Software for monitoring current trends on Twitter

Kapantai Eleni
SID: 3305160005

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in E-Business and Digital Marketing

January 2018
THESSALONIKI – GREECE
Software for monitoring current trends on Twitter

Kapantai Eleni

SID: 3305160005

Supervisor: Prof. Christos Moridis
Supervising Committee Members: Assoc. Prof. Name Surname
                        Assist. Prof. Name Surname

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of

*Master of Science (MSc) in E-Business and Digital Marketing*

January 2018
THESSALONIKI – GREECE
Abstract

This dissertation was written as a part of the MSc in E-Business and Digital Marketing at the International Hellenic University by student Kapantai Eleni under the supervision of Prof. Ch. Moridis.

Data lied in Social Networks can provide useful information for research purposes, along with valuable knowledge concerning user’s behavior. Context (e.g. preferences, opinions, intent, sentiment, activities) provided by social data cannot be reached by traditional research methods, helping to understand and interpret Social Media traffic on a more holistic level. By this master thesis we intend to develop a system that uses Twitter to understand how people are feeling about a topic that we choose. Primarily, we scroll the Twitter and gather information from the well-known platform on given queries. On the way forward, our efforts focused on the classification of those messages with respect to their sentiment, applying data mining techniques. The ideal of this work lies in the recognition of current trends on Twitter through the extraction of high-valued information dealing with any potential challenge that may arise.

At this point I would like to thank my supervisor for giving me the opportunity to work with him on a very demanding but interesting project.

I would also like to thank my family for supporting me throughout this process and my friend Kyriacos for giving me some extremely useful technical advice.

Kapantai Eleni

02/01/2018
Contents

Abstract ........................................................................................................................................... 3

Chapter 1: Introduction .................................................................................................................. 5
  1.1 Overview .................................................................................................................................. 5
  1.2 Problem Statement and Objectives ......................................................................................... 7
  1.3 Twitter ..................................................................................................................................... 7
  1.4 Sentiment Analysis ................................................................................................................ 8
  1.5 Structure ............................................................................................................................... 10

Chapter 2: Related Works ............................................................................................................. 10

Chapter 3: Methods and Technologies ......................................................................................... 13
  3.1 Summary ............................................................................................................................... 13
  3.2 Software Architecture ......................................................................................................... 13

Chapter 4: Methods of Approach ................................................................................................. 20
  4.1 Registration for Twitter API .................................................................................................. 20
  4.2 Installation of dependencies needed ..................................................................................... 21
  4.3 Creation of sentiment analyzer ............................................................................................ 22

Chapter 5: Testing and Evaluation ............................................................................................... 26
  5.1 Data gathering ...................................................................................................................... 26
  5.2 Methods .................................................................................................................................. 26
  5.3 Results / Validation ................................................................................................................ 27

6. Conclusions and Future Work .................................................................................................. 29
  6.1 Conclusions ........................................................................................................................... 29
  6.2 Future Work and Suggestions ............................................................................................ 30

Bibliography .................................................................................................................................... 32

Appendix .......................................................................................................................................... 34
Chapter 1: Introduction

In the first chapter of this dissertation we give an overview of the current state of existing conditions emphasizing on sentiment analysis tools their capabilities and obstacles. Following, we present the problem that we detect along with our objective and goals about this work. Special allusion made about Twitter as tool of data analysis following by a thorough description on sentiment analysis and the methodology around it. At last we provide the structure of this project and proceed with the presentation of relative chapters.

1.1 Overview

Given the scale of the ongoing digital revolution and the increasing reliance on digital technologies, there are obviously unanticipated opportunities for the study of human behavior and social trends. The rapid growth of mobile broadband adoption the latest years has undoubtfully played a pivotal role to this revolution. Linking billions of people in real-time from almost any place worldwide allows to understand in which way people engage in distinct aspects of life (shopping, politics, entertainment), serving as ubiquitous detector of contact for online activities (Mocanu et al., 2013). Taking this revolutionary shift into account in relation to the extensive use and the impact of micro blogging platforms on public discourse and communication, can someone consider the mass and power of the available to parse information which continues growing every second.

The great bloom of social media increased their usage as information sources providing qualitative details at a quantitative scale. Nowadays millions of people use social network websites like Facebook, Twitter, Instagram, etc. generating sentiment rich data in the form of tweets, posts, status updates, reviews, etc. In a world that people have become more than willing to share in public their opinions, experiences and feelings, social data represent the raw voice of the user opening the doors for research and experimental processes.

In the last few years, considerable attention has been paid to the utilization of this information and the extraction of high-valued results. Social data analysis attracts not
only the interest of the industry but also of researchers and academia whose efforts have been devoted to prove that this massive data can rapidly create an overview of almost any topic of interest. Living in a microblogging era where social media assumed to be the treasure trove of sentiment, opinion of the mass is important: 1) For the companies, as it is vital to have an overview of company’s health, feedback regarding their product brands or information on how to reach their competition. 2) For customers, as in a great extent people depend on published user content over online for decision making. And finally, 3) For political parties as it is crucial for them to have a clear view of their supports and opponents. For the academia on the other hand, sentiment analysis could promise satisfactory predictive results regarding human behavior.

Due to the importance of this field and the value of the information provided, numerous sentiment analysis tools have been developed in order to handle this type of tasks. They are categorized into three main categories: free online tools, paid tools and open source tools. Table 1 summarizes these categorizes providing briefly their most important limitations.

<table>
<thead>
<tr>
<th>Tools</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Free online tools</strong></td>
<td>1. Inaccurate results</td>
</tr>
<tr>
<td></td>
<td>2. 'Blackboxes' (we ignore the mechanism in the background)</td>
</tr>
<tr>
<td></td>
<td>3. Restricted functionalities/capabilities</td>
</tr>
<tr>
<td><strong>Paid tools</strong></td>
<td>1. High cost &gt;100$ per month</td>
</tr>
<tr>
<td></td>
<td>2. Demo versions 1-2 projects</td>
</tr>
<tr>
<td></td>
<td>3. Restricted functionalities</td>
</tr>
<tr>
<td><strong>Open source tools</strong></td>
<td>1. Small datasets</td>
</tr>
<tr>
<td></td>
<td>2. Lack of 'good' annotated datasets</td>
</tr>
<tr>
<td></td>
<td>3. Deal with issues like irony, sarcasm, poor spelling</td>
</tr>
</tbody>
</table>
1.2 Problem Statement and Objectives

Despite the fact the last decade there is a significant turnaround on sentiment analysis studies, we observed that people still is unable to understand the potential hidden behind this field. People have not realized yet the precision of possessing a tool like this, gathering a vast amount of data and interpreting the sentiment in order to predict behaviors. On the other hand, there may be those who find the development of this kind of software an impossible endeavor.

In order to enlighten this field a little more and give a motivation for more research activities on sentiment analysis field, our efforts focused on the development of a software tool that uses Twitter to understand how people are feeling about a topic that we choose.

Our added value under this thesis framework is to give a clear view of:

1. How someone can develop such a tool, avoiding the limitations of available commercial tools.
2. How to retrieve valuable information of the unorganized social data.
3. How to evaluate people’s sentiment about a particular topic with the power of machine learning.

1.3 Twitter

Twitter was created in March 2006 by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams and launched in July of that year. Twitter is a worldwide popular online news and social networking service where users post and interact with short SMS-like messages, called "tweets." In that regard, we could describe Twitter as a free, high-speed, global text-messaging service, that enables rapid and easy communication (Russell, 2013).

Tweets were originally restricted to 140 characters, but on November 7, 2017, the limit was doubled to 280 characters for all languages except Japanese, Korean and Chinese (Rosen, 2017). Registered users can post tweets, but those who are
unregistered can only read them. As of 2016, Twitter had more than 319 million monthly active users. The list of different ways to use Twitter could be really long, (share thoughts, links and pictures on Twitter, journalists comment on live events, companies promote products and engage with customer) and with 500 millions of tweets per day, there’s a lot of data to analyze and to play with (Bonzanini, 2016). Indicatively, on the day of the 2016 U.S. presidential election, Twitter proved to be the largest source of breaking news, with 40 million election-related tweets sent by 10 p.m. (Eastern Time) that day (Ember, 2016).

Nevertheless, Twitter is not just an information channel but also a treasure trove of sentiment as people display content full of emotions, opinions and behaviors. Human preferences are practically unpredictable. On that reason in order to understand and interpret them we invented on science of psychology and sociology to help us study these things. However, with this amount of data freely available we can do the same thing as scientists using Twitter as psychological tool. Given the fact that people around the world output thousands of reactions and opinions on every topic, every second and every day, Twitter work like a huge psychological database that is constantly being updated and can be used to analyze millions of text snippets in seconds.

1.4 Sentiment Analysis

Sentiment Analysis (SA) is an ongoing field of research in text mining field and the most complex machine learning task. As a process is by default a classification problem applied to determine whether a piece of writing (product/movie review, tweet, etc.) is positive, negative or neutral. Understanding and extracting human feelings from data makes SA a challenging and interesting task to work with. Humans are subjective creatures and opinions are important. Being able to interact with people on that level has many advantages for information systems finding wide applicability in different sectors of daily life. As a computational treatment of opinions, sentiments and subjectivity of text, can be used to identify the people’s attitude towards a topic, analyzing their reaction from variables like context, tone, emotion, etc. Of course, despite the great value of information extracted after implementing this task, we believe that we will never be able to reach a 100% accurate prediction about sentiment and this
is not because we don’t believe that human cannot construct such clever machines, but due to the fact that even two different people might have different opinion about the sentiment of the same text. On another reason, as we use our language to express emotions, words assumed to be the indicator of sentiment. The emerged problem here is that sometimes words are not enough to give as a realistic view of people’s feeling, as some of them have no direct translation giving a different meaning and sense depending on the nation.

Other limitations that make sentiment analysis even harder and for a machine a difficult task to accomplish are sarcasm, metaphors, irony, jokes poor spelling, lack of context, and the subtleties of sentiment. Additionally, negations and multiple sentiments in same text can also create difficulties in terms of the effective understanding behind an expression. Negations usually scored inaccurately by algorithms giving identically results e.g. the sentences “I like this weather” and “I do not like this weather” should be categorized as opposites. However, the majority of systems are unable to score them properly. Finally, concerning a complex text with multiple sentiments, some sections can be positive and others negative. The problematic issue here if the system will be able to understand the way that polarities should be aggregated. There are generally two main approaches to sentiment analysis:

The first one is the lexicon-based approach, where we split some given text into smaller tokens (words, phrases or whole sentences). This process is called tokenization. Then we count the number of times each word is showed up (bag of words model). At a next stage we look up the subjectivity of each word from an existing lexicon which is the database of emotion values for words prerecorded by researchers. After collecting those values, we are able to compute the overall subjectivity of our text (positive or negative).

The other approach uses machine learning. It is about a state-of-art but more computational expensive deep learning method which learns and generalizes vector representations from words. If we have a corpus of tweets that are labelled as positive or negative, we can train a classifier on it and then given a new tweet to classify it as positive or negative.

Undoubtedly, using a lexicon-based approach is easier but the machine learning methodology is more accurate. Limitations like the aforementioned ones are unable to
be recognized from lexicon algorithms as they seem like a particular thing but in reality they mean something totally different. Deep learning on the other hand can understand these parts and give valuable knowledge, as it takes abstract representations of what has already learned. These generalizations are called vectors and we can use them to classify data in efficiency.

1.5 Structure

In our project, we will discuss prior works before analyzing our approach, the performance of various models, and the quality of our results. The paper is organized as follows. Chapter 2 provides a review of the related literature. In particular, we analyze various studies associated with sentiment analysis in general as also potential prediction in finance field. In chapter 3 we detail the theoretical background of all methods and technologies used in terms of the development of our tool, whereas in chapter 4 we discuss our approach followed regarding the sentiment analysis process and algorithms used. Chapter 5 is referred to the implementation of the experimental process in order to test the feasibility of our software and reported the results of our empirical investigation which are then followed by discussions and conclusions in chapter 6.

Chapter 2: Related Works

Several publications have demonstrated various attempts in social media analytics including mainly computational methods. In much of the research, exceptional notice has focused on machine learning methods and classification algorithms due to the unstructured and unorganized form of social data (text, voice, images, videos) that makes traditional statistical methods unsuitable. Although some of these approaches are characterized by effectiveness and well-developed models, there are still interesting and relevant problems to be addressed.

In this Chapter we are going to demonstrate an indicative sample of related works on sentiment analysis field. Especially allusion will be given to finance field and the
potential correlation between sentiment and stock prediction, as it consists the subject of our experimentations.

After a thorough literature overview, we figured out that there are several scientific studies focused on the way someone can calculate the polarity of tweets. These studies demonstrate challenging methodologies mainly based on classification algorithms seeking to enhance the performance of the constructed model applying various tactics and preprocess methods. (Pak & Paroubek, 2010) propose sentiment analysis, building a Naïve Bayes classifier model trained with two different features: n-grams and part-of-speech distribution information. The training set used contained only tweets having emoticons. SVM and CRF also tested but Naïve Bayes yielded the best results. From their experimentations emerged interesting results. After related comparisons they figured out that using bigrams outperforms unigrams and trigrams performance. Supplementary, the attachment of negation words into n-grams enhances predictive accuracy in a high grade. Similar studies conducted by (Pang, Lee & Vaithyanathan, 2002) and (Dave, Lawrence & Pennock, 2003) who came out with contrary results. The first one reported that unigrams outperform bigrams when performing the sentiment classification of movie reviews, whereas the other claimed that bigrams and trigrams worked better for the product-review polarity classification. In 2014 (Liang & Dai, 2014) used Twitter API to collect twitter data. Their training data included data of three different categories (camera, movie, mobile) and labeled as positive, negative and non-opinions. Unigram Naive Bayes model was implemented and the Naive Bayes simplifying independence assumption was employed. They also eliminated useless features by using the Mutual Information and Chi square feature extraction method.

Taking into consideration all the above-mentioned works, we observe that despite the different time periods these studies conducted the subject of research activity moving into the same framework. In other words, we do not observe revolutionary approaches. Researchers choose to use the same machine learning algorithms that seem to perform well on sentiment analysis tasks and experiment with changes concerning the structure or the kind of datasets used.

However, as we have already mentioned there are two accepted sentiment analysis methods. Until now researchers’ interest has focused only on machine learning approaches as assumed to be the most accurate and well promising one, giving a lot of capabilities for progress. As a result, despite the large amount of studies that conducted
to address sentiment analysis problems, only a restricted amount of publications referred to lexicon-based approach and by extension to our work. (A. & Sonawane, 2016) presented an amazing comparative study regarding opinion mining techniques including both approaches and extracted very interesting results. Research outputs verified that machine learning methods, such as SVM and Naive Bayes appear the highest proportion of accuracy regarding sentiment prediction. On the other hand, lexicon-based methods are very effective in some cases but requires some effort in human-labeled document. They also concluded that the cleaner the data is, the better efficiency can be obtained in terms of models’ feasibility. Finally, they proved that use of bigram models provides better sentiment accuracy as compared to other models and suggested the combination of machine learning methods into opinion lexicon methods in order to improve the accuracy of sentiment classification and adaptive capacity to variety of domains and different languages. 

Following we are going to cite studies concerning stock market in order to give an overview of the research activity on finance sector. Stock market prediction has been extremely popular recently and the topic attracts people from various fields. There are numerous works conducted around this topic as except the prediction accuracy researchers also focused on the economic significance of each model. Indicatively, (Mao, Wang, Wei & Liu, 2012) investigated whether the daily number of tweets that mention S&P 500 stocks is correlated with several S&P 500 stock indicators. This was done at three different levels from the stock market to industry sector and individual stocks. They also applied a linear regression model to predict stock market indicators, using Twitter data as exogenous input. The extracted results demonstrated that daily number of tweets is correlated with the stock market indicators. Furthermore, it seems that Twitter data can be useful to predict stock market. In the same rationale (Pagolu, Challa, Panda & Majhi, 2016) tried to predict the correlation of Dow Jones Industrial Average Index (DJIA) using three different machine learning algorithms Random Forest, Logistic Regression and SMO using Word2vec and n-grams as training features. The results showed strong correlation between rise/fall in stock prices of a company and the public opinions or emotions about that company expressed on twitter through tweets.
Chapter 3: Methods and Technologies

In this chapter we are going to display an overview of methods and technologies used for the development of our sentiment analysis tool, presenting the background theory and principles needed. At first step a small recap on the application is mentioned. Following, the overall architecture of the system is described along with the functionality of individual parts. Finally, machine learning methods and algorithms are analyzed with respect to sentiment classification of the retrieved twitter data.

3.1 Summary

Our Twitter API analysis tool is a web based software application designed for scrolling Twitter, monitoring current trends. Through a homepage the user is able to enter a query of his interest (keyword or hashtag), a specific date and the number of days that the system should take into consideration. The expectation is to get a response concerning the general feeling of people regarding the given request. Every kind of information associated with the user and search content get stored into a database. In order to provide this kind of service to users, we implemented the system based on three basic components: a simple website giving user the capability to make his request to our system, a RESTful Web Service in Python as communication channel between the user and our internal system and finally a Python script working as Twitter crawler and sentiment analysis processor.

3.2 Software Architecture

In software development, it is common to separate a system into different smaller components. They usually separated in such a way that each one handles a distinct concern. Considering this thesis, as the objective of this project is far from an in-depth analysis of technical properties we are not going to get into much of details for each component of our system. Figure 1 represents the structure model we based on while building our tool.
When dealing with web applications the best programming language assumed to be PHP. Though as we work on a complex problem, Python and MySQL also applied seeking to ensure best performance and efficiency of the system.

In order to give a better view of our approach we present a brief interpretation of system’s functionality along with the technologies used:

1. The system built on Windows 10 operating system.
2. A local server created using Apache software and MySQL programming language.
3. For the database development and administration MySQL Workbench 6.3 and phpMyAdmin tools were used respectively. The EER diagram of our database is configured following in Figure 2.
The rules behind the designing model of the database are available below:

i. One specific keyword corresponds to many tweets

ii. One specific user corresponds to many keywords as one user can search for one or more different keywords

iii. Complementary to the two prior rules one specific user corresponds to more than one tweets

iv. If one user eliminated from the database, automatically every associated information including keywords and tweets will be permanently deleted.

v. If a specific keyword eliminated from the database every associated tweet will be permanently deleted.

4. Python 2.7 and Flask library used in terms of the Web Service implementation. Flask is a Python framework for web applications. Setting up Flask is simple and quick process, managing by pip package. All that has to be done is a pip
**install flask** prompt to the command line of Windows in order to be incorporated into system’s directory. Details of the theory and code used can be found in (Ullman, 2008).

5. Using PHP programming language, a simple website built, enable the user to gain access to the system either by using his credentials (email and password) or create a personal account (if his is not already enrolled). Information associated with the user stored into the database. After a successful login process, the authorized user is allowed to make his query about a topic of his interest. In the same time a list of options and capabilities become available. (In Appendices part there are screenshot flows of each option)

The requested data (keyword/hashtag, date, days) as well as user’s id will be the input of our internal system.

6. The internal system is a Python script responsible for the main operations of our tool. Due to the importance of this part, a thorough analysis of its functionality will be given in Chapter 4. However, what should be mentioned at this point concerns the generated data. As output the internal system returns json format data including the search of the user accompanied with polarity (general feeling) and subjectivity metrics.

7. The retrieved data addressed by the system in two ways. They are represented on software’s interface as response to user’s query, whereas at the same time the system stores them into the database. Screenshots of steps 5-7 are given below. Figure 8 is an indicative example of retrieved data as it is impossible to present all the fields and results included in our database. At this point we should also underline that we tried to include the majority tweet attributes provided by Twitter API, turning our database into a valuable information warehouse. Attributes included are available in Appendices part along with their descriptions.
Figure 1: User is added automatically into the database

Figure 3: User registration
Figure 5: User’s requested data

Figure 6: Response of the system on user's query
Figure 7: Database representation after information storage

Figure 8: Retrieved tweets corresponded to requested data of Figure 5
Chapter 4: Methods of Approach

This chapter integrates the entire methodology followed from data gathering to final results. As it has already mentioned Chapter 4 is actually a thorough description of the Internal System. Our efforts focus on performing sentiment analysis with Python after crawling Tweets on Twitter.

The reason behind our choice to work using Python is that among other languages outperforms to many points concerning opinion mining tasks. Python is an easy to use and understandable language offering a wide range of libraries that facilitate even more the procedure. In addition to this, there is extended and detailed documentation as also a variety of available material providing us direct solutions to coding obstacles that may arise. In terms of this part implementation three basic steps were followed, which are going to be discussed separately.

4.1 Registration for Twitter API

In order to gain access to the Twitter API and make requests about particular queries, we had to login the Twitter Developer website, and create an application. After registration we were able to get application’s Consumer Key, Consumer Secret, Access token and Access token secret. These credentials incorporated into our script as authenticate us as developers verifying our identity to Twitter. There are two basic ways to access Twitter data: 1) Twitter’s Search API and 2) Twitter’s Streaming API.

Search API gives access to tweets that already exist and is limited to the last 5,000 tweets per search criteria. On the other hand, Streaming API allows to get a sample of real-time tweets occur and push them to the user based on a set of search criteria. However, the sample provided by streaming API is at most 1% of the entire traffic and not randomized. Therefore, the data is not statistically representative. In addition to this using Streaming API provides us with a massive amount of data that could be a serious problem for our hardware equipment.

For all of the above mentioned elaborations, Search Twitter API was selected for our experimentations.
4.2 Installation of dependencies needed

Both Tweepy and Textblob libraries consist the cornerstone of our approach. The former is the key element to access the Twitter API. There is a bunch of Python-based clients that can used to interact with this service but Tweepy assumed to be the most important and straightforward to use. Tweepy provide us with capabilities that facilitate our process. For example, on literature we have seen many times issues regarding pagination and data redundancy. Twitter set a limit as far as the retrieved data is concerned embedding us to import an important number of items. Supplementary, repetitions of data could decrease significantly the effectiveness of our experimentations. To deal with pagination, Tweepy has the Cursor method which handles all this kind of work for us behind the scene. We just pass our parameters in Cursor interface and our code focus entirely on processing the results. Figure 9 represents Cursor expression for the demands of our research activity.

Figure 9: Tweepy performance in our case

Textblob is the dependency used in order to perform sentiment analysis. It is a new python natural language processing toolkit standing on the shoulders of giants like NLTK and Pattern and providing text mining, text analysis and text processing modules for python developers. Textblob also holds an extended documentation able to handle almost every opinion mining task in a fast and easy way.

With regards to sentiment, Textblob uses two measures: polarity and subjectivity. Polarity is the measure of how positive or negative is the feeling about a tweet and ranges from [+1, -1] with -1 representing very negative and +1 very positive. Subjectivity is the measure of how subjective or objective a text is. In regard to the Textblob scoring system, subjectivity metric ranges from [ 0, 1] with 0 being completely objective and 1 being completely subjective.
4.3 Creation of sentiment analyzer

In order to give an adequate description of the way we created our code script for sentiment analysis we present the background rationale through a sequence of relative steps demonstration. Figure 10 represents the model developed based on this rationale.

1. Input Data

A keyword, startdate and time period before the selected date are defined as input. User is able to scroll the Twitter for maximum 9 days backwards in time.

2. Tweets crawler

Given the predefined input data, we are connected with Twitter API in order to retrieve the tweets according to our interest topic. Due to time limitations and the volume of our experimentations it was decided to ask for retrieval 300 tweets per time. The corresponded tweets get fetched and stored into the database.

3. Preprocess Data

To reduce the consequences of noise that unorganized twitter data cause, data preprocessing was performed as optimization approach. This step assumed to be the most crucial task to our procedure as models built on optimized types of data tend to be of higher quality. Cleaning data from useless features can enhance in a great scale the predictive ability of our analyzer. Speaking about useless features we mean anything unable to provide us with sentiment or
anything able to distract the analyzer from interpret polarity in the most effective way. In order to face this challenge and clear our dataset from misleading information, a code script created based on regular expressions. Regular expressions are a sequence of specific characters that define a pattern. Python's built-in "re" module provides excellent support for regular expressions, with a modern and complete regex flavor. Symbols (also known as operators) and alphanumerical characters in specific order applied on subject string in order to execute a task. For our regular expression part development we used information available on Regular expressions.info website http://www.regular-expressions.info/python.html and the relative documentation module of Python official page https://docs.python.org/2/library/re.html. Figure 11 is just an indicative example of our preprocess methodology. The “Fix tweet lingo” part of this script was found available on https://github.com/stepthom/textblob-sentiment-analysis/blob/master/doAnalysis.py.

Figure 11: Preprocessing Twitter Data
4. Classification

At a next step of our approach the clean data are put into a classification algorithm. In the context of Sentiment Analysis there are three popular classifiers (Naïve Bayes, Maximum Entropy and Support Vector Machines). Our interest focused on Naïve Bayes Classifier which is available from our Textblob dependency.

As it has already described in Chapter 1 there are two approaches in terms of sentiment analysis. Lexicon-based approach and machine learning method. Due to the mechanism behind lexicon-based approaches that do not promise high-level of accuracy, researches avoid going with them. Despite the oppositions mentioned, we decided to test the feasibility of both approaches in terms of our experimentations. Textblob by default works with Pattern Analyzer that is a lexicon -based approach. This package is a convenient way to do a lot with Natural Language Processing tasks.

TextBlob goes along finding words and phrases it can assign polarity and subjectivity to, and it averages them all together for longer text. This method is based on uni-grams model and is able to assign polarity scores in a very clever way. In the same framework Textblob is able to handle some of our initial challenges giving more credits to our choice. Of course, we are not going to delve deeper on this information, but valuable material is cited on title: “Textblob sentiment: Calculating polarity and subjectivity”.

Regarding machine learning, Textblob provides Naïve Bayes Analyzer and Naïve Bayes classifier available to use. Naïve Bayes analyzer was rejected from our methodology plan after a number of tests. The analyzer has been trained on a movies dataset and as a result this type of analyzer could not promise satisfying outcomes considering that by default is biased to entertainment field prediction. We verified this assumption after a number of tests indicated that train test has great influence on unseen data.

So, moving on with Naïve Bayes classifier building our own classification system according to observations and characteristics that correspond better to our project, seems to be the best chosen scenario.
5. **Classified tweets and Sentiment**

Tweets are classified in three main categories according their polarity grade:

a. Positive: if polarity metric > 0  
   
b. Negative: if polarity metric <0  
   
c. Neutral: if polarity metric =0  

As a result, three different sentiment vectors are created which are normalized in order to calculate the overall sentiment of each category. The higher percentage is an indicator of people’s feeling about the input keyword. At this point we have to mention that we do not focus on the distinct number emerged as output but on the label this number referred to. What matters for us is the general feeling. For example, considering the results on Figure 6 we can say that people’s sense is positive for the particular days regarding IBM stock, avoiding the 0.6 number.

---

Figure 12: Sentiment analysis using Pattern Analyzer

```python
### -- Sentiment --
Polarity_List = []
Subjectivity_List = []
for tweet in tweetList:
    sent = TextBlob(tweet)
    Polarity_List.append(sent.sentiment.polarity)
    Subjectivity_List.append(sent.sentiment.subjectivity)
```

---

Figure 13: Normalization

```python
### -- Evaluate --
positive_polarity = [p for p in Polarity_List if p>0]
negative_polarity = [n for n in Polarity_List if n<0]
neutral_polarity = [r for r in Polarity_List if r==0]
total_size = float(len(positive_polarity) + len(negative_polarity) + len(neutral_polarity))
n_size = len(negative_polarity)/total_size
p_size = len(positive_polarity)/total_size
r_size = len(neutral_polarity)/total_size
```
Chapter 5: Testing and Evaluation

In order to test our tool’s functionality and ability to categorize tweets according to their sentiment impact we decided to make an experiment on the finance field. As research subject IBM stock was selected.

5.1 Data gathering

For the demands of this experiment we gathered tweets for IBM operations from the first half of 2017 (January – June 2017). Taking into account stock’s fluctuations as they presented on Yahoo Finance [https://finance.yahoo.com/quote/IBM?p=IBM](https://finance.yahoo.com/quote/IBM?p=IBM), we selected two points in each month. One point concerns a particular date where the stock’s price presented important increase, whereas the other associated with a particular date where stock’s price was in decrease.

For every point, we examined the feeling of people for three different time periods. More specifically, seeking to predict the progression of the stock (rise or fall) the selected date, the system took into consideration the feeling of people 1 day, 3 days and 7 days prior to this date.

The results showed slight differences between time periods without affecting the performance of our system. So, we decided to calculate the average percentage of these results.

5.2 Methods

The methodology followed in order to extract valuable information comprehends three main steps:

- Monitoring of polarity percentage using Textblob PatternAnalyzer method
- Monitoring of polarity percentage using Monkeylearn commercial tool
- Comparison between results
Monkeylearn is a paid commercial tool that promises the highest accuracy and best performance regarding text analysis and data processing. Considering that coordinates with big companies and uses state-of-art technologies hosts great databases with massive amount of tweets retrieved increasing automatically the quality of its services.

The reason that we choose Monkeylearn among other similar tools is that:

- Enable user to use it almost infinitely (even demo versions) with prerequisite the creation of new member account
- Connected to Twitter
- Works well with natural language (smileys, misspelled words, bad syntax)
- Distinguishes between negative facts and negative sentiments
- Give a great percentage of 75% as far as sentiment prediction is concerned

5.3 Results / Validation

Tables 1 and 2 depict the results of our experimentation’s conduct. The first column includes the date points selected for prediction. The second column consisted of the prices of the stock for each month retrieved from Yahoo finance and corresponds to closing prices. Yahoo Finance API updates them with 15 minutes delay but provides historical day-by-day stock data. The stock price variable will be used later to validate our system performance.

The percentages of PatternAnalyzer and Monkeylearn indicates the fluctuation of people’s feeling about IBM the time period before the chosen date points. It is worth mentioning at that all extract results had positive polarity index. We assume that the positivity of our outcomes is due to the fact that a lexicon-based approach used where actually the classification process based on “good” and “bad” words. Finance is a field that typically reports on stocks are described with neutral and positive vocabulary and mainly using verbs and adverbs. Text subjects lack of adjectives which seem to introduce strong sentiment. Under this framework it was decided to interpret our results observing the fluctuation of these percentages.

Despite the positive polarity if the value of the algorithm for one month is greater than the previous one then we consider that people’s opinion improved and conversely.
Table 2: Fluctuations of stock price and feeling of people (increase points)

<table>
<thead>
<tr>
<th>Date(prediction)</th>
<th>Stock price</th>
<th>PatternAnalyzer</th>
<th>Monkeylearn</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>25/1/2017 (up)</td>
<td>178.29</td>
<td>38%</td>
</tr>
<tr>
<td>February</td>
<td>15/2/2017 (up)</td>
<td>181.68</td>
<td>32%</td>
</tr>
<tr>
<td>March</td>
<td>16/3/2017 (up)</td>
<td>177.24</td>
<td>37%</td>
</tr>
<tr>
<td>April</td>
<td>17/4/2017 (up)</td>
<td>171.1</td>
<td>37%</td>
</tr>
<tr>
<td>May</td>
<td>4/5/2017 (up)</td>
<td>159.05</td>
<td>36%</td>
</tr>
<tr>
<td>June</td>
<td>12/6/2017 (up)</td>
<td>155.18</td>
<td>35%</td>
</tr>
</tbody>
</table>

Table 3: Fluctuations of stock price and feeling of people (decrease points)

<table>
<thead>
<tr>
<th>Month</th>
<th>Date(prediction)</th>
<th>Stock price</th>
<th>PatternAnalyzer</th>
<th>Monkeylearn</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>10/1/2017 (down)</td>
<td>165.52</td>
<td>30%</td>
<td>9%</td>
</tr>
<tr>
<td>February</td>
<td>8/2/2017 (down)</td>
<td>176.17</td>
<td>35%</td>
<td>20%</td>
</tr>
<tr>
<td>March</td>
<td>21/3/2017 (down)</td>
<td>173.88</td>
<td>35%</td>
<td>16%</td>
</tr>
<tr>
<td>April</td>
<td>19/4/2017 (down)</td>
<td>161.69</td>
<td>33%</td>
<td>11%</td>
</tr>
<tr>
<td>May</td>
<td>12/5/2017 (down)</td>
<td>150.37</td>
<td>28%</td>
<td>12%</td>
</tr>
<tr>
<td>June</td>
<td>7/6/2017 (down)</td>
<td>150.98</td>
<td>28%</td>
<td>15%</td>
</tr>
</tbody>
</table>

The validation of system’s efficiency was evaluated comparing the gradual progression between the price of IBM and algorithm’s outcomes. From the Tables we can easily observe that concerning PatternAnalyzer the results correlated in 10/12 cases with the fluctuation of stock price. Correspondingly, 8/12 cases of Monkeylearn results present correlation with the price fluctuation of IBM stock.
As a result, we can conclude that the prediction efficiency for PatternAnalyzer comes to a percentage of 83%, whereas for Monkeylearn this percentage is 66%. Comparing these two methods we can say that PatternAnalyzer present an important advantage over Monkeylearn.

6. Conclusions and Future Work

In the last part of our work we are going to express our feeling about the entire experience of working on this project. Results and observations of our experimentations are also about to be discussed along with relevant limitations. Finally, description of relevant approaches and capabilities are suggested.

6.1 Conclusions

Finalizing this project in terms of my Master’s thesis and having implemented this kind of software, I could say that sentiment analysis is a demanding but fascinating field that worth to be studied. Sentiment analysis shows a gradually progress without obvious borders. We are talking about a field without horizons but with incredible capabilities. Working these months on this subject I realized the massive amount of knowledge related to this research path from background theory to various technical approaches.

However, the great issue that should be pointed out here is not just the implementation of such a tool, but the fact that people have not realized yet the potential of these tools. Our objective through this work was to enlighten this field giving a more clear view on how someone can invest on this technology. Being able to gather large amount of data about any topic of interest gives tremendous advantages. Running in a data era, data and especially social data has become the new oil – it is like a raw material that can be extracted and refined. People should become aware of the value of the data they produce as also the meaning of store and utilization of this information. Taking all these into account we can say that undoubtfully we are talking about an upcoming well-promised field that need to be utilized effectively.
Concerning our experimentations and the results emerged, we can say that in terms of the predicted possible outcome and the actual behavior of the stock, the proportion of agreement was satisfying. Likewise, there was a relative agreement between our analyzer and Monkeylearn. To verify the validity of these proportions we propose experimentations about the prediction of different stocks. Of course, we have to mention that our analysis was based on a small sample only 12 points were analyzed and about 4000 tweets retrieved. We believe that our predictions could be more accurate giving more quality results if the amount of data gathered was much bigger. However, having developed a functional database give us the capability to select a wide range of information creating a valuable tool, ideal for future work and advanced research works.

6.2 Future Work and Suggestions

It is worth mentioning that as we decided to work with Textblob our analysis doesn’t take into account many factors. As it is impossible to incorporate all of our ideas in terms of this work and considering that research field does not actually end, we are going to present and suggest some ideas that will become actually the research area for me as researcher and will not remain as areas for future work.

One issue that we did not take into consideration while building our approach were Retweets. We decided not to discard retweets during the preprocess phase, considering that retweet actually consists a representation of a specific opinion. On the other hand, we know from statistics that repetition of data causes noise on data and creates misleading information. So as future experimentation we are going to test the validity of this assumption.

Another suggestion concerns the creation of customized classifier. Building feature extractors based on observations and characteristics that corresponds the problem analyzed, opens the door for a wide range of experimentations increasing prediction accuracy. The problems that have to deal with when come with this approach are time consumption (a lot of time is demanded during the training phase), as also the lack of good annotated datasets on finance field.
Additionally, we believe that tweets’ attributes that have retrieved and stored in our database can expand incredible the dimensions in terms of our research activities and give a lot of capabilities concerning the extracted information. For example, taking into account the “followers” attribute, we could select to monitor the behavior and opinion only of the “strong” users (users with a great amount of followers), or we could make comparisons between the feeling of “strong” and “weak” users.

Finally, as future work we also promise to improve our technical part adding more options and facilitations for the user like choice of Algorithmic model, increasing his satisfaction after his experience with our software.
Bibliography


### Appendix

1. Table below presents Tweets attributes included in our database along with their meaning

<table>
<thead>
<tr>
<th>Tweet Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>users_user_id</td>
<td>unique id given to user during enrollment to db</td>
</tr>
<tr>
<td>keywords_idkeywords</td>
<td>unique id given to keyword after a search</td>
</tr>
<tr>
<td>created_at</td>
<td>UTC time when this Tweet was created</td>
</tr>
<tr>
<td>favorite_count</td>
<td>Indicates approximately how many times this Tweet has been liked by Twitter users</td>
</tr>
<tr>
<td>favorited</td>
<td>Indicates whether this Tweet has been liked by the authenticating user</td>
</tr>
<tr>
<td>lang</td>
<td>language of Tweet</td>
</tr>
<tr>
<td>retweet_count</td>
<td>Number of times this Tweet has been retweeted</td>
</tr>
<tr>
<td>retweeted</td>
<td>Indicates whether this Tweet has been Retweeted by the authenticating user</td>
</tr>
<tr>
<td>source</td>
<td>Utility used to post the Tweet</td>
</tr>
<tr>
<td>text</td>
<td>The actual UTF-8 text of the status update</td>
</tr>
<tr>
<td>truncated</td>
<td>Indicates whether the value of the text parameter was truncated</td>
</tr>
<tr>
<td>user_created_at</td>
<td>UTC time when this User was created</td>
</tr>
<tr>
<td>user_followers_count</td>
<td>number of user followers</td>
</tr>
<tr>
<td>user_location</td>
<td>source user used to create the Tweet</td>
</tr>
<tr>
<td>user_lang</td>
<td>language of user</td>
</tr>
<tr>
<td>user_name</td>
<td>name of original Tweet’s author</td>
</tr>
<tr>
<td>user_screen_name</td>
<td>screen name of original Tweet’s author</td>
</tr>
<tr>
<td>user_friends_count</td>
<td>number of accepted users</td>
</tr>
<tr>
<td>polarity</td>
<td>sentiment score as output attribute</td>
</tr>
<tr>
<td>subjectivity</td>
<td>subjectivity score as output</td>
</tr>
</tbody>
</table>