	51	81124
KubeSUMMIT	1441755	378	230	8955
ThingsExpo	1355528	60	60	14734
cybersecboardrm	782766	14	14	43487
evankirstel	730356	17	16	243452
holgermu	678592	44	33	42412
VMware	597436	119	32	298718
IoTJournal	565444	37	37	7964
TheHackersNews	543200	438	424	543200
ZakiaDX	481828	314	245	2956
ExpoAI	440578	53	53	5123
craigbrownphd	390040	22	11	48755
Ulitzer	366640	54	52	4583
bamitav	338013	15	12	26001
EXPOFinTech	305457	38	36	3511
thenewstack	281515	151	73	21655
CloudBees	261552	41	2	32694
nixcraft	253796	66	0	126898
CloudNativeFdn	237573	70	5	33939
chanezon	203070	130	82	9670
GoNorthStack	200012	12	8	1613
avrohmg	188864	26	26	11804
ManningBooks	185328	109	12	20592
nyike	174528	52	23	9696
IanColdwater	172504	64	23	21563
rasangarocks	171900	31	20	42975
Arm	171885	115	12	57295
SteveSSingh	167450	159	9	9850
Jadirectivestwt	167336	140	139	6436
RedHat	162439	193	25	162439
cloud66	156180	30	22	10412
kubernetesio	155017	174	9	155017
CloudJobFair	150250	29	29	3005
nginx	149994	23	8	74997
devopsdotcom	146670	123	45	29334
ajeetsraina	141818	182	98	3083

username	spread score	mention time	retweeted time	follower number
puppetize	139342	27	4	69671
SYSCONtv	134619	89	81	1951
NetApp	129005	20	8	129005
velocityconf	126035	11	8	18005
julielerman	117076	26	26	29269
wpacademypro	113622	61	61	3918
ema_research	106484	22	1	15212
dinodaizovi	103938	21	21	34646
ahmedjr_16	102628	613	583	25657
openshift	101506	106	8	50753
stefscherer	96129	46	44	2913
IBMDeveloper	94380	12	10	94380
gitlab	91610	33	0	91610
HackRead	89795	25	24	89795
Ana M Medina	88620	54	20	7385
kubernauts	88104	187	139	3671
patrickdebois	86060	40	16	21515
rustlang	83032	23	0	41516
InfoQ	78156	58	19	39078
ExpoDX	77736	42	42	948
theCUBE	76608	23	9	12768
SUSE	76600	55	10	38300
Fisher85M	75894	258	180	75894
TechJournalist	71823	66	46	71823
wendynather	68742	20	0	22914
biconnections	66946	8	8	66946

As Table 7 shows, we identified the multipliers by comparing the factors of the spread score, mention times, and retweeted times, because the user activities can be considered as a part of user influence. For example, the Top2 from the table user “Docker” has more followers than the Top1 “CloudExpo”, while it lacks the ability to spread tweets and has fewer time being retweeted by others. It is reasonable that not using the number of followers to evaluate the influence of a Twitter user. Also, the result of the average of the time of being retweeted and being mentioned are 7.7 and 16. And it shows that these multipliers have highly involved in the topic.

Now, the sentiments of these multipliers are need to be analyzed. The Pattern Analyzer ranges the polarity of the sentences from -1 to 1, the polarity above 0 means “positive”. And the polarity below 0 means “negative”. And in some cases, 0 can be the “neutral”. Taking the consideration of that the Pattern Analyzer is a pre-trained analyzer and might have difficulty in rating the polarity of “neutral” tweets to 0 exactly in the situation like users did not follow the grammar when they post tweets in English. So that it will be a more proper way to arrange a range for the sentiment “neutral”.

Firstly, Figure 20 is the overview of distribution of all the tweets:

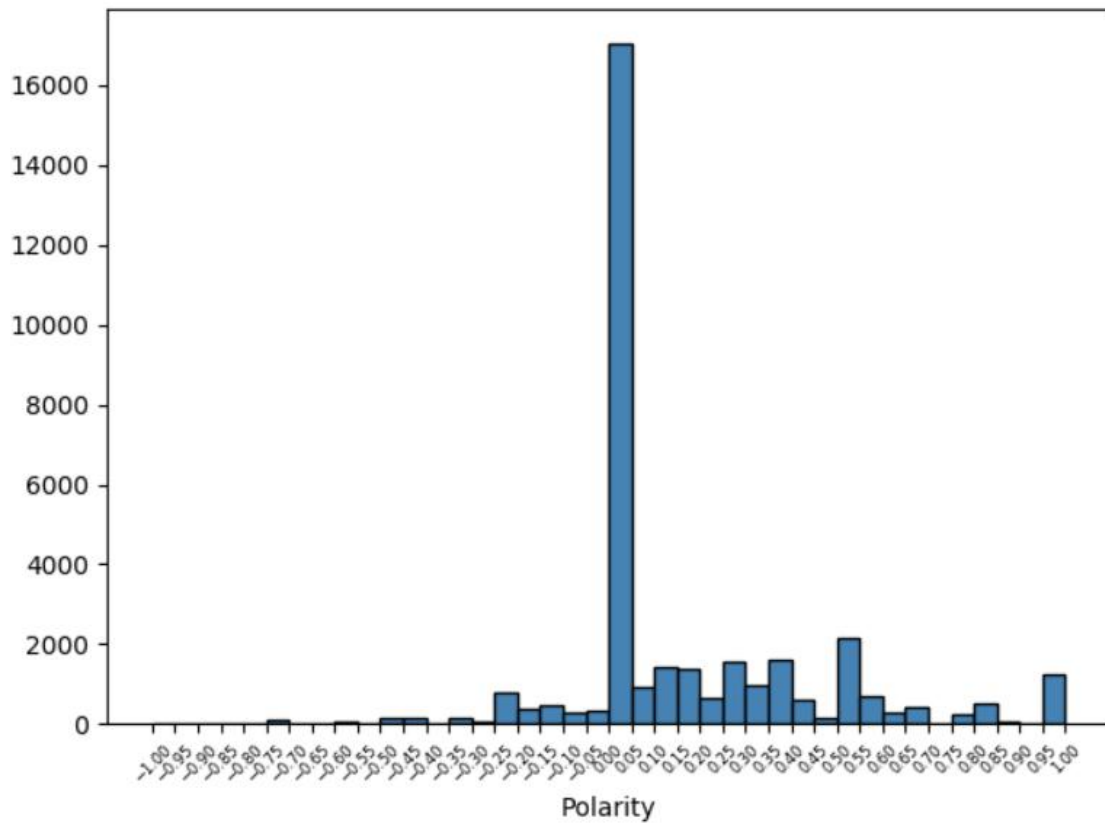


Figure 20: Overview of the distribution of the polarity I

As Figure 20 shows above, most tweets' polarities are around 0 then with some polarities above 0 and few polarities below 0. Based on this kind of distribution, we can set the range of sentiment classification as Table 8 shows:

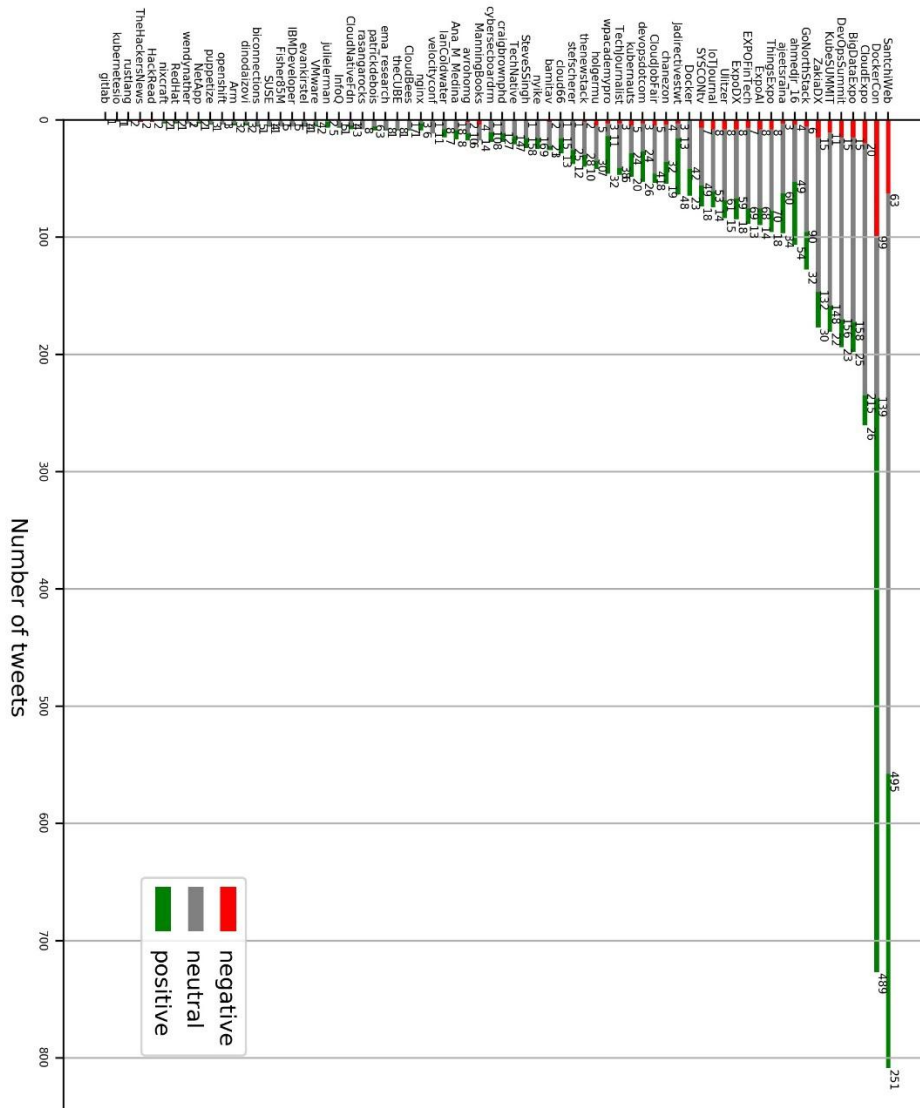
Table 8: Thresholds for the negative/neutral/positive categorization of tweets I

Positive	Neutral	Negative
$0.25 \leq \text{Polarity} \leq 1$	$0 < \text{Polarity} < 0.25$	$-1 < \text{Polarity} \leq 0$

Now, this analyzer can be applied to analyze the sentiments of these multipliers.

Then Figure 21 shows these multipliers' attitude towards the topic. The color green means positive, red means negative, and gray represents neutral. The bars in the chart will be arranged from negative over neutral to positive.

Figure 21: Multipliers's attitude towards use case I

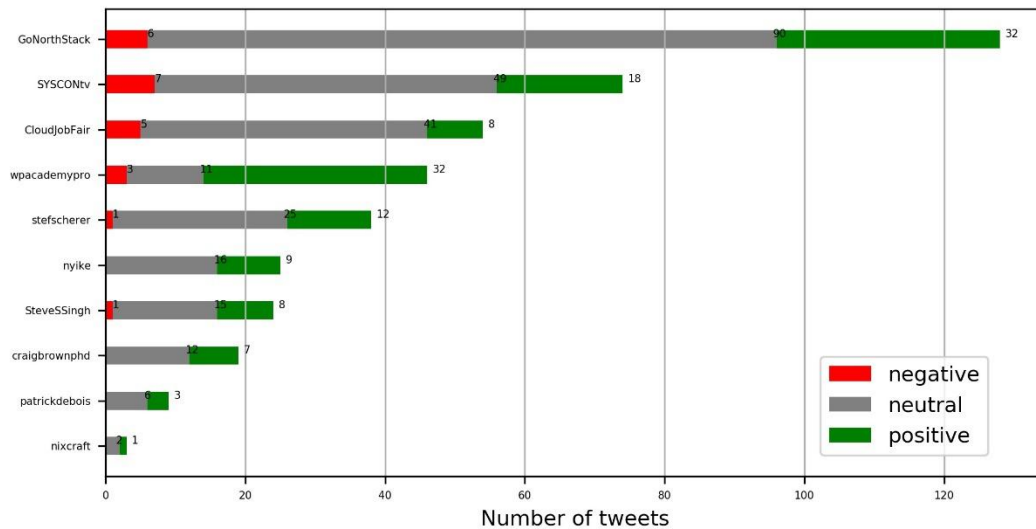


As Figure 21 shown above. During the process of representation, we found the user "DockerCon" and the user "SantchiWeb" posted too many tweets which caused other user's information cannot be presented properly and clearly. So the original chart will be separated into 2 charts. The first chart shows the results of the users excludes "DockerCon" and "SantchiWeb". Also, it will be better to present the overview of the result with part of the multipliers from the list. Because fewer rows of results will help us to see these multipliers' overall sentiments more clearly, 10 users from the user list

will be randomly chose to be presented. Then the result of the two users mentioned above was presented.

Now Figure 22 is the improved result charts, which excludes the user “DockerCon” and “SantchiWeb”:

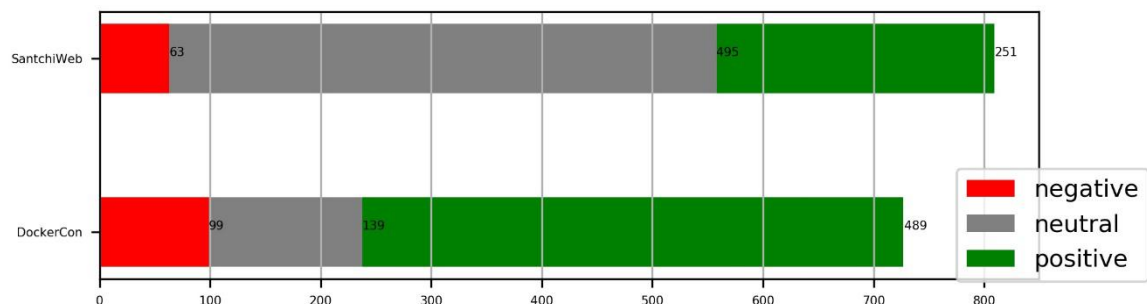
Figure 22: Result chart excluding "DokcerCon" and "SantchiWeb"



These random chose users all posted tweets with their neutral sentiment. Some users’ tweets contain all the three sentiments and positive tweets are more than the negative tweets in most cases.

Figure 23 contains the results of “SantchiWeb” and “DockerCon”:

Figure 23: Results of "DockerCon" and "SantchiWeb"



As we can see, most sentiments of the tweets posted by the user “SantchiWeb” tweets are neutral. Also the positive tweets are more than negative tweets. The user “DockerCon” posted most positive tweets and neutral tweets are more than the negative tweets, because “DockerCon” is the official Twitter account of the conference.

Finally, Figure 24 shows overall attitudes from these multipliers will be shown in a pie chart:

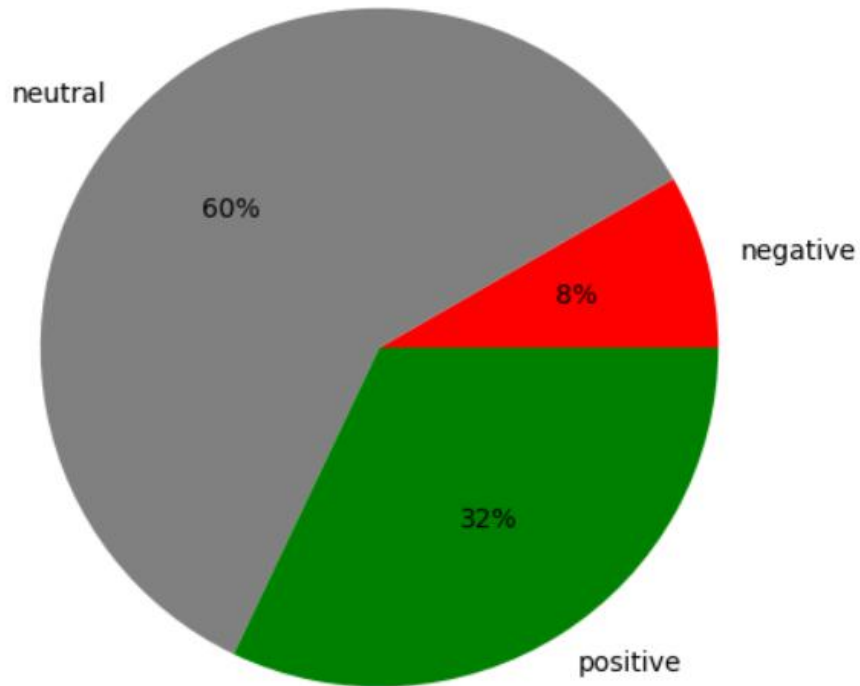


Figure 24: Overall sentiments from multipliers on the topic I

As a result, 60% of the multipliers' sentiments are "neutral", and the percentage of sentiments "positive" is 32% which is 8% higher than the sentiment "negative". This shows that most multipliers still maintain a neutral attitude when participating in this topic, and the overall sentiment is positive.

5.2 Use case 2: KubeCon+ CloudNativeCon 2019

According to the website's official introduction, the “KUBECON + CLOUDNATIVECON” is “The Cloud Native Computing Foundation’s flagship conference gathers adopters and technologists from leading open source and cloud native communities in Barcelona, Spain from May 20-23, 2019”. It will be held in Barcelona, Spain from May 20 -23,2019 [44].

Then, we also found the “official” hashtags are “#KubeCon” and “#CloudNativeCon” from its website [44]. And we use the website “hashtagify” to get two lists of related hashtags. Also, we recorded 700 tweets from May 13 to May 15 with the “official” hashtags and counted the frequency of the hashtags.

Table 9 shows the comparison of these hashtags from two sources.

Table 9: Comparison of two sources II

Top 10 mentioned hashtag(not case-sensitive)		
Source:https://hashtagify.me/ hashtag/cloudnativecon [46]	Source:https://hashtagify.me/ hashtag/kubecon [45]	Source: 700 tweets recorded with #KubeCon and #CloudNativeCon from May 13 to May 15.
#DataScience	#Serverless	#kubecon
#Serverless	#OpenShift	#cloudnativecon
#DataAnalytics	#Docker	#kubernetes
#BigData	#cloud	#cncf
#Kubernetes	#redhat	#serverless
#KubeCon	#CloudNative	#multicloud
#CloudNative	#containers	#pacha
#Cloud	#CNCF	#cloudnativenerd
#infographic	#CloudNativeCon	#istio
#fluentd	#Kubernetes	#cloudnative

The hashtags will be selected if they appear both in two sources. As a result, 6 hashtags were selected: “#kubecon”, “#cloudnativecon”, “#kubernetes”, “#cncf”, “serverless”, “cloudnative”. And we applied these hashtags to the filter.

```

1. twitterStream.filter(
2.     track=["#KubeCon", "#CloudNativeCon", "#Kubernetes", "CloudNative", "Serverless",
           "#CNCF"])

```

Also, language filter was removed because the conference was held in Europe and it is possible that some users prefer using their own language to participate in the topic rather than English. So an extra chart will be made to show the language of language by users.

Figure 25 shows the distribution of language usage.

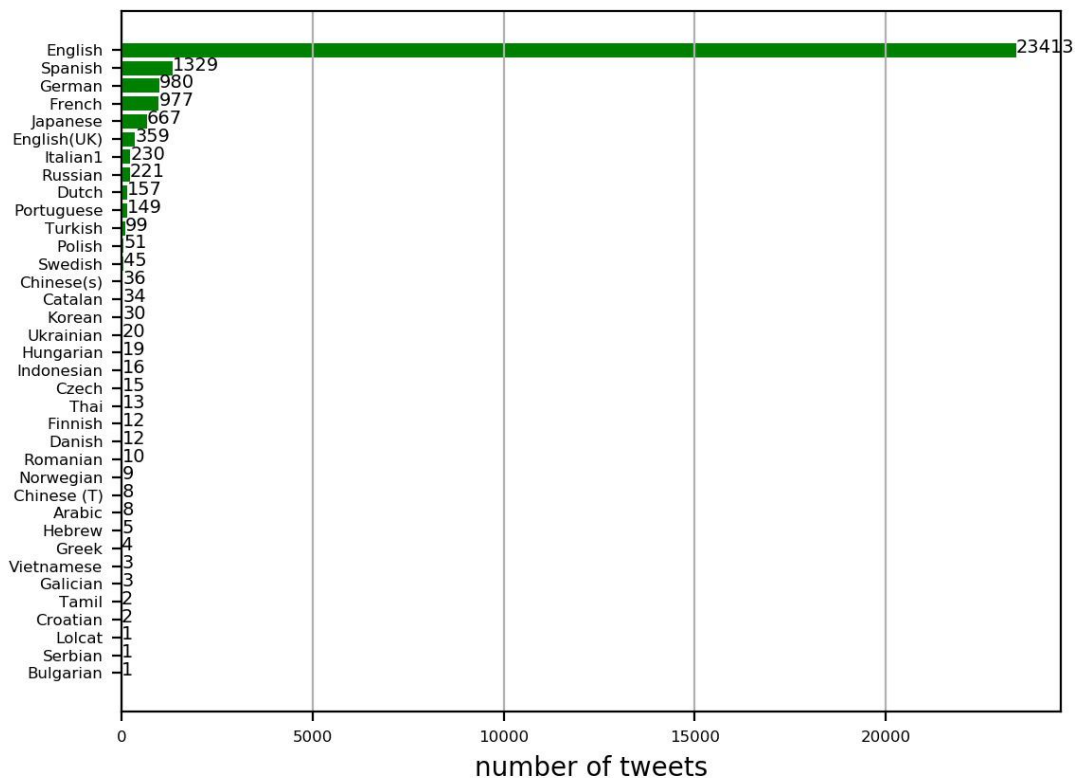


Figure 25: Language usage chart

From the Figure 25, it can be found that English is the language with the highest frequency because English is the most widely used language in the world and it is also used as the general language of the conference. The second is Spanish, it can be explained that the conference was held in Spain which has a certain impact on the frequency of Spanish. Then comes with two Europe countries because it was also held in Europe.

Figure 26 shows the activity of the tweets:

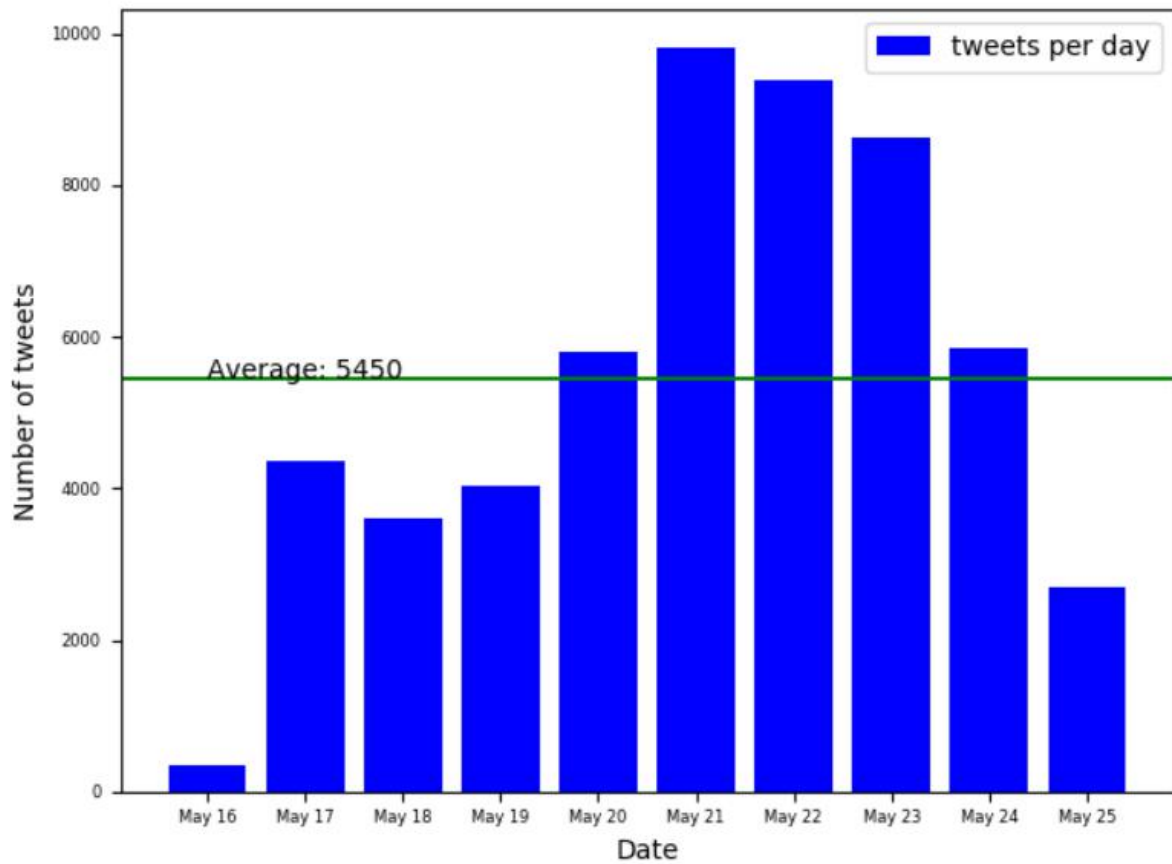


Figure 26: Activities of tweets of use case 2

As can be seen from Figure 26, the activity of the use case 2 is as same as the use case 1 as discussed in Chapter 5.1. The tweets generated during the period of the conference are all above the average. Also, the low number of tweets on May 16 because it the recording process did not cover the whole day.

Figure 28 shows the overall relationships among three factors: spread score, mention times, and retweeted times:

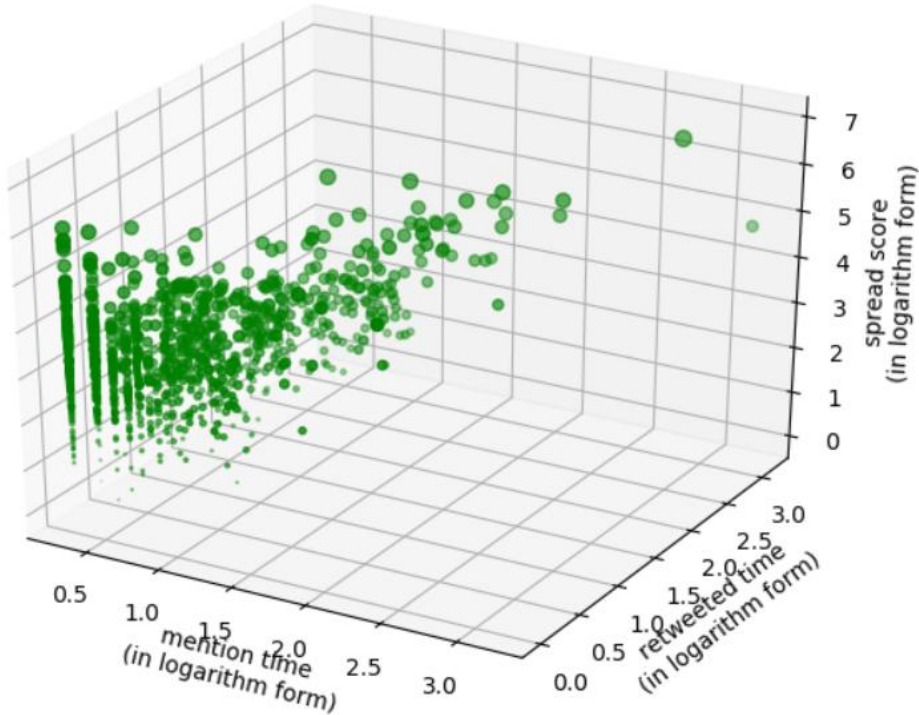


Figure 28: Overall relationships among spread score, mention times and retweeted times II

Figure 29 uses 2d scatter chart to display the relationship between spread score and mention times.

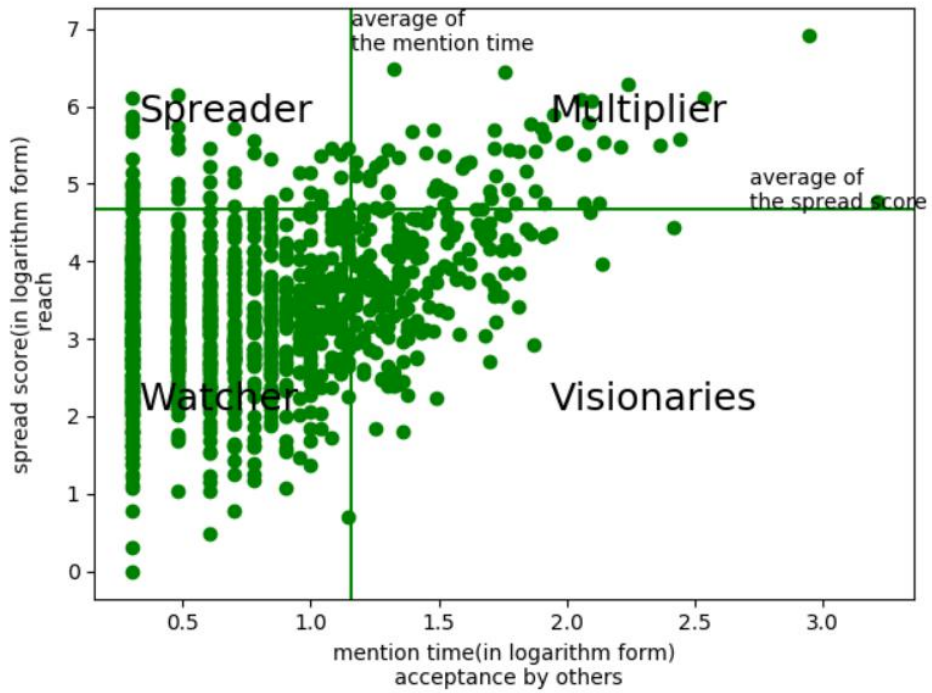


Figure 29: Overview with the factor of spread score and mention times II

Figure 30 is the chart with the factor of spread score and retweeted times.

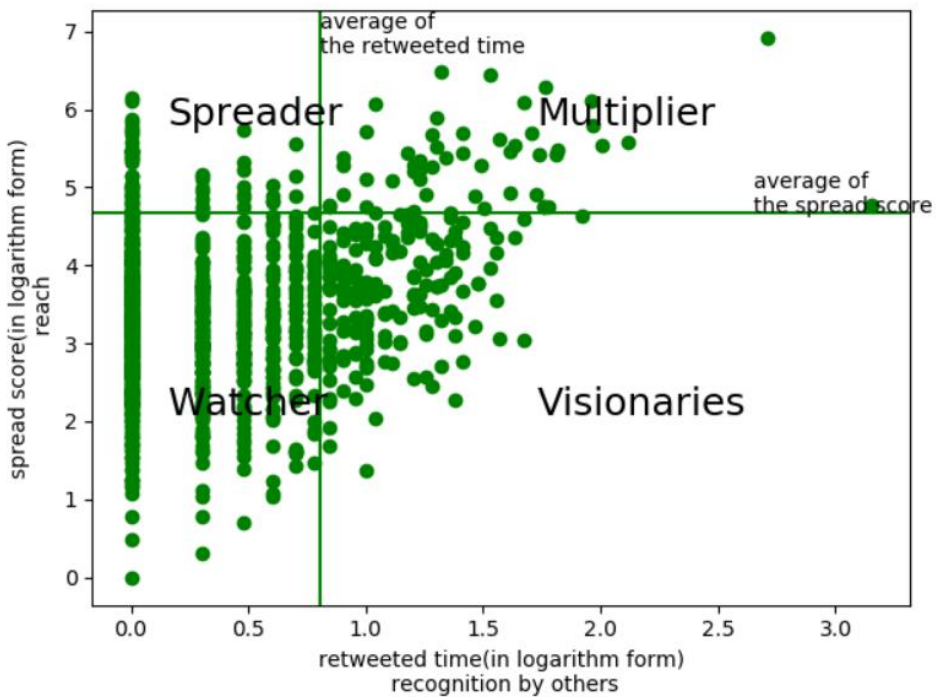


Figure 30: Overview with the factor of spread score and retweeted times II

Table 10 contains all the multipliers from the conference “KubeCon+ CloudNativeCon”:

Table 10: List of multipliers II

username	spread score	mention time	retweeted time	follower number
CloudExpo	8094020	881	511	73582
BigDataExpo	3002347	20	20	29149
ServerlessFan	2803794	56	33	1939
DevOpsSummit	1955170	174	57	18445
CloudNativeFdn	1291855	346	90	34915
digitalocean	1230362	112	46	175766
VMware	1198112	125	10	299528
GCPcloud	786519	88	19	786519
couchbase	618292	120	92	154573
KubeSUMMIT	590898	71	0	8953
openshift	513670	79	9	51367
bamitav	490846	51	50	25834
OpenAtMicrosoft	485600	29	25	60700
ThingsExpo	470304	24	18	14697
theCUBE	411456	81	36	12858
gitlab	370792	277	129	92698
QuinnyPig	351378	98	42	13014
Docker	343867	138	100	343867
RedHat	329060	96	19	164530
kubernetesio	315992	230	0	157996
thenewstack	305620	162	65	21830
GoNorthStack	293364	51	40	1686
IanColdwater	280632	19	14	23386
OracleCloud	279184	59	25	69796
APGuha	266396	74	64	5668
bridgetkromhout	262719	64	54	29191
jboner	251867	27	0	22897
jbeda	244980	116	21	24498
evankirstel	244835	12	7	244835
googlecloud	232053	32	0	232053
TechJournalist	215388	21	16	71796
ZakiaDX	192725	41	30	2965
IBMDeveloper	188950	14	7	94475
OracleDevs	187818	18	15	93909
mesosphere	187284	39	18	46821
erlp	166170	17	16	2865
Cloudflare	161110	38	15	80555
HelmPack	144404	68	1	11108
NetApp	128910	52	9	128910
cloud66	125064	16	16	10422
javascriptflx	120352	12	12	30088
makerfaire	113754	16	0	113754
tmclaughbos	95613	30	1	3297

username	spread score	mention time	retweeted time	follower number
krisnova	86056	32	0	10757
John Papa	84210	58	40	84210
ChrisShort	81991	19	17	6307
goserverless	81645	49	7	16329
alexellisuk	81256	74	52	10157
adrianco	81232	29	2	40616
ranrib	78820	19	4	5630
SUSE	77158	19	3	38579
danielbryantuk	75970	34	28	7597
ZalandoTech	65910	50	0	10985
gp_pulipaka	58976	1635	1419	58976
dankohn1	57512	133	59	4108
cloudnativeapps	56196	62	9	6244
dabit3	56180	81	56	11236
theburningmonk	55854	116	56	6206
stu	54064	32	31	13516
ahmetb	49272	17	0	12318
kubernauts	48945	19	15	3765
garethr	48024	43	22	12006
rauchg	47005	15	5	47005
openfaas	46376	46	10	5797

As we can see, the number of followers contributes less to these multipliers' high score of spreading. And it is interesting that the user "CloudExpo" ranks top 1 in both two use cases.

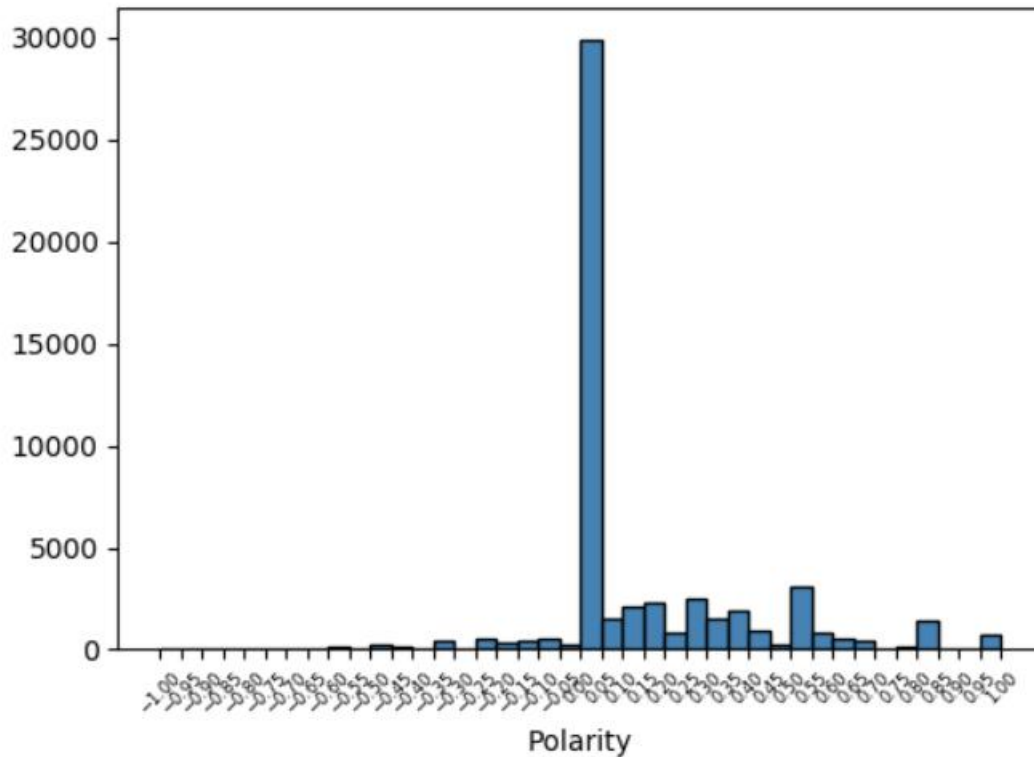


Figure 31: Overview of the distribution of the polarity II

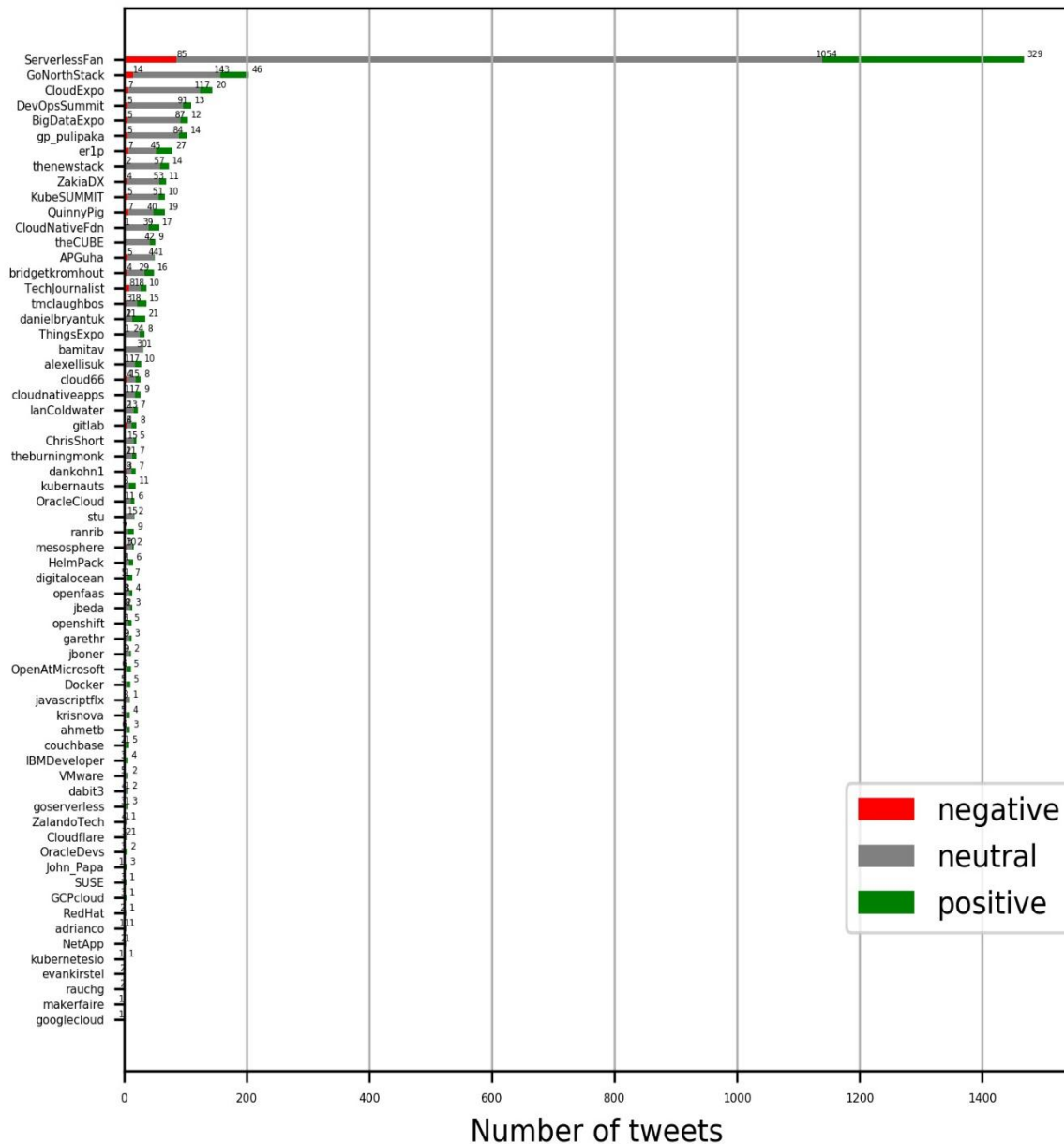
As Figure 31 shows above, the distribution of the polarity is also like the results of the use case 1 (DockerCon). Most results are around 0 then some results above 0 and few results below 0. So the setting of thresholds for the negative/neutral/positive categorization can be still used as the setting used in the use case 1. As Table 11 shows:

Table 11: Thresholds for the negative/neutral/positive categorization of tweets II

Positive	Neutral	Negative
$0.25 \leq \text{Polarity} \leq 1$	$0 < \text{Polarity} < 0.25$	$-1 < \text{Polarity} \leq 0$

Figure 32 shows all the multipliers' attitude:

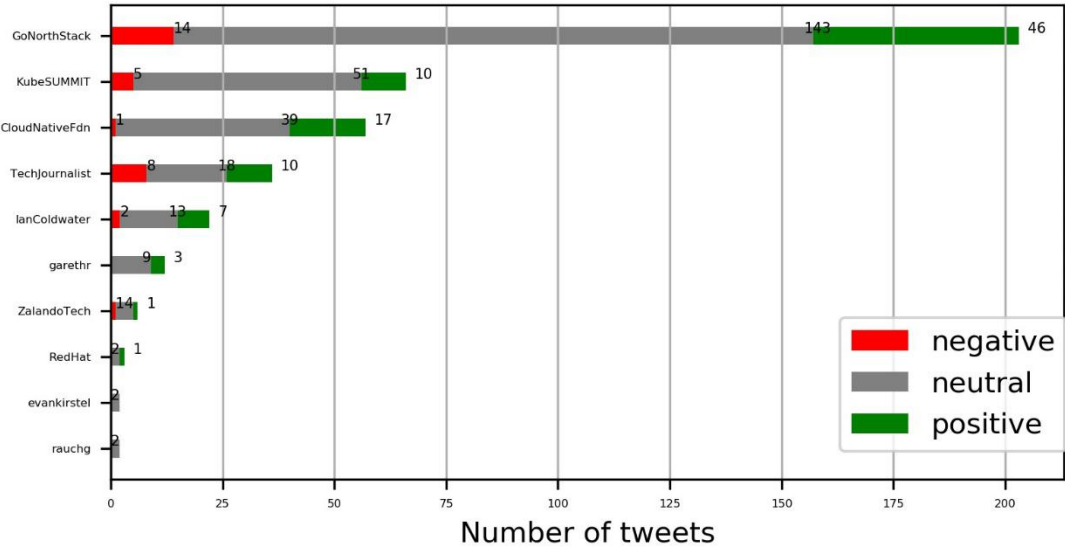
Figure 32: Multipliers' attitude towards use case II



In this case, we found the user “ServerlessFan” posted the most tweets and we will separate the overall chart into 2 sub charts like what we did in use case 1.

The first chart is Figure 33 will show random chosen 10 users excludes the user “ServerlessFan”:

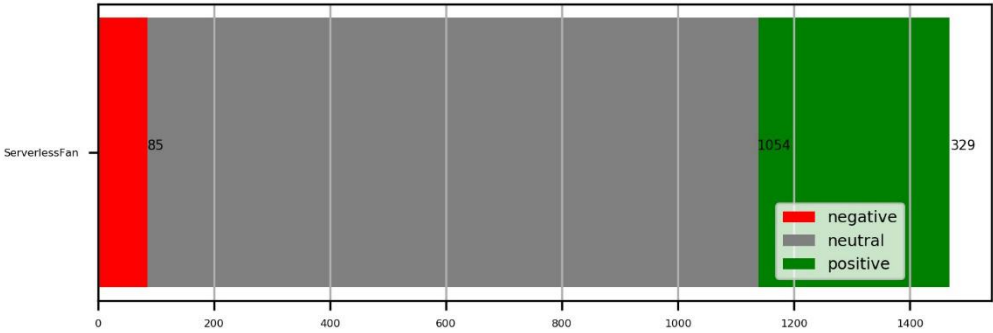
Figure 33: Result chart without “ServerlessFan”



These random chosen users all had neutral tweets and some users had both positive and negative sentiments, while the positive tweets are more than negative tweets.

Figure 34 shows the result of the user “ServerlessFan”:

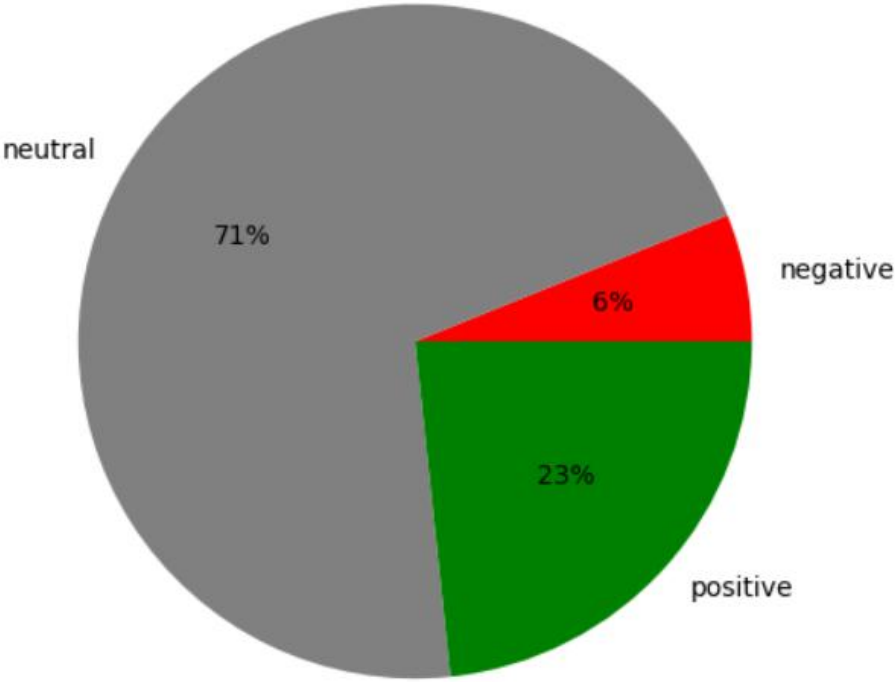
Figure 34:Result chart of “ServerlessFan”



The user “ServerlessFan” also posted most neutral tweets and had more positive tweets than negative ones.

Finally, the pie chart Figure 35 was used to present the overall attitudes from these multipliers:

Figure 35: Overall sentiments from multipliers on the topic II



Users of the “KubeCon+CloudNativeCon” still remained neutral attitudes towards this conference, the percentage is 71%. And the percentage of positive attitudes are 17% higher than negative ones. So the overall sentiment is positive.

6 Threats on validity and technical misuse

6.1 The limitation to the number of tweets recorded from Twitter Streaming API

The standard API rate limits the number of available tweets for a period of time, that means not all the tweets about the topic can be recorded locally. 1% of all public tweets were offered freely by the sample stream [39]. And a costly 10% sample of all Tweets can be retrieved from the Decahose stream [40]. So the results based on the recorded tweets cannot represent the whole Twitter network. While according to the research conducted by “Pfeffer, Mayer, and Morstatter” in 2018, Twitter's sampling mechanism will cause the sample tweets to be influenced deliberately. Using automated bots or accounts and sending tweets at a specific time period, which makes tweets be more likely to appear in the Simple API [41]. The validity of the tweets will have the risk of being manipulated and damaged.

6.2 Limitations due to Twitter User Protection Terms and Ethical Considerations

For the sake of protecting users' privacy and safety, Twitter offers the user an option to protect their accounts. If a user agreed to protect his tweets, only his followers can have access to his tweets and these tweets will no longer be public on Twitter or appear in public Twitter search results [42]. So these protected tweets will not be recorded, even these protected discussions towards the topic may have great influence.

6.3 Limitation of the accuracy of sentiment analysis

TextBlob is mainly designed for analyzing sentences written in English so tweets in other languages will be excluded during the process of the sentiment analysis. Classifying all the tweets and using other language processing toolkits will be a solution while it takes extra time and efforts. Also, the Pattern Analyzer from TextBlob is a pre-trained analyzer so that the performance will not be so stable when analyzing different text sources. It can be solved by training analyzer with the text source before analyzing. Finally, some users may not follow the normal grammar to write tweets, it will also cause the performance of the accuracy decrease.

6.4 Limitation of the evaluation of the influence

In this thesis, we only considered the user's ability to spread tweets and topics and evaluate their influence based on the time of their names being mentioned and the tweets being retweeted. It proves that measuring user's influence by their followers is not advisable. While other factors can also be considered to measure influence, like the interactions among the users and the activity of the accounts.

6.5 Technical misuse

On the one side, as a new way for government departments to understand public opinion and for enterprise to understand the review on their products or service, Social network analysis is becoming a trend. Fundamentally speaking, its function is to timely capture the text, data, pictures, and video information from social media and process these data with their own analysis system. The views behind the data that represent the opinions of the majority of people were summarized, so as to understand the public opinion to find and solve the problem. Like Alzheimer's Association using Twitter to generate mass awareness and concern about Alzheimer's disease from public [52], or Hswen, Y., Naslund, J., Brownstein, J. and Hawkins, J. propose to use social media to detect, respond and prevent suicide by monitoring people with serious mental illnesses and their discussion [53]. These examples provide many promising opportunities for many industries if this technology was properly used.

On the other side, this technology will risk invading others' privacy or safety and controlling freedom of speech. When the negative information was found, rather than finding out whether the fact is true and understand the public's attitudes and comments on these negative issues, but thinking about how to delete these negative remarks, or through administrative means to suppress the posters, control the media's report dissemination and so on. As the American Civil Liberties Union(ALCU) claimed that Geofeedia gained the data access right to Twitter, Facebook, and Instagram to monitor activists and protesters [54]. According to a report from The Computational Propaganda Project, in Iran, Malaysia, Russia, Saudi Arabia and Tanzania, instances of abuse of using technology to combat dissidents, minorities and human rights defenders have happened [55]. At this time, this technology becomes a tool of public opinion control.

7 Conclusion & Outlook

7.1 Conclusion

This thesis uses Twitter as the data source, 35000 English tweets for use case 1 and 23743 tweets written in English for use case 2 were collected. Then we tried to use database Neo4j to store and search the information and relationships from tweets, and natural language processor TextBlob was applied to analyze the sentiments. In order to finish the task of finding the “hot topics” and the multipliers. The former we used the word cloud to present the “hot topic” with highly frequency appearing words from the tweets, and the later we identified the multiplier with the factors of their ability to spreading the topic and their own influence.

As a result, we found the selected multipliers have a high activity of participating in the topic and the number of the followers cannot be the only standard to identify the users’ influence. And the word cloud shows the hot topics are these proper nouns, companies or services related to the conference. To the sentiment part, most of the sentiments remain neutral and the positive sentiments are higher than negative sentiments.

This thesis explores and implements a way to find “hot topics” and “multipliers”. Hot topics will help users to know the keywords of the topic, it will help them to get the overview of the whole topic then make the decision to participate in or not depends on the main contents of the topic. Then the method of finding multipliers can be applied for companies or social media for searching for valuable users to promote their products or service on specific topics to research the target groups. For users, this method can help them to locate and follow users with knowledge in some area to get their opinion and related updates of the topic.

7.2 Outlook

As the most popular information exchanging platform, Twitter contains a large amount of user information, which has great commercial and research value. Future improvement will be made on the sentiments analysis part, we used pre-trained analyzer in this thesis because of the time limitation. Using machine learning especially deep learning to train the analyzer will generate more precise results of sentiment analysis. Also, taking the relationships of users and users into consideration will make the process of user influence evaluation more comprehensive. To the graph database part, Neo4j is a powerful database which has the potential to be used to explore more on the interactions of users and tweets.

8 Acknowledgements

At this part of the thesis, I want to extend my sincere gratitude to my supervisor Nane Kratzke. Professor Kratzke is a considerable and responsible person who gave me detailed advice. Then, my classmates and some friends in China provided me with their comments on my thesis. So I want to express my thanks to them all for taking the time to read my thesis and not giving up reading when getting bored. Also, thanks for the regular tutorials on writing thesis which offered by Lenka Kleinau.

Honestly, so without their help, it will be hard for me to finish my thesis in time. Thanks you!

Bibliography

- [1] About.twitter.com. (2019). About. [online] Available at: <https://about.twitter.com/> [Accessed 10 May 2019].
- [2] Sarlan, A., Nadam, C., & Basri, S.B. (2014). Twitter sentiment analysis. Proceedings of the 6th International Conference on Information Technology and Multimedia, 212-216.
- [3] MacCartney, B. (2014). Understanding Natural Language Understanding. [image] Available at: <https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf> [Accessed 24 Apr. 2019].
- [4] Wasserman, Stanley; Faust, Katherine (1994). "Social Network Analysis in the Social and Behavioral Sciences". *Social Network Analysis: Methods and Applications*. Cambridge University Press. pp. 1–27. ISBN 9780521387071.
- [5] D. Munezero, M., Montero, C. S., Sutinen, E., & Pajunen, J. (2014). Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text. *IEEE Transactions on Affective Computing*, 5(2), 101–111. doi:10.1109/taffc.2014.2317187
- [6] A.H.Huang, D.C. Yen, & X. Zhang, “Exploring the effects of emoticons,” *Information & Management*, 45(7), 466–473, 2008.
- [7] Education, M.G.H. (2003). *Glencoe Computer Connections: Projects and Applications, Student Edition*. McGraw-Hill Education. ISBN 978-0-07-861399-9. Retrieved August 11, 2018.
- [8] Y. Zhou, and Y. Fan, “A Sociolinguistic Study of American Slang,” *Theory and Practice in Language Studies*, 3(12), 2209– As humans often turn to emoticons to properly express what they cannot put into words2213, 2013. doi:10.4304/tpls.3.12.2209-2213
- [9] Pak, Alexander & Paroubek, Patrick. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of LREC*. 10.
- [10] Otte, E. and Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6), pp.441-453.
- [11] W3schools.com. (2019). Introduction to Python. [online] Available at: https://www.w3schools.com/python/python_intro.asp [Accessed 30 Apr. 2019].
- [12] Insights.stackoverflow.com. (2019). Stack Overflow. [online] Available at: <https://insights.stackoverflow.com/survey/> [Accessed 5 May 2019].
- [13] Hoyer, S., & Hamman, J. (2017). xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, 5(1), 10. DOI: <http://doi.org/10.5334/jors.148>
- [14] Numpy.org. (2019). NumPy — NumPy. [online] Available at: <http://www.numpy.org/> [Accessed 30 Apr. 2019].
- [15] Bobriakov, I. (2019). Comparison of Top 6 Python NLP Libraries. [online] Medium. Available at: <https://medium.com/activewizards-machine-learning-company/comparison-of-top-6-python-nlp-libraries-c4ce160237eb> [Accessed 12 May 2019].

- [16] Nltk.org. (2019). Natural Language Toolkit — NLTK 3.4.1 documentation. [online] Available at: <https://www.nltk.org/> [Accessed 12 May 2019].
- [17] Textblob.readthedocs.io. (2019). TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation. [online] Available at: <https://textblob.readthedocs.io/en/dev/> [Accessed 12 May 2019].
- [18] Textblob.readthedocs.io. (2019). Tutorial: Quickstart — TextBlob 0.15.2 documentation. [online] Available at: <https://textblob.readthedocs.io/en/dev/quickstart.html#create-a-textblob> [Accessed 5 May 2019].
- [19] Textblob.readthedocs.io. (2019). Advanced Usage: Overriding Models and the Blobber Class — TextBlob 0.15.2 documentation. [online] Available at: https://textblob.readthedocs.io/en/dev/advanced_usage.html [Accessed 5 May 2019].
- [20] Textblob.readthedocs.io. (2019). Tutorial: Building a Text Classification System — TextBlob 0.15.2 documentation. [online] Available at: <https://textblob.readthedocs.io/en/dev/classifiers.html> [Accessed 5 May 2019].
- [21] Robinson, I., Webber, J. and Eifrem, E. (2015). Graph databases. Sebastopol, CA: O'Reilly, pp.xi,1,11-19.
- [22] Db-engines.com. (2019). DB-Engines Ranking per database model category. [online] Available at: https://db-engines.com/en/ranking_categories [Accessed 10 May. 2019].
- [23] Db-engines.com. (2019). historical trend of graph DBMS popularity. [online] Available at: https://db-engines.com/en/ranking_trend/graph+dbms [Accessed 9 May. 2019].
- [24] Db-engines.com. (2019). ArangoDB vs. Neo4j vs. OrientDB Comparison. [online] Available at: <https://db-engines.com/en/system/ArangoDB%3BNeo4j%3BOrientDB> [Accessed 12 May 2019].
- [25] Neo4j Graph Database Platform. (2019). What Is a Graph Database and Property Graph | Neo4j. [online] Available at: https://neo4j.com/developer/graph-database/#_what_is_neo4j [Accessed 12 May 2019].
- [26] Google Trends. (2019). Google Trends. [online] Available at: <https://trends.google.com/trends/explore?q=Neo4j,ArangoDB,orientDB> [Accessed 10 May 2019]. Db-engines.com. (2019). ArangoDB vs. Neo4j vs. OrientDB Comparison. [online] Available at: <https://db-engines.com/en/system/ArangoDB%3BNeo4j%3BOrientDB> [Accessed 3 Apr. 2019].
- [27] Neo4j Graph Database Platform. (2019). Walmart - Neo4j Graph Database Platform. [online] Available at: <https://neo4j.com/case-studies/walmart/> [Accessed 12 May 2019].

- [28] Docs.tweepy.org. (2019). Streaming With Tweepy — tweepy 3.3.0 documentation. [online] Available at: http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html [Accessed 3 May 2019].
- [29] Help.twitter.com. (2019). How to use hashtags. [online] Available at: <https://help.twitter.com/en/using-twitter/how-to-use-hashtags> [Accessed 3 Apr. 2019].
- [30] Williams, H., McMurray, J., Kurz, T. and Hugo Lambert, F. (2015). Network analysis reveals open forums and echo chambers in social media discussions of climate change. *Global Environmental Change*, 32, pp.126-138.
- [31] GitHub. (2019). amueller/word_cloud. [online] Available at: https://github.com/amueller/word_cloud [Accessed 5 May 2019].
- [32] Hsu, S. (2019). Introduction to Data Science: Custom Twitter Word Clouds. [online] Medium. Available at: <https://medium.com/@shsu14/introduction-to-data-science-custom-twitter-word-cloud-s-704ec5538f46> [Accessed 10 May 2019].
- [33] Rajaraman, A., & Ullman, J. (2011). Data Mining. In *Mining of Massive Datasets* (pp. 1-17). Cambridge: Cambridge University Press. doi:10.1017/CBO9781139058452.002
- [34] Developer.twitter.com. (2019). Introduction to Tweet JSON. [online] Available at: <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.html> [Accessed 10 May 2019].
- [35] GitHub. (2019). nicolewhite/neo4j-jupyter. [online] GitHub. Available at: <https://github.com/nicolewhite/neo4j-jupyter/blob/master/scripts/twitter.py> [Accessed 10 Mar. 2019].
- [36] Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P. K. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. *ICWSM*, 10, 10-17.
- [37] DockerCon San Francisco 2019. (2019). About - DockerCon San Francisco 2019. [online] Available at: <https://www.docker.com/dockercon/about/> [Accessed 22 May 2019].
- [38] Hashtagify. (2019). #dockercon: Popularity, Trend, Related Hashtags | Hashtagify. [online] Available at: <https://hashtagify.me/hashtag/dockercon> [Accessed 19 Mar. 2019].
- [39] F. Morstatter, J. Pfeffer, H. Liu, and K. Carley. Is the Sample Good Enough? Comparing Data from Twitter’s Streaming API with Twitter’s Firehose. In *ICWSM*, 2013.
- [40] Developer.twitter.com. (2019). Sample. [online] Available at: <https://developer.twitter.com/en/products/tweets/sample.html> [Accessed 24 Apr. 2019].
- [41] Pfeffer, J., Mayer, K. and Morstatter, F. (2018). Tampering with Twitter’s Sample API. *EPJ Data Science*, 7(1).

- [42] Help.twitter.com. (2019). About public and protected Tweets. [online] Available at: <https://help.twitter.com/en/safety-and-security/public-and-protected-tweets> [Accessed 13 May 2019].
- [43] Microsoft.com. (2019). [online] Available at: https://www.microsoft.com/en-us/Useterms/OEM/Windows/10/Useterms_OEM_Windows_10_English.htm [Accessed 14 May 2019].
- [44] Linux Foundation Events. (2019). KubeCon + CloudNativeCon Europe 2019 - Linux Foundation Events. [online] Available at: <https://events.linuxfoundation.org/events/kubecon-cloudnativecon-europe-2019/> [Accessed 7 May 2019].
- [45] Hashtagify. (2019). #kubecon: Popularity, Trend, Related Hashtags | Hashtagify. [online] Available at: <https://hashtagify.me/hashtag/kubecon> [Accessed 16 May 2019].
- [46] Hashtagify. (2019). #cloudnativecon: Popularity, Trend, Related Hashtags | Hashtagify. [online] Available at: <https://hashtagify.me/hashtag/cloudnativecon> [Accessed 16 May 2019].
- [47] Json.org. (2019). JSON. [online] Available at: <https://www.json.org/> [Accessed 23 May 2019].
- [48] Spacy.io. (2019). [online] Available at: <https://spacy.io/> [Accessed 23 May 2019].
- [49] Scikit-learn.org. (2019). scikit-learn: machine learning in Python — scikit-learn 0.21.1 documentation. [online] Available at: <https://scikit-learn.org/stable/#> [Accessed 23 May 2019].
- [50] Radimrehurek.com. (2019). gensim: topic modelling for humans. [online] Available at: <https://radimrehurek.com/gensim/> [Accessed 23 May 2019].
- [51] PyPI. (2019). polyglot. [online] Available at: <https://pypi.org/project/polyglot/> [Accessed 23 May 2019].
- [52] Marketing.twitter.com. (2019). Alzheimer’s Association uses Twitter’s Promoted Trends to raise awareness. [online] Available at: <https://marketing.twitter.com/na/en/success-stories/alzheimers-association-uses-twitter-promoted-trends.html> [Accessed 16 May 2019].
- [53] Hswen, Y., Naslund, J., Brownstein, J. and Hawkins, J. (2018). Monitoring Online Discussions About Suicide Among Twitter Users With Schizophrenia: Exploratory Study. *JMIR Mental Health*, 5(4), p.e11483.
- [54] ACLU of Northern CA. (2019). Facebook, Instagram, and Twitter Provided Data Access for a Surveillance Product Marketed to Target Activists of Color. [online] Available at: <https://www.aclunc.org/blog/facebook-instagram-and-twitter-provided-data-access-surveillance-product-marketed-target> [Accessed 16 May 2019].

[55] Blogs.oii.ox.ac.uk. (2019). [online] Available at: <http://blogs.oii.ox.ac.uk/politicalbots/wp-content/uploads/sites/89/2017/06/Casestudies-ExecutiveSummary.pdf> [Accessed 16 May 2019].

[56] Gartner. (2019). Magic Quadrant Research Methodology. [online] Available at: <https://www.gartner.com/en/research/methodologies/magic-quadrants-research> [Accessed 17 May 2019].

[57] Py2neo.org. (2019). The Py2neo v4 Handbook — The Py2neo v4 Handbook. [online] Available at: <https://py2neo.org/v4/index.html> [Accessed 20 May 2019]

Appendix

Appendix A – List of Figures

Figure 1 : Comparison between NLP and NLU [3].....	7
Figure 2 : Example of nodes and relationships on Twitter.....	9
Figure 3 : Example result of TextBlob API.....	11
Figure 4 : Example result of NaiveBayesAnalyzer.....	12
Figure 5 : Example result of NaiveBayesClassifier.....	12
Figure 6 : Trend of database [22].....	15
Figure 7 : Example of the model friends and friends-of-friends in a relational database[21]..	16
Figure 8 : Ranking of the graph database [23].....	17
Figure 9 : Example of a word cloud[21].....	21
Figure 10 : Structure of Tweet’s JSON.....	23
Figure 11 : Overview of the part of data from Neo4j.....	24
Figure 12 : Example 3d scatter chart with the factors of spread score, mention times and retweeted times.....	26
Figure 13 : Example chart with the factors of the spread score and retweeted time.....	27
Figure 14 : Example result chart of sentiment analysis.....	28
Figure 15 : Overview of activities I.....	30
Figure 16 : Word cloud of the topic I.....	31

Figure 17 : Overall relationships among spread score, mention times and retweeted times I..	31
Figure 18 : Overview with the factor of spread score and mention times I.....	32
Figure 19 : Overview with the factor of spread score and retweeted times I.....	32
Figure 20 : Overview of the distribution of the polarity I.....	35
Figure 21 : Multipliers's attitude towards use case I.....	36
Figure 22 : Result chart excluding "DokcerCon" and "SantchiWeb".....	37
Figure 23 : Results of "DockerCon" and "SantchiWeb".....	37
Figure 24 : Overall sentiments from multipliers on the topic I.....	38
Figure 25 : Language usage chart.....	40
Figure 26 : Activities of tweets of use case 2.....	41
Figure 27 : Word cloud of the topic II.....	42
Figure 28 : Overall relationships among spread score, mention times and retweeted times II.	43
Figure 29 : Overview with the factor of spread score and mention times II.....	44
Figure 30 : Overview with the factor of spread score and retweeted times II.....	44
Figure 31 : Overview of the distribution of the polarity II.....	47
Figure 32 : Multipliers' attitude towards use case II.....	48
Figure 33 : Result chart without “ServerlessFan”.....	49
Figure 34 :Result chart of “ServerlessFan”.....	49
Figure 35 : Overall sentiments from multipliers on the topic II.....	50

Appendix B – List of Tables

Table 1: Description of two datasets.....	12
Table 2: Example tweets from dataset1.....	13
Table 3: Example tweets from dataset2.....	14
Table 4: Result set of accuracy.....	14

Table 5: Comparison of three most top-ranked open source graph databases [24].....	17
Table 6: Hashtag comparison of two sources I.....	29
Table 7: List of multipliers I.....	33
Table 8: Thresholds for the negative/neutral/positive categorization of tweets I.....	35
Table 9: Comparison of two sources II.....	39
Table 10: List of multipliers II.....	45
Table 11: Thresholds for the negative/neutral/positive categorization of tweets II.....	47

Appendix C – List of Equations

Equation 1: Calculation of spreading score.....	25
---	----