Eindhoven University of Technology

MASTER

Measuring the influence of greenwashing practices on the public opinion a Twitter sentiment analysis

Kroes, D.J.

Award date:
2018

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Eindhoven, February 2018

Measuring the influence of greenwashing practices on the public opinion

- A Twitter sentiment analysis

by
Dian J. Kroes

BSc Architecture, Urbanism and Building Sciences
Student identity number 0753153

in partial fulfillment of the requirements for the degree of

Master of Science
in Innovation Management

Supervisors:
dr. A.M. Wilbik, TU/e, Information Systems
dr. M. Firat, TU/e, Information Systems
Isa Yilmaz MSc, EY, Data & Technology

TUE. School of Industrial Engineering.
Series Master Theses Innovation Management

Subject headings: greenwashing, sentiment analysis, data analysis, Twitter, BP, Deepwater Horizon

Preface & acknowledgments

This thesis presents the result of the final graduation project at the chair of Information Systems at the University of Technology Eindhoven (TU/e) and the Department of Data and Technology at Ernst & Young (EY).

There are several people I would like to thank. First I would like to thank Anna Wilbik for being my main supervisor for the project. Your help was always appreciated and directed me towards more structure in the project and helped me to try and implement a new classification method. I have experienced your guidance as supportive and friendly. Second I would like to thank my supervisor from EY, Isa Yilmaz. Thank you for your practical guidance and involvement in the project. Thirdly I would like to thank Murat Firat for his strong mathematical feedback which helped me to polish the mathematical parts of the thesis.

I would like to thank all the friends that have supported me not only throughout this thesis but during my studies at the TU/e. Some parts have been tougher than others, but in the end it has been all worth it. I thank you all for being in my life and helping me become who I am today. Last but not least I would like to thank my family for their unconditional love and support, especially my mother for always being there. I can always count on you, thank you for all your support.

Abstract

Nowadays there are a lot of different greenwashing practices ongoing, these practices hinder honesty and transparency which is needed to make meaningful decisions for businesses and consumers. Greenwashing can falsely influence company investment decision or consumer decisions to buy a product or not. If greenwashing works it provides no incentive to actually invest in more eco-friendly solutions which is crucial for the survivability of the planet and the people. One sector that is famous for greenwashing is the energy sector, it uses greenwashing to create a better profile of the companies and its sector in the minds of the public. There is little research towards the effectiveness of greenwashing and the influence of these practices on the public opinion, which makes it hard to assess the effects of greenwashing. In this project the oil spill in the Gulf of Mexico from BP in 2010 is used as a specific case to try and measure the effects of greenwashing practices.

Greenwashing has been around for some time and has found its way into society. It can be defined as simple as "the unwarranted or overblown claims of sustainability or environmental friendliness in an attempt to gain market share" (Dahl, 2010). Greenwashing is about intentionally making claims that are not true and are for the benefit of a company. The greenwashing practices cause confusion and make consumers suspicious about any company that makes green claims. Greenwashing relates to Corporate Social Responsibility (CSR) because the CSR strategy applied by a company can determine if greenwashing can be used to enhance a firm's reputation. The antecedents to greenwashing consist of market drivers such as competitive pressure, nonmarket drivers as regulations and law, organizational drivers which stem from how a company is run and individual psychological drivers such as optimistic bias and narrow decision making. It appears that the lack of legislation and the pressure from consumers for companies to go "green" are the main drivers the promote greenwashing. This all relates to how greenwashing is used in the oil and gas sector where there is extensive use of greenwashing.

To prove that greenwashing is used by BP after the Deepwater Horizon oil spill the concept of framing is used to analyze TV commercials aired by BP after the disaster. Framing analyzes how information is used and communicated towards an audience to address events and issues. Frameworks are used to convey information to the audience and make use of a frame focus and frame functioning, the focus refers to the event and how it is presented where the functioning passes the moral judgment on the subject presented. In total there are 13 advertisements aired that are considered to be greenwashing events because they make use of framing in such a way that the audience is deceived. These commercials do not represent the full facts available, episodic and didactic frames are used to "teach" the audience and leave them with an incomplete and narrow view toward the effects of the oil spill.

To make a sentiment trend that can show if the identified greenwashing practices have any influence on the opinion expressed on Twitter several algorithms are used to classify tweets into identified sentiment classes ("Positive", "Neutral" and "Negative"). The starting point is a standard Naïve Bayes model based on the Bayesian method of probability and takes only the words that appear in a Tweet into account. Also, a Multinomial Naïve Bayes model is deployed that is based on the same Bayesian probability but also takes the word counts into account. The third algorithm is a novel and new algorithm for sentiment analysis: Fuzzy Fingerprints (FFP). It constructs fingerprints of large datasets and classifies Tweets based on those constructed fingerprints. Model four uses an improved preprocessing method with negation and emoticon handling to improve the results of the used algorithms so far. The final model improves the existing Multinomial Naïve Bayes model by penalizing the probabilities based on word occurrences in the different sentiment classes. All of these algorithms are based on supervised learning as they take in data for training that needs to be accompanied with sentiment labels. For the construction of this sentiment trend there are two datasets used, one is a dataset from Niek Sanders accompanied by sentiment labels to test the results of the 2nd dataset against. The other dataset is Twitter data regarding BP extracted from the Twitter webpage. The data is explored and it shows that the datasets are imbalanced regarding the sentiment labels and there

are not sentiment labels for the BP data, but otherwise there are no major issues or deficiencies apparent in the data.

Before the models can be used the data is prepared to make it more suitable for analysis, in total there are 13 pre-processing steps used to clean the Tweets ranging from converting contracted words and slang to lemmatization. The result is a set of cleaned tweets that only consists of words that hold sentimental value and have the most common form, this data serves as input for the algorithms.

The first Naïve Bayes model makes use of the words in the Tweets and the word features extracted from all the Tweets in a sentiment class to classify the Tweets. 10-fold cross-validation is used to reduce bias, the data is split into 10 sets of 10% data, here 90% of the data is used for training and 10% test data where 10 cycles are conducted that use 10% as test data and the remainder as training data. The first model uses the two datasets to compare the results of both datasets as the BP dataset contains no sentiment labels and have to be added manually. It serves as a check for errors in the manual labeling process of the BP data. F-1 scores for the classifier range from 0.66 for the un-preprocessed BP data to 0.56 for the data of Niek Sanders and 0.65 for the pre-processed data of BP compared to 0.57 for the pre-processed data of Niek Sanders. The precision and recall scores per sentiment class show that the classification of the "Positive" sentiment class does not perform well due to the low amount of data points available compared to the "Negative" and "Neutral" class.

The Multinomial Naïve Bayes model classifies Tweets based on the words and the number of times it appears in a sentiment class. Again a 10-fold cross validation is used to split the data. Only the BP dataset is used and the results have improved compared to the standard Naïve Bayes model. The Multinomial model achieves an F-1 score of 0.66 for the un-preprocessed data and 0.69 for the pre-processed data. Again the "Positive" sentiment class is the worst performing class which reduces the performance of the classifier. A new model needs to be used that can better handle imbalanced data. The Fuzzy Fingerprint model is used for this particular reason.

The FFP model constructs small fingerprints out of large data that represents the large dataset, it uses the words in these fingerprints to classify Tweets in the test set. The fingerprints are constructed based on the word frequency order in the data per sentiment class and a fuzzification formula is applied to assign a fuzzy value to the ranked words. When there are corresponding words in a Tweet and fingerprint these values are summed to compute a final score used for classification. For the un-preprocessed data the model achieves the highest F-1 scores for fingerprints that contain 900 words (0.605). The pre-processed data achieves the best results with again 900 words in a fingerprint with an F-1 score of 0.648.

The fourth model uses improved pre-processing to improve the classification results. In this pre-processing method also negation is used to make words more sentiment class specific and emoticons are used as they hold sentimental value. The results show some minor improvements for the Bayesian methods but none for the FFP method. The Multinomial Naïve Bayes model performs best with an F-1 score of 0.70.

The final model uses techniques learned from the Fuzzy Fingerprint model to I prove the best performing algorithm so far. It used the technique of the Inversed Document Frequency (IDF) to penalize the conditional probabilities of the Multinomial Naïve Bayes. The results from this new model are good and the F-1 score is increased by 0.16 compared to the standard Multinomial Naïve Bayes classifier. The model shows very good prediction scores for all sentiment classes and good recall scores for the "Negative" and "Neutral" sentiment classes.

The sentiment trend is constructed based on the best performing method. First, the predicted "Positive" and "Negative" tweets are shown in a sentiment graph that also shows the identified greenwashing events. It appears that there are not many positive Tweets expressed regarding BP on

Twitter, there is a major spike in negative sentiment right after the oil spill on 20 April 2010. For constructing the actual sentiment trend there are values assigned to the predicted classes and per day these values are summed. The sentiment shows a big drop after the oil spill but unfortunately shows no relation to the identified greenwashing events. There are no consistent drops or spikes in the sentiment trend in the periods after a commercial is aired. There is also sought for a relationship between the stock prices of BP and the sentiment trend. Besides the drop in stock prices after the oil spill there is no relationship found between other sentiment drops or spikes and the stock prices.

The findings thus show no relationship between the greenwashing practices identified and the sentiment expressed on Twitter. This does not conclude that there is no relationship, but in this case it is not visually present. This can be due to the severity of the oil spill case as the sentiment expressed can be influenced by the severity of the oil spill. Also, the classifiers are not perfect (F-1 score of 0.86) which means that the results are not 100% accurate and thus influence the sentiment trend that is plotted.

Sentiment analysis can partially be deployed for measuring greenwashing practices and aid the discussion towards the effectiveness of greenwashing practices. It can never establish any causal relationship as the classifiers do not perform perfect and there are influences that stem from events that are outside of the scope of a sentiment analysis. What sentiment analysis can do is show the peaks and valleys in online sentiment in relation to certain identified events with relatively low effort as most of this data is publicly available.

Table of Contents

Preface & acknowledgments III

Abstract IV

List of figures and tables X

Abbreviations and glossary XII

1 Chapter: Introduction 1
1.1 Background 1
1.2 Relevance 1
1.3 Problem statement 2
1.3.1 Research questions 2
1.4 Scope 3
1.5 Delimitations 3
1.6 Research design 3
1.6.1 Phase 1: Business understanding 3
1.6.2 Phase 2: Data understanding 4
1.6.3 Phase 3: Data preparation 4
1.6.4 Phase 4: Modeling 4
1.6.5 Phase 5: Evaluation 5
1.6.6 Phase 6: Iteration 5
1.6.7 Phase 7: Deployment 5
1.7 Thesis outline 6

2 Chapter: Theoretical Background 7
2.1 Summary sentiment analysis systematic literature review 7
2.2 Greenwashing literature review 8
2.2.1 Greenwashing 9
2.2.2 Corporate social responsibility 11
2.2.3 Antecedents of greenwashing 12
2.2.4 Greenwashing in the oil & gas sector 13
2.2.5 Conclusion 14

3 Chapter: Case BP greenwashing 15
3.1 Framing 15
3.2 Identifying greenwashing practices 16
3.3 Frame analysis 16
3.3.1 Voices of the Gulf: business owner's impact 16
3.3.2 Voices of the Gulf: tourism 18
3.3.3 BP's commitment 19
3.3.4 Conclusion 20

4 Methodology 21
4.1 Classification background 21
4.1.1 Naïve Bayes 21
4.1.2 Multinomial Naïve Bayes 22
4.1.3 Fuzzy Fingerprints 23
4.1.4 Evaluation metrics 26
4.2 Classification models 28

4.2.1 Naïve Bayes 28
4.2.2 Multinomial Naïve Bayes 28
4.2.3 Fuzzy fingerprint 29
4.2.4 Improving pre-processing 29
4.2.5 IDF penalized Multinomial Naïve Bayes 29
4.3 Data acquisition 30
4.3.1 BP data 30
4.3.2 Niek Sanders data 32
4.4 Data Exploration 32
4.4.1 BP data 32
4.4.2 Niek Sanders data 33
4.5 Data preparation 36
4.5.1 Manual labeling 36

5 Modeling and results 37
5.1 Data preprocessing 37
5.1.1 Removing duplicates 38
5.1.2 ReTweet removal 38
5.1.3 Tweet hashtag removal 39
5.1.4 Tweet removal usernames 39
5.1.5 Tweet Conversion lowercase 40
5.1.6 Tweet HTML removal 40
5.1.7 Converting contracted words 40
5.1.8 Punctuation removal 41
5.1.9 Slang conversion 41
5.1.10 Standardizing words 41
5.1.11 Spelling correction 42
5.1.12 Filter out stop words 42
5.1.13 Lemmatization 42
5.2 Model 1 – Naïve Bayes 43
5.2.1 Evaluation classification model 1 45
5.2.2 Conclusion classification model 1 46
5.3 Model 2- Multinomial Naïve Bayes 46
5.3.1 Evaluation classification model 2 48
5.3.2 Conclusion classification model 2 48
5.4 Model 3- Fuzzy fingerprints 48
5.4.1 Evaluation classification model 3 50
5.4.2 Conclusion classification model 3 50
5.5 Model 4- Improved pre-processing 52
5.5.1 Evaluation classification model 4 53
5.5.2 Conclusion classification model 4 54
5.6 Model 5- Penalized Multinomial Naïve Bayes 54
5.6.1 Evaluation classification model 5 55
5.6.2 Conclusion classification model 5 56

6 Influence analysis 57
6.1 Sentiment trend 57
6.2 Sentiment relation to stock prices 59
6.2.1 Data acquisition 59
6.2.2 Data understanding 59
6.2.3 Data Validity 60
6.2.4 Relationship sentiment and stock prices 60

7 Conclusion 61
7.1 Summary 61
7.2 Discussion 62
7.3 Future work 63
7.4 Limitations 64
7.5 Recommendations 64

Reference list 66

8 Appendices 72
8.1 Systematic literature review 72
8.1.1 Introduction 74
8.1.2 Research area 75
8.1.3 Search protocol 78
8.1.4 Analysis 82
8.1.5 Discussion and conclusions 93
8.1.6 References 94
8.1.7 Appendix 92
8.2 TV commercials aired 96
8.2.1 Voices of the gulf: business owner's impact 96
8.2.2 Voices of the gulf: tourism 96
8.2.3 BP's commitment 96
8.3 Protocol Manual labeling Tweets 97
8.4 Results classifier 1 98
8.5 Results classifier 2 102
8.6 Results classifier 3 104
8.7 Results classifier 4 108
8.8 Results classifier 5 112
8.9 Sentiment trend 113
8.10 Web crawler code 116
8.11 Data cleaning code 1 118
8.12 Classification code 1 Naïve Bayes -BP 122
8.13 Classification code 1 Naïve Bayes – Niek Sanders 125
8.14 Classification code 2 Multinomial Naïve Bayes - BP 129
8.15 Classification code 3 Fuzzy Fingerprint - BP 131
8.16 Improved pre-processing 138
8.17 Classification code IDF Multinomial Naïve Bayes 143

List of figures and tables

Figure 1: CRISP-DM framework (Azevedo & Santos, 2008) 4
Figure 2: Thesis research design 5
Figure 3: drivers of greenwashing (Delmas & Burbano, 2011) 12
Figure 4: Greenwashing TV advertisements 20
Figure 5: fuzzification functions 24
Figure 6: sentiment model 1 28
Figure 7: Sentiment model 2 29
Figure 8: sentiment model 3 29
Figure 9: sentiment model 4 29
Figure 10: sentiment model 5 30
Figure 11: Working Python-based web crawler 32
Figure 12: Tweet amount per date 33
Figure 13: Tweet amount per date 34
Figure 14: Tweet topic distribution 35
Figure 15: Tweet sentiment distribution per topic 35
Figure 16: Pre-processing steps cycle 1 37
Figure 17: classification approach Naïve Bayes 43
Figure 18: classification approach Multinomial Naïve Bayes 47
Figure 19: classification approach fuzzy fingerprints 49
Figure 20: Adapted multinomial Naïve Bayes approach 55
Figure 21: "Positive" and "Negative" predicted sentiment 57
Figure 22: Predicted sentiment trend 58
Figure 23: Boxplots BP stock data 59
Figure 24: Relationship sentiment trend and stock price 60

Table 1: Elements of greenwashing (Laufer, 2003) 9
Table 2: The seven sins of greenwashing (TerraChoice, 2010) 10
Table 3: Notation Naïve Bayes 22
Table 4: Notation Multinomial Naïve Bayes 23
Table 5: Notation fuzzification 24
Table 6: Notation Fuzzy Fingerprint 25
Table 7: ICF scores and ranking example 25
Table 8: Notation T2S2 score 26
Table 9: Notation evaluation metrics 26
Table 10: Multiclass confusion matrix 27
Table 11: search term 1 and constraints 30
Table 12: search term 2 and constraints 30
Table 13: search term 3 and constraints 30
Table 14: search term 4 and constraints 31
Table 15: Dataset BP Tweets format 32
Table 16: Properties BP dataset 33
Table 17: Dataset Sanders Twitter 34
Table 18: Sample dataset Sanders 34
Table 19: Properties Niek Sanders dataset 36
Table 20: Example data preprocessing 38
Table 21: Data after removing duplicates 38
Table 22: Data after removing reTweets 39
Table 23: Data after removing hashtags 39
Table 24: Data after removing usernames 39
Table 25: Data after removing capital letters 40

Table 26: Tweets after removing HTML links 40
Table 27: Tweets after converting contracted words 41
Table 28: Tweets after removing punctuation 41
Table 29: Tweets after converting slang 41
Table 30: Tweet after standardizing words 42
Table 31: Tweets after spellings check 42
Table 32: Tweet after stop word filtering 42
Table 33: Tweets after lemmatization 43
Table 34: Naïve Bayes results classification BP data (un pre-processed data) 44
Table 35: Naïve Bayes results classification BP data (pre-processed data) 45
Table 36: Naïve Bayes results classification Niek Sanders data (un pre-processed data) 45
Table 37: Naïve Bayes results classification Niek Sanders data (pre-processed data) 45
Table 38: Multinomial Naïve Bayes results classification BP data (un pre-processed data) 48
Table 39: Multinomial Naïve Bayes results classification BP data (pre-processed data) 48
Table 40: Fuzzy Fingerprint $\mu 1i$ results classification BP data (un pre-processed data) 51
Table 41: Fuzzy Fingerprint $\mu 2i$ results classification BP data (un pre-processed data) 51
Table 42: Fuzzy Fingerprint μ1i results classification BP data (pre-processed data) 51
Table 43: Fuzzy Fingerprint $\mu 2i$ results classification BP data (pre-processed data) 51
Table 44: Tweet examples 52
Table 45: Tweets after converting emoticons 52
Table 46: Tweets after handling negation 52
Table 47: Naïve Bayes results improved pre-processing 53
Table 48: Multinomial Naïve Bayes results improved pre-processing 53
Table 49: Fuzzy Fingerprint $\mu 1i$ results improved pre-processing 53
Table 50: Fuzzy Fingerprint $\mu 2i$ results improved pre-processing 53
Table 51: Adapted Multinomial Naïve Bayes results 55
Table 52: BP stock prices dataset 59

Abbreviations and glossary

ACM –	Association for Computing Machinery
ADS –	American Depositary Share
API –	Application Programming Interface
BN –	Bayesian Network
BOW –	Bag Of Words
BP -	British Petroleum or Beyond Petroleum
CFC –	ChloroFluoroCarbon
CRISP-DM –	Cross-Industry Standard Process for Data Mining
CRM –	Customer Relationship Management
CSR –	Corporate Social Responsibility
CSV –	Comma Separated Value
ERP –	Enterprise Resource Planning
FFP –	Fuzzy Fingerprint
FTSE –	Financial Times Stock Exchange
GEO –	Geographical
GRI –	Global Reporting Initiative
IBM –	International Business Machines Corporation
ICF -	Inversed Class Frequency
IDF -	Inversed Document Frequency
ISO –	International organization for standardization
ME –	Maximum Entropy
ML –	Machine Learning
NB –	Naïve Bayes
NGO –	Non-Governmental Organization
NLP -	Natural Language Processing
NN –	Neural Network
POS –	Part Of Speech
ROC –	Receiver Operating Characteristic
SVM –	Support Vector Machine
URL –	Uniform Resource Locator
USD –	United States Dollar

"Honesty is a very expensive gift, don't expect it from cheap people"
– Warren Buffet

1 Chapter: Introduction

One's life is a dynamic process, not a static one. A dynamic process implies change and adjustments which results in the shifting of what might be important in life. As my own life continues to change and new experiences arise it becomes clearer what is truly important. Not only should we take good care of ourselves and our loved ones, we should take good care of every life-form on this planet as it is our collective responsibility to ensure a clean and habitable environment for everyone. In my opinion, there is a lack of honesty about the environmental impact of companies, this makes it hard to assess what our environmental impact actually is and what the consequences are. Being honest implies being transparent about the actions that a company or a person takes, this information either puts us in a strong or vulnerable position. Businesses that do not act in an eco-friendly way are frequently not transparent about the impact of their business processes because this might influence consumer decisions toward their offerings (Collins, Steg, & Koning, 2007). Instead of being honest, these businesses can promote their actions through "greenwashing". Greenwashing is the spreading of incomplete or false information toward the public regarding the company's environmental responsibility (Aggarwal & Kadyan, 2014; Furlow, 2010). These practices lead the public to believe that a company or its products are more environmentally responsible than that they actually are. It is important to gain insight into the influence of such practices on the public opinion because sustainability and ecological impact are a major societal issue nowadays. It influences business decisions, but it also influences consumer decisions to buy a product or not (Aggarwal & Kadyan, 2014; Collins et al., 2007; L. D. Mitchell & Ramey, 2011; Welford, 2013). An industry that is famous for using greenwashing is the energy sector (M. Cherry & Sneirson, 2011), the success of greenwashing practices in this industry might partly explain the lack of incentives regarding more environmental friendly developments in this sector. To promote these kinds of developments in the energy sector there first needs to be more transparency about the effects of their greenwashing practices on the public. This transparency is needed to create awareness among consumers and stakeholders, they can then demand that this sector needs to invest more in eco-friendly alternatives.

1.1 Background

Greenwashing has been researched extensively since the interest towards green companies spiked and social responsible investing became popular again in the beginning of the millennium. Since then greenwashing practices have been growing in countries such as the United States because there are loose regulations regarding environmental marketing (Dahl, 2010; Furlow, 2010). Studies have been published towards the antecedents of greenwashing and how companies use greenwashing to their advantage, but few studies try to prove the effects of greenwashing. Greenwashing is a problem that occurs in every sector but one of the most environmentally impactful sectors that use greenwashing is the energy sector. This sector is also famous for its greenwashing practices to reduce the negative associations that come from focusing on non-renewable energy forms and fossil fuels (M. A. Cherry & Sneirson, 2012; M. Cherry & Sneirson, 2011).

1.2 Relevance

Greenwashing is no new concept and many papers have been written regarding "greenwashing" or "false green advertising". Studies have been written about what greenwashing is and how it affects consumers and companies (Dahl, 2010; Furlow, 2010; Laufer, 2003; L. D. Mitchell & Ramey, 2011). There have also been studies about the antecedents of greenwashing and the solutions to stop these practices (Aggarwal & Kadyan, 2014; Horiuchi, Schuchard, Shea, & Townsend, 2009; Oppolecchia,

2010). There are also studies that focus specifically on the energy sector or BP in particular and its greenwashing practices (Barrage, Chyn, & Hastings, 2014; M. A. Cherry & Sneirson, 2012; M. Cherry & Sneirson, 2011). Most of these studies so far do not measure the actual influence of the greenwashing practices on the public opinion, only one paper (Barrage et al., 2014) uses BP's advertising data, gasoline price data and green preferences data to prove the effects of BP's greenwashing. To my knowledge, there have been no attempts made to actually measure the influence of greenwashing practices on the public opinion. Mitchell & Ramey (2011) have also called for research that addresses measuring the influence of greenwashing practices to show the impact and the (un)usefulness of greenwashing.

This research tries to show if the greenwashing practices applied by BP after the Deepwater Horizon oil spill have been successful. This information contributes to the rich literature available by not only identifying greenwashing practices, or looking at the causes, but it tries to quantify the impact of the greenwashing events on the public opinion. Furthermore, it applies a new analysis form that to my knowledge, has not been used in greenwashing literature. Twitter data is used to conduct a sentiment analysis by means of classifying Tweets and then analyzing the sentiment trend. This sentiment trend is then used to measure the public opinion about BP starting in 2009 and ending in 2014. This analysis thus contributes in two ways. First, it contributes practically by showing the (un)usefulness of greenwashing and second it contributes scientifically by applying new analysis methods to quantify the impact of greenwashing.

1.3 Problem statement

Within the energy sector there are a lot of greenwashing practices to create better company profiles in the minds of the public. BP is accused of using greenwashing to create a more eco-friendly and sustainable image for their company than their actions suggest. There is little research towards the effectiveness of greenwashing and the influence of these practices on the public opinion which makes it hard to assess the effects of greenwashing. The aim of this thesis is to increase the understanding of the influence that greenwashing practices have on the public opinion by using sentiment analysis.

1.3.1 Research questions

The main research question is as follows:
How can sentiment analysis be used to measure the influence of BP's greenwashing practices on the public opinion?

In order to answer the main research question, several sub-questions need to be answered:

- 1. How can greenwashing events be identified that BP used concerning Deepwater horizon?
- 2. What data and attributes are needed to analyze the influence of greenwashing practices on the public opinion?
- 3. How can a ground-truth be established that supports the sentiment expressed by the public opinion?
- 4. How can the sentiment expressed by the public opinion be measured?
- 5. What is the influence of greenwashing practices on the public opinion?

1.4 Scope

Within this thesis project I will try to measure the influence of greenwashing practices on the public opinion via a sentiment analysis on Twitter data. Beyond Petroleum/British Petroleum (BP) has been accused of using greenwashing after the environmental disaster caused by the drilling platform Deepwater Horizon in the Gulf of Mexico in 2010 (Frick, 2014). A case will be developed to show the greenwashing events used by BP. After the greenwashing events have been established a sentiment trend will be developed based on Twitter data to show the influence of these practices on the public opinion expressed on Twitter.

1.5 Delimitations

It can be argued that analyzing Twitter alone does not fully represent the public opinion, other media such as Facebook, (micro)blogs and forums also need to be analyzed to get a better grasp of the whole public opinion picture. The above-mentioned media all contain opinions expressed online, there are also people who do not express their opinion online. Thus to get the best public opinion a lot of data from multiple sources, both online and offline, need to be gathered. Due to time limitations it is impossible to gather all this data. Even data gathering from multiple online sources is tricky as different data need different processing and classification, i.e. an approach that works for one type of data is not necessarily good for other types of data. There is thus chosen to limit the data to one information source namely Twitter. Twitter gives access to their Tweets by use of a Twitter Application Programming Interface (API), therefore their data is available and this platform is chosen.

1.6 Research design

Sentiment analysis is a part of the text mining field which lies within the data mining domain. Sentiment analysis is also commonly referred to as "*opinion mining*", it is a technique that derives opinions toward a specific topic from data. The text mining research field uses different data mining techniques such as classification, and clustering approaches to derive specific information on a systematic basis from data sources which are too vast to be analyzed by humans (Talib, Hanif, Ayesha, & Fatima, 2016). In this thesis a classification approach based on supervised learning will be used that classifies Tweets into sentiment categories based on training data that already has sentiment labels. This project thus needs a data mining framework to guide the process. The Cross-Industry Standard for Data Mining (CRISP-DM) framework is widely used and acknowledged to be a suitable framework for data mining projects (Piatetsky, 2014). The chosen framework is presented and the sub-questions are divided into the subjects of the CRISP-DM framework.

The CRISP-DM framework is a cycle that is comprised of 6 stages, see Figure 1: CRISP-DM framework on the next page. This framework has been widely used and is well proven within the data mining field (Chapman et al., 2000), although this framework has been invented many years ago it is still widely used (Piatetsky, 2014). The framework does not exclude any possible routes of the data mining process but models and idealized sequence of events. It is an iterative process and some tasks may need to be refined and iterated on when subsequent steps do not provide satisfactory results.

1.6.1 Phase 1: Business understanding

This phase starts with studying the background of the thesis subject. A systematic literature review towards sentiment techniques is conducted, a proper understanding of the workings of greenwashing needs to be obtained by studying relevant greenwashing literature. From this literature a business case is developed which argues that BP used greenwashing to improve its position within the minds of the public after the Deepwater Horizon oil spill. Greenwashing events are identified based on the concept

of framing which originates in the communications research domain. Thereafter business and data mining objectives and success criteria of the thesis are established.

Figure 1: CRISP-DM framework (Azevedo & Santos, 2008)

1.6.2 Phase 2: Data understanding

First the initial data needs to be collected from Twitter, then a description of the initial data is prepared that describes the properties of the data. This Twitter data is extracted from the Twitter advanced search pages by using a web crawler that extracts the necessary information and the Tweets from these web pages. The Tweets are gathered before (2009), during (2010) and after the Deepwater Horizon oil spill (2010-2014). A pre-labeled Twitter Tweet corpus made by Niek Sanders (Sanders, 2011) is acquired and will be used for the first modeling phases to build a well-functioning classification model and to compare results to the manually labeled BP dataset. Also, other data is gathered to support the analysis of the influence of greenwashing practices such as the stock prices of BP. After acquiring the data it is explored and the data is verified to make sure that the quality of the data is good enough.

1.6.3 Phase 3: Data preparation

In this phase, the relevant data is included and any irrelevant data is excluded based on the data report from the previous phase. Thereafter the data is cleaned to reduce any invaluable information that consists within the data. This data cleaning consists of several actions such as: removing duplicate Tweets, converting slang, stop word removal and other pre-processing steps (see chapter 5.1 Data preprocessing). After cleaning the data is stored in the right format for further analysis.

1.6.4 Phase 4: Modeling

The first classification model that is being built is based on the data retrieved towards BP and the pre-labeled twitter corpus (Sanders, 2011), the data of the model is based on the first pre-processing steps and uses a Naïve Bayes algorithm to classify the Tweets. This algorithm is scalable and is useful for small and large datasets, it is a simple algorithm that is very efficient and is known to outperform highly sophisticated and complex algorithms (Chen, Huang, Tian, & Qu, 2009; H. Zhang, 2004). First an initial classification model is built, this model is then assessed and any action required to fine-tune

the model is taken. These actions include the evaluation of the model and taking necessary steps to improve the model or try new models when previous models are insufficient.

1.6.5 Phase 5: Evaluation

In this phase the outcomes of the model are evaluated based on the precision, recall and F1 score of the model and the data mining process are evaluated. The evaluation leads to changes and recommendations for the iteration step of the process.

1.6.6 Phase 6: Iteration

After the evaluation of the classification first model improved models are built, these models try to improve the classification task based on the evaluation of the previous phase. Also new pre-processing steps are executed when deemed necessary. These new models are tested BP Twitter data.

1.6.7 Phase 7: Deployment

The knowledge that stems from the model built is organized and analyzed, it is elaborated on and presented. An analysis is made of the greenwashing events that have been identified and the sentiment trend that is developed by the best performing model. The insights and conclusions stemming from the comparison of the sentiment trend and the greenwashing events are presented in this phase.

The research design of the thesis is shown in Figure 2: Thesis research design.

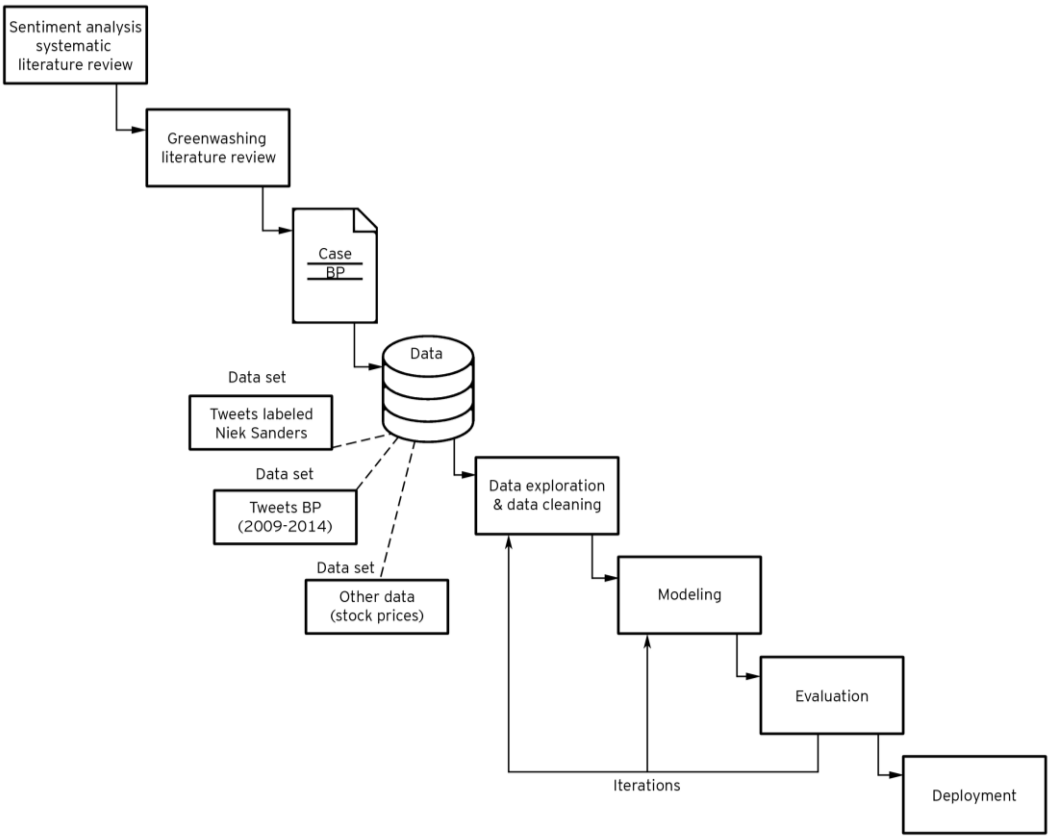


Figure 2: Thesis research design

1.7 Thesis outline

The structure of the thesis is as follows: first the summary of the systematic literature review regarding sentiment analysis is given thereafter the greenwashing literature review is presented in the same chapter. Subsequently the case regarding BP's greenwashing practices is made and elaborated on. Thereafter a description of the data, data gathering, cleaning and analysis is given. Then the model is evaluated and iterated on. At last the results are elaborated on and an analysis of the sentiment trend, greenwashing events and other data is made. Based on these results a discussion is presented.

2 Chapter: Theoretical Background

To gain a better understanding of the available research done towards greenwashing and sentiment analysis two literature studies are conducted. The systematic literature review regarding sentiment analysis has been conducted before the start of the thesis project to gain a better understanding of the techniques available for sentiment analysis and its application domains. The systematic literature review is only shortly summarized. Thereafter a literature review towards greenwashing is executed to improve the knowledge of the concepts and antecedents of greenwashing, also knowledge about previous research done towards greenwashing is extracted from this literature review. These literature reviews shed light on sub-research questions one: how greenwashing events can be identified, two: what data and attributes are needed for sentiment analysis, and three: how a ground-truth can be established.

2.1 Summary sentiment analysis systematic literature review

Due to the limited space of the thesis document only a short summary of the findings of the systematic literature is stated, for the whole systematic literature review please see Appendix 8.1 Systematic literature review. The following databases have been used to search for literature: ACM digital library, IEEE-Xplore, Springer and Web of science. In total 53 papers are retrieved, this amount is reduced to 16 papers that are thoroughly analyzed. This review uses articles from the last 5 years due to quick developments in the sentiment analysis field and the vast amount of literature on this topic that is available. The literature review analyzes common components from the 16 articles that made the cut to the literature shortlist.

There is a wide variety of methods used to classify sentiment, but all are based on three approaches: Machine learning, Lexicon and hybrid approaches. Some of these methods provide very good accuracies ranging from 85%-95% and some none at all. The main data sources that are used within the article pool are product reviews, movie and video reviews and Tweets. To increase classifying results there is often made use of data pre-processing, the main pre-processing steps observed are: 1. preparing data before cleaning which includes tokenization and Part-of-Speech (POS) tagging, 2. correcting words, 3. simplifying words which consist of stemming and lemmatization and last but not least 4. data cleaning where features are selected, stop words are removed, punctuation is erased and URL's and hashtags are removed. The papers state that sentiment can be extracted on three different levels (document, sentence and aspect), within the articles analyzed sentence and aspect level are the most used forms of sentiment analysis. The ground truth that is used for classifying depends on the type of input data used. For movie and product review data there is mainly chosen to use assigned "star-ratings" as sentiment label. At other data where there is no rating given there is a corpus used with assigned labels, these labels can be labeled manually or a pre-labeled corpus is used.

Clusters in the articles are defined and presented in an analysis table and this shows that there are some similarities between the analyzed papers. The highest accuracies are achieved by aspect-based sentiment analysis and these all include movie and video review data, these papers mainly use more extensive data pre-processing which partly explains the higher accuracies. From all the articles that could not provide a performance, two third used KNIME as sentiment analysis tool, also the approaches used are all (partly) lexicon based. For every article that uses Twitter data as input there is a sentence-level sentiment analysis approach used. Although this approach is adopted it does mostly not achieve high accuracy, this might also be due to the low usage of pre-processing steps. The last cluster shows the lowest achieved accuracies within the article pool and all these sentiment methods are based on pure machine-learning.

2.2 Greenwashing literature review

The greenwashing literature review aims to shed light on the topic of greenwashing and how events can be identified that make use of greenwashing. Also, the aim of this literature review is to improve the knowledge on the concepts and antecedents of greenwashing, furthermore it sheds light on former research conducted that relate greenwashing to BP and its Deepwater Horizon oil spill. The literature review starts with reading a highly cited paper, published in a high impact journal about the subject of greenwashing: social accountability and corporate greenwashing (Laufer, 2003). From this paper several topics and variables are extracted which are then used to guide the literature search. Although this paper is not a recent one, it is still being highly cited and thus considered still relevant. Based on the subjects and concepts stated in this paper, more literature is sought via the snowballing method and looking at forward and backward citations (Lecy & Beatty, 2012). First the references are screened based on their title and any interesting literature is accessed. Based on reading the introductions and conclusions of these papers there is chosen to include or exclude papers from the analysis. The inclusion of papers is based on the relevance of the research towards greenwashing as a concept, the relevance of the research towards the antecedents of greenwashing or the relevance of greenwashing in relation to the oil & gas sector or BP in particular.

Laufer (2003) states that there is a dramatic rise in the number of socially screened mutual funds and indexes such as the: Dow Jones Sustainability Index (DJSI), FTSE4Good Index, etc. There is clearly a demand for independent reviewers of companies and their sustainability efforts. This is mainly due to the practices applied by companies that are termed as "greenwashing", "bluewashing", and other forms of disinformation from companies that seek to repair their public image and relations. Many corporations publish non-financial reports, but few address the full range of social issues needed to adequately assess the corporation's behavior (Lydenberg, 2002). According to the International Organization for Standardization (ISO) it's difficult for shareholders, investors and other interested parties to make meaningful assessments about the Corporate Social Responsibility (CSR) practices of a company (ISO 2002). Laufer notes that there are significant omissions of information concerning ethical issues in these reports and that simply relying on the integrity of corporate representations seems increasingly naïve. Social and environmental disclosures are generally made for strategic reasons that have almost nothing to do with the obligations and responsibilities of a company. These strategies are aimed at: gaining or extending legitimacy, maintain its legitimacy level or to repair/defend lost and threatened legitimacy. In reality the words stated in the non-financial reports are decoupled from the organization's operational code, it is nothing more than a convincing façade designed to hide the backstage activities from the public's eyes (Milne & Patten, 2002). "Greenwashing" is a strategy that is used to shift the focus and attention away from the firm's actual actions, to create confusion, undermine credibility, criticize viable alternatives, deceptively posture firm objectives and commitments/accomplishments.

Laufer adapts a framework from Beder et al. (1998) that shows the elements of greenwashing, see Table 1: Elements of greenwashing on the next page. This figure shows that according to Laufer greenwashing depends on three elements of deceptions namely: Confusion, fronting and posturing. Laufer identifies two ways of using these elements: internally and externally. The internal confusion stems from the complex nature of a company and its corporate form, external confusion relates to document control and the strict limit of the information flow made available to the public and regulators. Internal fronting can be accomplished through representations of retained counsel, ethics officers, compliance officers and ethics committees. External fronting is realized by the use of scapegoating of subordinates and reverse whistleblowing. Internal posturing is the convincing of internal customers (as much as external stakeholders) that the company has a collective commitment to ethics. External posturing is achieved through actively using a public affairs department and the retention of an outside public relations firm. Interesting to note is that the very firms that wash their reputation through public relations, complex front coalitions and fore-front fighting against, for instance, global warming remain some of the worst corporate offenders.

Laufer argues that at the heart of the practical debate over corporate social accountability are questions of regulations. He argues that social and environmental disclosure should not be voluntary but be guided by external auditors. Although there have been some initiatives to increase the regulations for social and environmental reporting such as the Global Reporting Initiative (GRI) Laufer argues that companies use the GRI, but fail to use external auditing to validate their reports. A mere 29% of the fortune 250 companies use external auditing for their social and environmental reports. Concerns are raised about the external auditing practices as auditors use widely differing approaches and they are verified by the same financial auditor institutions, not by social accounting specialists. Laufer calls out for corporations to fully disclose their actual data on social and environmental impact instead of hiding it behind complex and carefully prepared reports.

Confusion	Fronting	Posturing	Examples
	Cast doubt on the severity of the problem	Employ "front groups" or coalitions of firms to oppose solution or legislation	Promote image that assumes ethical leadership in the field
	Disclose or publish exaggerated claims	Employ "front groups" or coalitions of firms to support solution or legislation	Unveil projects that have negligible value but appear on surface to be significant
	Emphasize uncertainty associated with problem or accusation	Use front group to promote moderate "middle ground" positions	Promote image of a committed corporate culture
	Acknowledge problem by questioning available solutions	Use data to suggest that front groups enjoy widespread public support	Publicly align firm with NGOs that are sympathetic to cause or issue
	Rebrand to avoid past associations; use image advertising to suggest a "green" association	Employ front groups to examine, define, and redefine industrial standards	Publicly align firm with NGOs that offer certification, accreditation, or award without provisions for accountability or verification Counter threats to an organization's legitimacy Seek publicity for recognition from or membership in an "ethics" organization

Table 1: Elements of greenwashing (Laufer, 2003)

From this paper several subjects are defined that need further elaboration: Greenwashing, Corporate Social Responsibility (CSR) and antecedents of greenwashing. The next paragraphs will explain these concepts and will also focus on greenwashing practices that occur specifically in the oil and gas sector.

2.2.1 Greenwashing

Greenwashing is not a recent phenomenon according to (Dahl, 2010). It has been around since mid-1980 and currently it has gained broad recognition and acceptance in society. Dahl (2010) defines greenwashing as the unwarranted or overblown claims of sustainability or environmental friendliness in an attempt to gain market share. Dahl partly agrees with Laufer on the definition of greenwashing. They both agree that greenwashing is about intentionally making environmental claims that are not

true and that this is intended to be for the benefit of the company. Dahl's definition does not include the creation of confusion, other researchers also include these terms in their definition. Mitchell & Ramey (2011) define greenwashing as "a deliberate act by an organization to obscure potentially harmful information or deliver information in a way that portrays a false image that the organization is green or eco-friendly". This definition does propose the creation of confusion and the deliberate act of misstating firm objectives and accomplishments. Other articles have similar definitions that imply the creation of confusion and the deceptive stating of information (Aggarwal & Kadyan, 2014; Furlow, 2010). It can be said that firms engage in two behaviors simultaneously: poor environmental performance and positive communication about their environmental impact (Delmas & Burbano, 2011).

There are seven sins of greenwashing identified (TerraChoice, 2010) and these are mentioned by Aggarwal & Kadyan (2014), Dahl (2010) and Mitchell & Ramey (2011). These seven sins are stated in *Table 2: The seven sins of greenwashing (TerraChoice, 2010)*

Sin	Description
1. Hidden trade-off	committed by suggesting a product is "green" based on an unreasonably narrow set of attributes without attention to other important environmental issues
2. No proof	committed by an environmental claim that cannot be substantiated by easily accessible supporting information or by a reliable third-party certification. Common examples are tissue products that claim various percentages of post-consumer recycled content without providing any evidence.
3. Vagueness	Committed by every claim that is so poorly defined or broad that its real meaning is likely to be misunderstood by the consumer. "all-natural" is an example. arsenic, uranium, mercury, and formaldehyde are all naturally occurring, and poisonous. "all natural" isn't necessarily "green".
4. Irrelevance	committed by making an environmental claim that may be truthful but is unimportant or unhelpful for consumers seeking environmentally preferable products. "CFC-free" is a common example, since it is a frequent claim despite the fact that CFCs are banned by law.
5. Lesser than two evils	committed by claims that may be true within the product category, but that risk distracting the consumer from the greater environmental impacts of the category as a whole.
6. Fibbing	committed by making environmental claims that are simply false.
7. Worshipping false labels	committed by a product that, through either words or images, gives the impression of third-party endorsement where no such endorsement actually exists; fake labels, in other words.

Table 2: The seven sins of greenwashing (TerraChoice, 2010)

These seven sins as identified contribute to the earlier statement of Laufer (2003) that greenwashing is about creating confusion and deceptive statements. These seven sins make it hard for consumers to accurately assess the actual sustainability of a company or its product(s). The downside as stated in these papers is that through the application of greenwashing practices consumers have become

suspicious about any company that makes green claims (Aggarwal & Kadyan, 2014; Dahl, 2010; Furlow, 2010; L. D. Mitchell & Ramey, 2011).

There is also a distinction made between "hard-greenwashing" and "light-greenwashing" (Aggarwal & Kadyan, 2014) where hard-greenwashing refers to environmental communication and claims without any form of CSR in place, and light-greenwashing refers to reducing CSR and focusing more on advertising green claims. Vauday (2013) distinguishes more than two forms of greenwashing, Three types of greenwashing are considered: "hard greenwashing", "constrained light greenwashing" and "ad light greenwashing". In this paper hard greenwashing is defined as acting green without spending a single dollar on actual CSR. Constrained light greenwashing refers to greenwashing practices that stem from the presence of other greenwashing activities, and ad light greenwashing occurs when the firm can communicate a non-verifiable CSR claim. These two papers agree on the definition of hard greenwashing, also the light greenwashing definition appears to be closely linked. Vauday (2013) adds the notion of incentive to use light greenwashing in comparison to Aggarwal & Kadyan (2014). Besides the forms of seriousness of greenwashing there are also distinctions made between product level greenwashing and firm-level greenwashing, where product level greenwashing specifically applies to products and firm-level greenwashing applies to the image of a company (Delmas & Burbano, 2011).

In general it can be stated that greenwashing refers to causing confusion regarding the actual sustainability performance of a company or product to make it look more sustainable.

2.2.2 Corporate social responsibility

Greenwashing relates to Corporate Social Responsibility (CSR) as the CSR approach adopted by a company can determine if a company would use greenwashing as a strategy to enhance its reputation. A general and consistent definition for CSR is not yet apparent (McWilliams, Siegel, & Wright, 2006). McWilliams et al. (2006) have gathered the multiple definitions that are used to define what CSR is and conclude that there are many, which make it hard to compare results of research about the effect of CSR on profitability or firm performance. The most commonly used term of CSR is "The situation where the firm goes beyond compliances and engages in actions that appear to further some social good, beyond the interests of the firm and that which is required by law" (Mcwilliams & Siegel, 2001). They also show that there are multiple perspectives that motivate or discourage the use of CSR within a company, examples of these theories are: Stakeholder theory, Stewardship theory, Transformational leadership and institutional theory. Stakeholder theory suggests that it can be beneficial for firms to conduct CSR because non-financial stakeholders may view them as important, if not performed this group of stakeholders might withdraw their support towards a company. Stewardship theory shows that there is a moral incentive that managers have to "do what is right" without looking at the impact on financial performance. The institutional theory implies that the norms and regulations within an institution define the level of CSR within a firm. Besides these theories there are also mechanisms related to these views that explain why companies invest in CSR. The most common mechanisms being to increase the firm's reputation or to avoid the loss of reputation due to the threat of Non-Governmental Organizations (NGO's) (Vauday, 2013). Other factors to employ CSR are the ability to charge a price-premium and the use of CSR to recruit and retain high-quality workers (Siegel & Vitaliano, 2007). The employment of CSR relates to strategic implications for the firm, which in term depends on how CSR is viewed. CSR can be viewed from three perspectives: as altruism, coerced egoism and strategic CSR (Husted & De Jesus Salazar, 2006). Altruism refers to the sincere motivation of being socially responsible, coerced egoism relates to firms that act in a socially responsible way only when regulations or laws force them to do so. Strategic use of CSR occurs when firms see clear benefits for their firm in employing CSR. This last view is also termed as "profit-maximization" (Siegel & Vitaliano, 2007). Siegel & Vitaliano (2007) note that how firms view CSR affects the strategies they apply towards bringing their CSR efforts to the public, this shows in the way companies advertise. Advertising can be used to try to influence the consumers taste towards a product or company, it can also be used to merely inform the public. Besides

advertising there are also other strategic uses of CSR which are influenced by the firm's view towards CSR. Consider a firm that uses CSR to gain competitive advantage, firms might then use their CSR in political strategies to impose regulatory barriers to imitation. These firms might try to persuade the government to impose CSR efforts also on competitors that do not adopt a CSR view to raise costs for them.

With CSR efforts also comes reporting. Interesting to note is that the analysis of annual reports made regarding CSR is considered to be more a "social disclosure" than CSR per se (Vauday, 2013). One can also view this activity as advertising, the information in these reports can be seen as biased because these reports are filtered through senior management (Siegel & Vitaliano, 2007). These reports are not always considered to be objective because there is a relation established between the disconnection of actual CSR and its communication to the public (Aggarwal & Kadyan, 2014).

2.2.3 Antecedents of greenwashing

The antecedents of greenwashing can be classified into different drivers: market drivers, non-market drivers, organizational drivers and individual drivers (Delmas & Burbano, 2011). Delmas & Burbano (2011) propose an exhaustive framework that shows the relationship between the drivers of greenwashing, as shown in Figure 3: drivers of greenwashing (Delmas & Burbano, 2011).

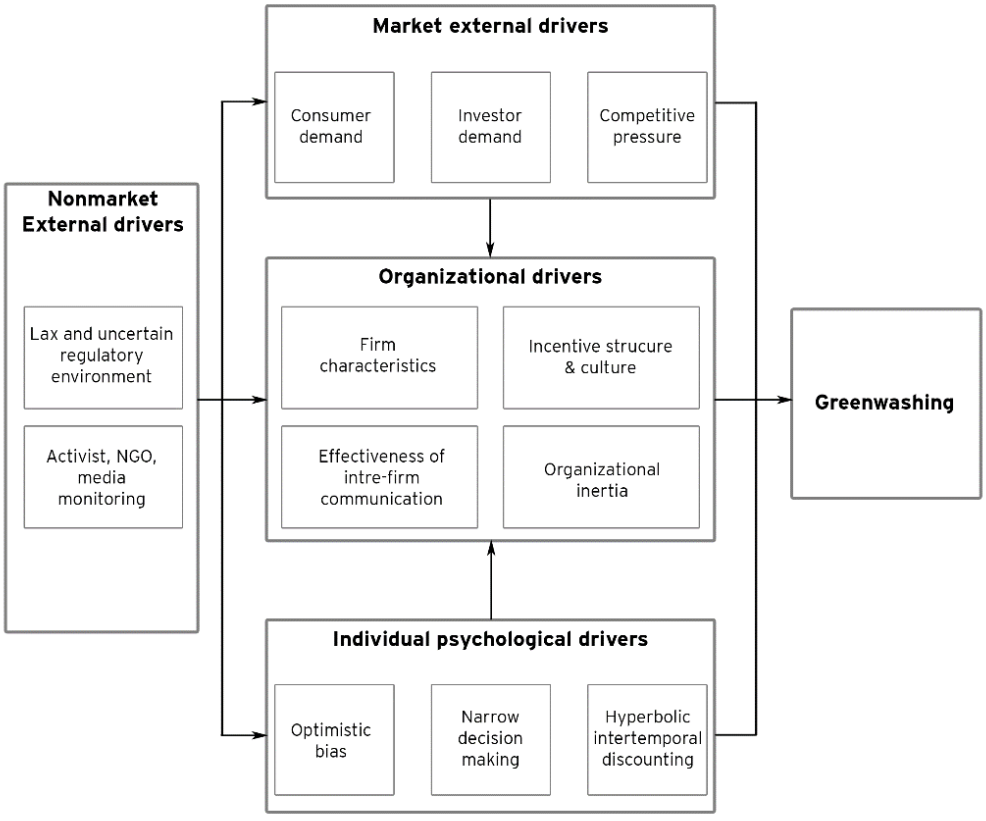


Figure 3: drivers of greenwashing (Delmas & Burbano, 2011)

Delmas and Burbano describe several factors that influence the decision made to apply greenwashing for a firm itself or its products. Figure 3 provides the overview of all those factors, next the main categories will be explained. Nonmarket external drivers: The first and most important factor is the absence of adequate regulations regarding environmental claims (especially on a firm level). Although this article is primarily focused on the United States they also state the same problem for European and other countries. This statement is backed by several other papers that state the same problem as

the main driver for greenwashing (Aggarwal & Kadyan, 2014; Barrage et al., 2014; Oppolecchia, 2010). In addition, it is not mandatory for corporations to disclose their environmental practices (there are a few exceptions such as toxic releases). Another driver is the pressure applied by NGO's and activists to push companies to "go green", which deters some positive communication from bad performing firms, but the lack of regulations make it hard to enforce anything. Meaning that NGO's and other parties can only bring some reputational damage which diminishes their threat to companies. Market external drivers: More prevalent pressure comes from consumers and stakeholder which are looking for green products and investments, firms feel the need to communicate better environmental performance than currently in place (Vos, 2009). The same reasoning applies to pressure that stems from the competitive landscape. Companies need to be perceived as legitimate and successful and fear falling behind to their rivals (Delmas & Toffel, 2008). Organizational drivers: firm characteristics such as their culture and inertia influence the overall strategies that a firm can apply (also greenwashing) and the costs and benefits associated with a particular action. For instance publicly traded firms face greater levels of investor pressure than small private firms, thus strategies might be influenced. Individual psychological drivers recognize the roles of individuals and leaders in a firm's behavior. Tendencies such as narrow decision framing and optimistic bias for instance influence decisions made to apply greenwashing (Kahneman, 2003). In general it can be stated that the lack of information available about firm environmental performance contributes to greenwashing in multiple ways: stakeholders cannot hold a firm accountable because there is a lack of readily available information; managers and consumers cannot properly assess a firms environmental performance; managers and consumers cannot make justified decisions regarding environmental performance of companies (Delmas & Burbano, 2011). This contributes to the high level of applied greenwashing practices in general.

2.2.4 Greenwashing in the oil & gas sector

Greenwashing in the oil and gas industry is no new phenomenon (M. Cherry & Sneirson, 2011). Studies are found that research greenwashing practices for BP (Barrage et al., 2014; M. Cherry & Sneirson, 2011), ExxonMobil (Plec & Pettenger, 2012) and Chevron (M. A. Cherry & Sneirson, 2012). In the case of Chevron, Cherry & Sneirson (2012) show that greenwashing is applied by the means of advertising campaigns after the BP oil spill in 2010. The same order of events, first an oil spill then greenwashing advertisements airing on TV is identified for the ExxonMobil case (Plec & Pettenger, 2012). Antecedents mentioned by Cherry & Sneirson (2012) regarding Chevron consists of their fight against claims made towards Chevron for polluting the Amazon rainforest in Ecuador. Also the increased pressure of consumers and investors for socially responsible behavior increases pressure to adopt greenwashing practices. The act of greenwashing resides on Chevron's website and advertising suggest ethical behavior, but their actions depict otherwise. The events of greenwashing in this paper are established based on the advertisements shown on TV and the factual data about its operations. These two sources of information are inconsistent with one another and thus greenwashing is present. The same earlier mentioned pressure from consumers and investors is apparent in the ExxonMobil case. Plec & Pettenger (2012) argue that two trends have emerged from this increased pressure. Trend one focusses on the production, distribution and promotion of eco-friendly alternatives to existing products, the second trend involves corporate advertising campaigns that use greenwashed environmental frames to promote products or companies that are not sustainable or eco-friendly. Plec & Pettenger (2012) use another way to prove the acts of greenwashing. They use the concept of framing, which stems from social and behavioral studies, to analyze the information and view that is presented to the public in relation to the bigger picture of information that should be expressed. Plec & Pettenger (2012) conclude that ExxonMobil indeed used greenwashing by supplying the public an incomplete and incorrect image of the whole truth by applying a didactic frame full of technocratic and authoritarian values. This incomplete truth is used to position ExxonMobil and its actions as greener than their actual actions, and it discourages the public to engage in environmental activism. The cases that investigate greenwashing at BP are both about BP's Beyond Petroleum campaign before the oil spill. Its advertisement campaign was so successful that BP went from a middle-of-the-

pack oil company to the greenest oil company in the minds of the public (Barrage et al., 2014; M. Cherry & Sneirson, 2011). Cherry & Sneirson (2011) show that while BP's advertising campaign was very successful, the oxymoron of an environmentally friendly oil company that engages in deepwater drilling did not reach the public's mind. Even so it seems that BP struck a nerve with the American public, not only consumers but also legal academic commentators, environmentalists and CSR watchdogs fell for the advertising campaign (M. Cherry & Sneirson, 2011). This clearly showed when BP's stock price doubled. In reality though, the company ignored workplace safety regulations and ignored environmental laws for more oil and profit (Lustgarten & Knutson, 2010). Barrage, Chyn, & Hastings (2014) use data from gas stations, advertising and green preference to measure the influence of the Beyond petroleum campaign on the punishment given by the public after the Deepwater oil spill and the public attitude towards BP. They show that this advertisement campaign dampened the punishment given by the public and that consumers attributed the oil spill more towards bad luck than negligent practices.

2.2.5 Conclusion

From the literature studied it can be stated that the main effect of greenwashing is to cause confusion about the real CSR performance of a company. This can happen in multiple ways and on multiple levels but the end result remains the same. The framework presented by (Delmas & Burbano, 2011) shows the antecedents of greenwashing, several of these drivers are mentioned in the papers read, others not, this is attributable to the focus of the literature study. Psychological and organizational drivers are mostly not apparent in the papers read due to their focus on industrial applications and the effects of greenwashing. It appears that the main drivers for applying greenwashing practices in the read papers are: the lack of legislation towards greenwashing and the increased pressure from consumers and investors towards companies to go green. These findings apply to the oil and gas sector as well, it is found that several oil and gas companies make use of these practices to improve their corporate image. In this sector advertising is mainly the medium subject to greenwashing practices. Several methods to identify greenwashing practices are found in the literature review. Factual operations data can be used (if available) in relation to statements made by the company to identify greenwashing, another way is to use the concept of framing. Framing looks at the picture or view that is presented to the public and compares it to the bigger picture of available context and information.

3 Chapter: Case BP greenwashing

Before a sentiment trend can be constructed the greenwashing events used by BP need to be identified. Previous research towards BP's greenwashing practices have analyzed the effects of BP's Beyond Petroleum campaign (Barrage et al., 2014; M. Cherry & Sneirson, 2011), to my knowledge there are no studies that focus on or analyze the effects of BP's greenwashing after the Deepwater Horizon oil spill. To identify BP's greenwashing events there is drawn on the concept of framing, also applied by Plec & Pettenger (2012) to identify the greenwashing frames that ExxonMobil. First a brief explanation of framing is presented by elaborating on key papers in this area and the papers used in Plec & Pettenger (2012) for their frame analysis. Subsequently the advertisements aired on American TV by BP after the oil spill will be studied by using a frame analysis to reconstruct the mental frames invoked by BP on the public.

3.1 Framing

Framing stems from psychology and sociology. It describes the concept of how people (and organizations) use information and representation features in communication to address issues, and how this forms impressions in the minds of the audience (Scheufele & Tewksbury, 2007). The analysis of framing can show the influence of the communicated message from the communicator on the audience's conscious (Entman, 1993). Entman (1993) argues that the function of framing is to select and highlight elements of information to construct an argument about a certain problem, its causation, evaluation and its solution. Framing thus highlights certain parts of information to elevate them in salience. The highlighted information becomes more noticeable, meaningful and memorable in the minds of the audience (Entman, 1993). This, in turn, determines the evaluation and actions that the audience undertakes towards the provided problem. What results from highlighting only certain parts of information is that the attention is directed away from other information, meaning that power is exerted to show the audience what is really important from the communicator's point of view. This is especially troublesome when the audience is not informed well about a certain topic because framing then majorly influences the opinion and cognitive activities towards the presented topic (Scheufele & Tewksbury, 2007). Frames thus serve as devices that highlight certain features of a circumstance and direct our attention towards a specific form of action (Plec & Pettenger, 2012).

The same approach to frame analysis is adopted as used in Plec & Pettenger (2012). Several concepts are used in the paper to conduct their frame analysis: Social and natural frameworks (Goffman, 1974), substantive features and foci of frames (Entman, 2004) and poetic categories (Burke, 1984). Natural frameworks identify occurrences in their pure physical form and are undirected, consider for instance the weather as stated in a report. In contrast to natural frameworks are social frameworks. These frameworks incorporate controlling efforts of a party and provide the audience with a controlled will and aim. Social frameworks thus provide a way of describing events with a directed motive, goal and intent. Entman (2004) makes a distinction between the focus and the functioning of frames. A framing focus consists of issues, events and actors presented. The functions of a frame can be divided into four primary functions: (1) defining effects or conditions as problematic, (2) identifying causes, (3) conveying moral judgment and (4) endorsing remedies or improvements. Poetic categories are verbal acts applied by a party, Burke (1984) defines it as the confrontation of a subject and how this confrontation is presented in poetic and conceptual imagery that helps the audience to accept the confrontation presented. These three parts together (the framework, focus and functioning of the framework and its poetic category) present the full picture of how a problem or event is presented to the public and how it is painted "Green".

3.2 Identifying greenwashing practices

The advertisements that are aired on American TV are inventoried by using data available from Ispot (Ispot, 2017), an article from the Huffington Post (Huffington Post, 2013) that summarizes BP's advertisement efforts since the oil spill and BP's YouTube channel. Ispot is a website that keeps track of advertisements, analyzes the content and serves as a database. This thesis focuses solely on TV advertisements that are directed towards the Deepwater Horizon Oil spill and have aired in the United States. Only TV commercials are considered, this information is readily available and can be accessed without paying large subscription fees to companies that also store newspaper and other media advertisements. In total there are 13 TV commercials identified starting from October 2010 till May 2013, these commercials aired after the Deepwater Horizon oil spill in April 2010. Only commercials that concern the Gulf of Mexico region and the Deepwater Horizon oil spill are selected for analysis, for the full list of commercials see Appendix 8.2 TV commercials aired.

3.3 Frame analysis

There are two dominant discourses that BP has put forward in TV advertisements after the Deepwater Horizon oil spill. The first one concerns their TV advertisements that represent local voices from inhabitants of the Gulf of Mexico in a series called: voices from the Gulf. The second concerns BP's commitment to restore the Gulf area and America. In a broad perspective BP tries to position the oil spill and its effects on the gulf area as less severe than in reality, also it tries to emphasize their "good deeds". The analysis of the framing applied by BP will show that BP poses a green view towards the oil spill and its impact.

BP advertises in three different ways after the Deepwater Horizon oil spill, it advertises with a series called Voices of the Gulf and BP's commitment series. Within the voices of the Gulf series there are two approaches adopted. The first approach presents inhabitants of the gulf that run a business and that have been impacted by the oil spill. It shows that the oil spill has impacted the inhabitants, subsequently it shows that BP kept their promise to make it right again and that there are no losses or consequences for them in the end. The second approach presents inhabitants and business owners from the Gulf area (Mississippi, Louisiana, Florida and Alabama) and shows that there is absolutely no reason not to visit the Gulf area because tourism has been thriving. The third and final approach shows BP's commitment to the gulf by spending billions of dollars on recovery practices and millions on a research fund, they are proud to announce that beaches, waters and tourism have recovered. It also poses that BP is committed to America by being its largest energy investor and sharing knowledge extracted from the oil spill. BP has actively used a greenwashed frame that highlights the beliefs that the gulf area is not much affected by the oil spill, beaches and waters are clean and that BP is committed to clean up and recover the negative effects of the oil spill.

3.3.1 Voices of the Gulf: business owner's impact

In the first approach of advertisement BP is putting forward an episodic frame which simplifies a complex issue into a specific example and emphasizes the actions taken by BP in reducing the negative effects stemming from the oil spill. BP selectively highlights positive facts and paints a different more positive picture than the actual status of the oil spill. Episodic frames are used to emotionally engage the audience and to make the story more compelling. Complex concepts and issues are divided into simpler terms, certain aspects are emphasized and other aspects are left out. This "cherry picking" paints a different picture from one that would be derived from a more balanced and elaborated overview (Gross, 2008). The advertisements in this series draw upon business owners that have their business in the Gulf of Mexico. What is told is not the whole story about the impact of the oil spill on business owners in the Gulf of Mexico, but a story about a specific business which BP helped recover from the oil spill. It features frames that convey moral judgment with a strong focus on the actors (business owners) and the presented issue (impact of the oil spill). The social frameworks applied in

these advertisements positions BP as a voluntary "good-doer" while the oil spill is barely mentioned. The episodic frame applied to these commercials tries to enhance the reputation of BP by showing that it is helping affected business owners after the oil spill and highlighting their commitment to made promises. The ways in which the episodic frame is presented is the same for the whole series, only one advertisement (Louisiana Restaurant Owners – October 24, 2010) analysis is presented here as it applies to all 5 commercials in this series.

The most apparent way in which this episodic frame is presented is through the introduction of Bryan and Brooke Zarr, business owners of a local seafood restaurant near the Gulf of Mexico. The advertisement starts with restaurant employees working in the restaurant accompanied by slightly sad piano music. Bryan Zarr is telling the audience that he worked in this restaurant when he was younger and that he bought the restaurant last year with his wife. The slightly sad music intensifies when Brooke Zarr then tells that they have worked really hard to build their business and then came the oil spill. Subsequently there is shot of a sad looking Bryan Zarr in an empty kitchen which is followed by the restaurant being filmed with a time-lapse feature that shows almost no one entering the restaurant, leaving a sad image in the minds of the audience. The following is a close-up shot signifying intimacy and a personal relationship, that shows the slightly sad looking restaurant owners in an empty restaurant. They are stating that without customers they could close up, the slightly sad music fades out intensifying the moment. After a moment of silence uplifting music starts to play and the restaurant owners appear on screen together with a slight smile stating that they did not close up. They tell the audience that instead of serving customers they served BP by feeding the workers that were cleaning up the oil spill, all accompanied by a video that shows the business owners working to produce food for BP. Bryan and Brooke tell that they had to put the whole family to work to make it happen, a shot of the whole family working to prepare the worker's food is introduced pointing to the family bonds that are being tightened because of the oil spill. Thereafter the audience is told by the restaurant owners that they were not the only ones, that BP worked with all kinds of small business owners to keep businesses open and working. While the video shows work in progress in the restaurant, Bryan tells the audience that BP said to them that they would make it right. Brooke then accompanies Bryan by telling that it has been tough, but they are still cooking. While the uplifting music has intensified Bryan and Brooke introduce themselves to the audience and share the story that BP asked them to tell their story to the public. In the bottom of the screen is a footer visible that shows BP's logo and a phone number of a helpdesk hotline for further questions. Bryan and Brook's story continues by telling that they are now back and open for business, they have local seafood back on their menu and that it is going to be a good time to come to the Gulf coast. While the uplifting music reaches its conclusion, Brooke Zarr informs the audience that they would love to have the audience over at their restaurant insinuating that everything is as good as it ever was in the Gulf area.

BP employs a greenwashed frame in their advertisement series voices of the Gulf. BP invokes feelings of empathy for the business owners in the advertisements, BP also shows their effort in helping these business owners to the extent that it helps these business owners get back on their feet. BP completely sidesteps information that would provide a bigger picture of the situation. BP is forced by law to compensate for all losses made because of the oil spill (Brennan, 2013), while BP insinuates that it is willingly helping these business owners. The advertisements show "cherry picking" by highlighting some business owners that were helped by BP, many others were misinformed by BP about their legal rights and had unknowingly waivered their right to file a lawsuit by settling for a compensation from their fund (Goguen, 2015; Webb, 2011). Furthermore, BP sells the image that the Gulf is open for business again and that the impact of the oil spill is not severe. The impact of the oil spill is hard to estimate but one thing is for sure, the impact is severe and long-term effects are uncertain (Beyer, Trannum, Bakke, Hodson, & Collier, 2016; Mendelssohn et al., 2012).

3.3.2 Voices of the Gulf: tourism

The second approach of advertisement is again putting forward an episodic frame which focusses on the tourism sector of the Gulf area, these advertisements aired mainly in 2011. The advertisements in this series put forward a different spokesperson for the different American states that surround the Gulf. The audience is subject to a "battle" between the states represented over which one is the best to visit. There is no mention of the oil spill in these commercials, the Gulf region is represented as undamaged and open for business which understates the actual status of the gulf. The same framing is used throughout the whole series, see appendix 8.2 Voices of the gulf: tourism. Only one advertisement (voices from the Gulf: Best Place) analysis is presented as the same reasoning fits the other commercials in this series.

The audience is introduced to four actors that represent the American states that surround the Gulf of Mexico. Every actor is positioned in front of a décor that is representing the ambiance of the specific state. First Mississippi is introduced by swinging music and with Rip Daniels representing the Harrison County Gulf Tourism Commission in front of a live music and dancing décor in the evening. Rip is smiling and states that "everybody knows the best place for a good time is Mississippi". Immediately after the audience is introduced to Tom Hymel, owner of Delcambre seafood in Louisiana. Tom stands broadly smiling in front of a fish Wharf accompanied by country music and replies to Rip that "and that's only until they've visited us in Louisiana". Dawn Moliterno comes on screen as the representative for Florida, replying to Tom that Louisiana "is a distant second to sunny Florida". Dawn is positioned smiling on a sunny restaurant terrace near the beach with sports activities going on. The last introduction is Bill Barrick which represents Alabama, he announces that "for a beautiful vacation, nothing beats Alabama". Bill is standing in front of a green, garden-like décor with people enjoying the greenery in the back. The actors are replaced for video shots of people enjoying waterslides, parasailing, entertainment parks etc. While the video plays one of the actors announces that the will never agree on who's best, but they agree that the Gulf is the world's number one vacation spot. The different states are then again showed to the public accompanied by their actor stating the best activities from the state. Mississippi is portrayed by dancing people, musicians playing and fishing activities, Rip states that Mississippi has "wonderful people, great music and the beautiful outdoors". Louisiana takes the screen again and is accompanied by seafood and people enjoying seafood cuisine. A sailboat appears on screen, fading into an aerial view of white beaches while Dawn says that "Florida means beautiful beaches and sugar white sands". Bill then replies to Dawn that "experts agree that the best beaches are in Alabama", while the video shows people enjoying the beach. Again the actors "battle" by stating the best activities from their state accompanied by video images of that activity. While the music plays and they video continuously to show people enjoying themselves the representative of Louisiana tells the audience that "the Gulf is the world's good times headquarters, and we are 100% open for business". The representative of Alabama replies that he is glad that they've got that settled, while the video fades over into an image of BP inviting everyone to visit the Gulf.

Again BP used a greenwashed frame to understate the severity of the Gulf situation and draw attention away from the negative effects by focusing on all good characteristics of the states represented. In this commercial BP invites everyone to visit the Gulf area and that it is business as normal. The advertisements are mainly aired in 2011 and one of the series is aired in 2012, BP implies that there is nothing unworldly going on in the Gulf area and that tourists have no reason not to come over. Wild and marine life are a major part of the Gulf's tourism activities (Nadeau, Kaplan, Sands, Moore, & Goodhue, 2014), research speaks differently than BP by showing that around 2011/2012 bottlenose dolphins in the Gulf died at twice the normal rate (Braun, 2015). Also the nesting habits of sea turtles changed dramatically at this time (Putman et al., 2015) and the coral reefs in the area have been damaged by the explosion and oil spill (The Guardian, 2012). This, for example, paints a different picture of the impact on tourism and its activities than BP does suggest in its advertisements. This also shows in the tourism numbers as tourism had fallen ill in the region after the Deepwater Horizon oil spill, many vacations are canceled and booked elsewhere (OCEANA, 2013).

3.3.3 BP's commitment

The third approach of advertisement used by BP puts forward a didactic frame to teach the audience about the actions that BP has taken and the progress made in the Gulf. This didactic frame functions by asserting that BP does everything to solve the problem of the Deepwater Horizon oil spill and even going beyond that by being committed to America. These advertisements aired in late 2011 and until halfway 2013. This particular series of advertisements put forward one spokesperson that works for BP and informs the audience about the money that BP has spent on cleanup costs, funding and research initiatives. Also the spokesperson shows the actions BP has taken to increase safety on the job. Besides the commitment to the gulf the spokesperson also informs the audience about BP's commitment to America, and what BP does to keep the American economy going. The same framing is used throughout the whole series, see appendix 8.2 BP's commitment. Only one advertisement (BP's Commitment to America) analysis is presented as the same reasoning fits the other commercials in this series.

The didactic frame in these advertisements is presented through the authority and approachability of using a spokesperson (a so-called talking head) (Plec & Pettenger, 2012). The advertisement starts with the picture of a clean ocean accompanied by calm music with a serious tone. A BP spokesman starts telling that two years ago, the people of BP made a commitment to Gulf. The video fades from a clean ocean to the BP spokesman who is standing on a pier near the coast. The screen introduces Fred Lemond, the operations manager at BP. Fred continues his story by telling that every day since we (the people of BP) worked hard to keep it. The video starts showing people windsurfing, which fades into seafood being served in a restaurant while the music starts to gradually lose its serious tone and moves into a more happy tune. Meanwhile, Fred informs that BP has paid over 23 Billion dollars to help people and businesses who are affected and to cover cleanup costs. At the end of this sentence a slide with a BP logo is visible stating the number of dollars spent. After this slide the video turns into people enjoying the beach, diving near marine life, fishing and sailing. Fred announces that the beaches and waters are open for everyone to enjoy, and many areas are reporting their best tourism seasons in years. Fred continues by telling the audience that BP has shared what they have learned with governments and across the industry so they can all produce energy more safely. Accompanying this are videos of people working on Gulf research and simulations with lots of computer monitors in a dark computer room. Fred appears on screen again stating that he would like us to know that there is another commitment that BP takes just as serious: BP's commitment to America. While the video starts showing BP personnel working, Fred says that BP supports nearly 250.000 jobs in communities across the country. A slide with a BP logo is introduced stating the number on the screen. Fred continues by saying that BP just hired 3000 people last year and that BP invests more in America than in any other country. While Fred informs about these figures, several employees of BP are put on screen standing in front of their working environment. The video starts showing engineers in labs, overseers inspecting BP pipelines and ground workers that are drilling for oil. The audio plays Fred's voice that tells: "in fact, over the last five years no other energy company has invested more in the U.S. than BP". Fred appears on screen again and states that they are working to fuel America for generations to come, accompanied by a video of a mother and child near the coastline. A slide of BP is put on the screen stating: "Commitment to the Gulf, commitment to America" and Fred closes by saying: "Today our commitment to the gulf and America has never been stronger".

BP again uses a greenwashed frame by delegating authority to the spokesperson that makes it look like BP is voluntarily paying for their errors, but in fact this is mandatory and forced by law (Brennan, 2013). Again BP states that the gulf beaches and waters are open, which in fact is not untrue but they sidestep the issues of damage made to the coasts, marine life and coral (Braun, 2015; Mendelssohn et al., 2012; Putman et al., 2015; The Guardian, 2012). Fred states that many states report their best tourism season in years, this information is not complete as other media suggest that it is not sure if states most affected by the oil spill are improving as much as other states regarding tourism (Reuters, 2012). BP argues that they are committed to America by investing heavily in the country and hiring many American citizens. This is called a "red herring" fallacy (Hurley, 2011) where Fred tries to

sidetrack the audience. The problem is the oil spill and its impact, Fred uses the commitment to America to engage the audience into believing BP is doing good by investing in America and supplying jobs. This argument does not contribute in any way to the solution to the original problem and is thus considered sidetracking.

3.3.4 Conclusion

All three advertisement series are considered greenwashing practices as they use episodic and didactic frames to persuade the audience into thinking positively about BP while the impact of the oil spill is neglected in these advertisements. Practices adopted by BP in these advertisements can be related to the greenwashing sins from Terrachoice (TerraChoice, 2010). BP mainly uses the sin of hidden trade-off, which markets a product or company as "green" based on a narrow set of attributes while neglecting other important environmental issues. BP mainly states superficial information or focusses on one specific case while neglecting other valuable information. Also the sin of irrelevance is adopted in the last series concerning BP's commitment. BP argues that its commitment to America makes BP a better company though this does not solve anything regarding the original problem of the oil spill. In total there are thus 13 greenwashing advertisements identified, see Figure 4: Greenwashing TV advertisements.

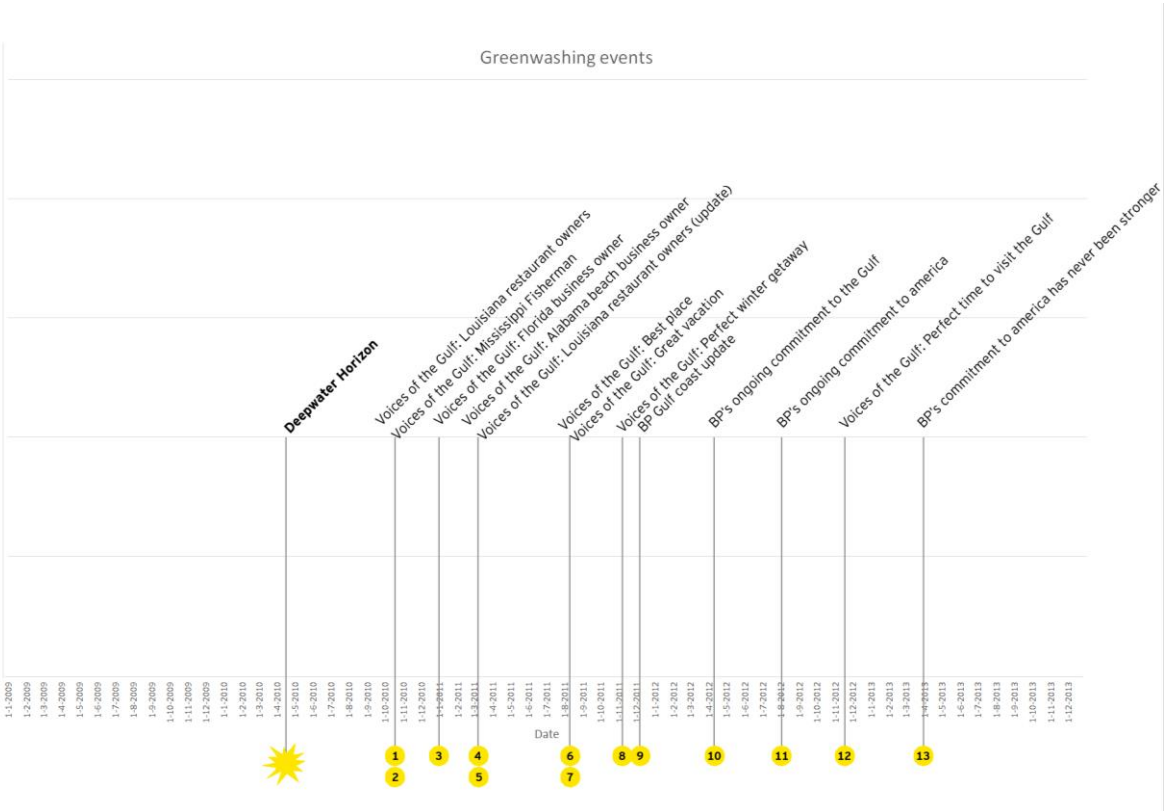


Figure 4: Greenwashing TV advertisements

4 Methodology

In this chapter the methodology that is used to create the actual sentiment trend will be discussed. To make a sentiment trend there first needs to be data, after the data is gathered a model needs to be build that can classify the opinion of Tweets gathered. The goal is to build a supervised-learning classifier that can, to a certain degree, accurately classify Tweets into sentiment categories. From this classification a sentiment trend can be constructed that can be used to measure the influence of greenwashing practices on the public sentiment. First the background for classification is discussed, thereafter the executed models and their structure are presented, subsequently the data acquisition procedure is elaborated on and finally the data is explored before the actual modeling in chapter 5.

4.1 Classification background

There is chosen to use classification as the data mining tool to establish a sentiment trend, classification is a common approach for conducting sentiment analysis (Jurafsky & Martin, 2017). The goal is to first build a relatively simple and common classification model and then extend this model or revise the process when the standard approach is deemed insufficient. Classification takes in observations, extracts features from the observations and then classifies new observations into a set of available classes (Michie et al., 1994). Classification relies mainly on supervised learning, in a supervised learning situation there is a set of training data available. Training data consists of observations that have corresponding class labels that will later be predicted by the classifier for new observations that are parsed through (Jurafsky & Martin, 2017). This training data is fed to the classifier where it learns the combination of classes and the corresponding observations.

4.1.1 Naïve Bayes

There are many classification algorithms available, in this case the starting point is a standard Naïve Bayes model. This classifier is a useful and scalable algorithm that is suitable for small and large datasets, it is a simple probabilistic classifier that is efficient and is known to outperform highly complex and sophisticated algorithms (Chen et al., 2009; H. Zhang, 2004), it is mainly used for text classification problems (McCallum & Nigam, 1998). From papers read the Naïve Bayes algorithm can achieve accuracies around 85% (Bakliwal et al., 2012; McCallum & Nigam, 1998), the expectation is that a Naïve Bayes model can thus be successfully implemented and achieve similar results.

The Naïve Bayes algorithm is a machine learning algorithm for classification problems (B. Liu, Blasch, Chen, Shen, & Chen, 2013) that relies upon a simplistic view of a set of documents as a Bag of Words (BOW) model where all words are represented in arbitrary order without any grammar or word orders (Jurafsky & Martin, 2017). The Naïve Bayes classifier is mainly used for text classification problems (McCallum & Nigam, 1998). The word naïve in refers to the assumption that the algorithm makes, also known as the Naïve Bayes assumption. The algorithm assumes that all the input data is independent of one another, meaning that the occurrence of a certain feature is independent of the occurrence of other features (McCallum & Nigam, 1998).

Bayes theorem is the base of this algorithm where the probability is denoted by $P()$, in this theorem there is a sentiment class denoted by A and a Tweet considered as a set of words B. The probability that the sentiment class is A given Tweet B is written as $P(A|B)$. There is known what the prior probability is of a Tweet belonging to a certain class based on the occurrence of the sentiment class in the data, this can be stated as $P(A)$. Also there is a need to test for the likelihood that a tweet belongs to a class given the tweet itself, this is stated as $P(B|A)$. The total probability of the Tweet occurring is denoted as $P(B)$. See also table Table 3: Notation Naïve Bayes on page 22.

Notation	Description
A	Sentiment class
B	Tweet considered as a set of words
i	Index of a word in a tweet
B_i	Tweet feature i
$P(A)$	Probability of a Tweet belonging to a sentiment class based on occurrence in data
$P(B)$	Probability of Tweet B occuring
$P(A\|B)$	Probability of sentiment class A given Tweet B
$P(B\|A)$	Probability that B belongs to sentiment class A

Table 3: Notation Naïve Bayes

$$P(A|B) = \frac{P(B|A)\ x\ P(A)}{P(B)} \tag{1}$$

In the Bayesian method of probability, $P(A)$ is called the prior probability of "proposition" and $P(B)$ is called the prior probability of "evidence". $P(A|B)$ is called the "posterior" and $P(B|A)$ is the "likelihood". Meaning that:

$$Posterior = \frac{Likelihood\ x\ Proposition}{Evidence} \tag{2}$$

The word Naïve in the Naïve Bayes algorithm stems from the independence assumption that assumes that every word in a document appears independent of one another. In most cases this assumption does not hold, consider the words "University" and "Degree" that appear in the same Tweet. The independence assumption states that every feature denoted as B_i in the to be classified document B is independent of every other feature B_{i+n} given the sentiment class A. For this algorithm only the extracted words from the tweet are used to calculate the probabilities.

$$P(B_i|B_{i+1}, \dots, B_{i+n}, A) = P(B_i|A) \tag{3}$$

In simpler words, this means that the algorithm calculates the probability of a Tweet being "Positive", "Neutral" or "Negative". It does so based on the occurrences of certain words in a Tweet with its corresponding label being "Positive", "Neutral" or "Negative" $P(B|A)$, and also based on the overall probability of a Tweet being "Positive" or "Negative" without looking at the content of a Tweet $P(A)$. This is divided by a constant $P(B)$ which is the same for all "Positive", "Neutral" or "Negative" Tweets. In the twitter sentiment case this means that the data fed to the model must be represented as a vector of unique words. From this a feature set is derived, this feature set states the occurrence of a word within this vector and is also labeled as either being "Positive", "Neutral" or "Negative". Based on these feature sets the classifier can calculate the probability of a Tweet begin a class based on the words that occur within a newly fed Tweet.

4.1.2 Multinomial Naïve Bayes

The Multinomial Naïve Bayes algorithm is an extension to the standard Naïve Bayes algorithm and is used quite often for sentiment analysis (Jurafsky & Martin, 2017; McCallum & Nigam, 1998). The difference between the two models is that the standard Naïve Bayes only takes the words from a tweet into account to calculate the probabilities and the Multinomial Naïve Bayes model takes in the words and the word frequencies making it a bit more sophisticated. The algorithms are both based on the Bayesian method of calculating the probability, see equation (1). In the Multinomial Naïve Bayes model the likelihood $P(B|A)$ is calculated differently, based on the word frequencies instead of only the words.

Notation	Description
A	Sentiment class
B	Tweet considered as a set of words
i	Index of a word in a tweet
B_i	Tweet feature i
$P(B\|A)$	Probability that B belongs to sentiment class A
$count(B_i, A)$	Count of word B_i occurring in class A
$count(A)$	Count of the total number of words in class A
α	Alpha smoothing
$\|V\|$	Vocabulary size of unique words in a sentiment class

Table 4: Notation Multinomial Naïve Bayes

The Multinomial Naïve Bayes algorithm states that:

$$P(B_i|A) = \frac{count(B_i, A) + \alpha}{count(A) + |V|} \tag{4}$$

P(B|A) is again the likelihood but it is calculated differently. The word frequencies are used and they are denoted as $count(B_i, A)$ where B_i is a word feature belonging to sentiment class A. In this algorithm a smoothing parameter α is used to avoid zero probabilities when words do not occur in the training data (α=1). The total amount of words in a sentiment class is denoted as $count(A)$ and the total amount of unique words in that class is denoted as $|V|$.

The algorithm thus calculates the probability of a Tweet belonging to a class that is present in the provided data. It does this based on the probability that a document belongs to a class based on the appearance of the class within the dataset. Also the words and their counts that appear in a Tweet and the words and their counts in the classes are used. When the document tokens do not provide a clear direction to which class a document belongs the highest prior probability of a class is chosen as the classification class (Kibriya, Frank, Pfahringer, & Holmes, 2004).

4.1.3 Fuzzy Fingerprints

A different approach than the Bayesian methods described above is Fuzzy Fingerprints (FFP). The idea that underlies FFP is also used in forensic sciences where a subject leaves evidence-based the actions taken and that a subject can be identified by the left evidence (Stein, 2005). In computer science fingerprints are also a known technique for mapping large datasets into a smaller and compact form of features that can correctly identify the original data (Marujo et al., 2015). To perform a correct classification the constructed fingerprint must be able to correctly identify the features apparent in the classification classes, the probability that two classes yield the same fingerprint must be low to minimize errors. To construct such a fingerprint first the top-k-word frequencies are gathered for all classification classes, after the frequencies have been gathered the fingerprints for each class have to be determined. The fingerprint is constructed by ranking the top-k words from most frequent to less frequent, after the ranking a fuzzification function is applied to the ranked words to derive membership values per top-k ranked word.

The amount of top-k words is stated as k and the rank of the words in the constructed fingerprint can be denoted by i. The fuzzification functions that calculate a score based on the top-k words and the rank in the fingperint are denoted by μ, he fuzzification of a word in the fingerprint is thus represented by $\mu(i)$ see equation 5 and 6 on the following page.

Notation	Description
k	Top-k words
i	Rank of word in fingerprint
μ	Fuzzification function
μ_i	Fuzzification of word with rank i

Table 5: Notation fuzzification

Two alternative membership attribution functions are used for the fuzzification:

The simplest linear fuzzification function is $\mu 1(i)$:

$$\mu 1(i) = \frac{k - i}{k} \tag{5}$$

$\mu 2(i)$ takes more weight into account for words that appear more frequently:

$$\mu 2(i) = \begin{cases} 1 - (1 - 0.2) * \dfrac{i}{0.2 * k} & if\ i \leq 0.2 * k \\ 0.2 \left(1 - \dfrac{i - 0.2 * k}{k - 0.2 * k}\right) & if\ i > 0.2 * k \end{cases} \tag{6}$$

These functions are shown graphically in Figure 5: fuzzification functions.

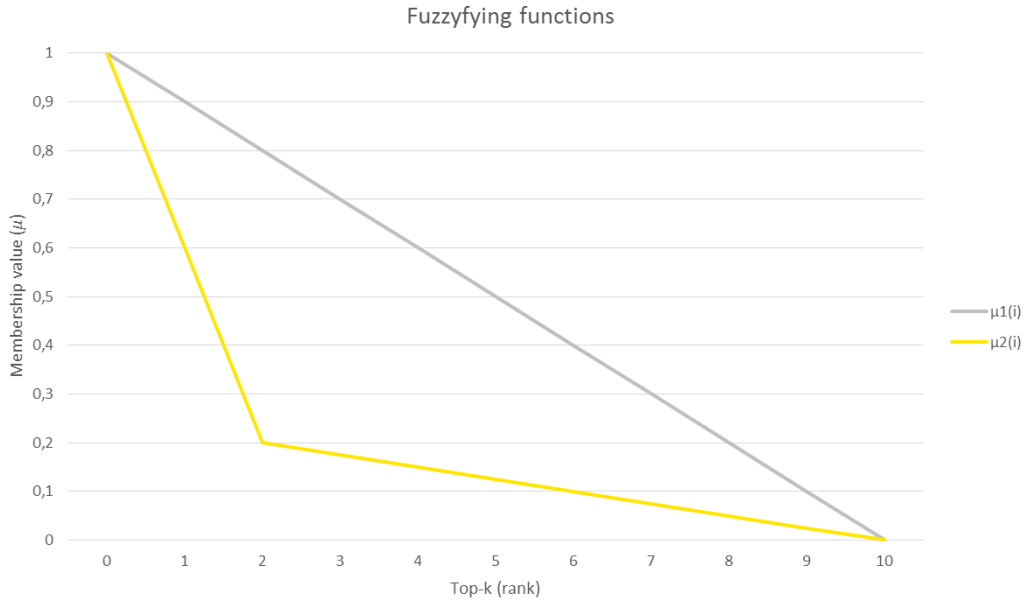


Figure 5: fuzzification functions

When the fingerprints have been constructed for each class the classification is performed. In the original algorithm the classification is based on the constructed fingerprints per class and compared between the fingerprints of the to be classified document (Marujo et al., 2015).

Notation	Description
D	Tweet
A	Sentiment class
Φ	Fingerprint
$\Phi(D)$	Fingerprint of a Tweet
$\Phi(A)$	Fingerprint of a sentiment class
$sim\Phi D, A$	Similarity between fingerprint of Tweet D and sentiment class A
μ_v	Membership value that stems from fuzzification
k	Amount of top-k words

Table 6: Notation Fuzzy Fingerprint

In the fingerprint algorithm there is a to be classified Tweet, it is represented by D. The fingerprint itself is denoted as Φ, the fingerprint of a tweet thus becomes $\Phi(D)$. The sentiment classes are again represented by A, the fingerprint of a sentiment class is thus $\Phi(A)$. The classification is based on the most similar fingerprint of a Tweet to the sentiment class, this is stated as $sim\Phi D, A$. This is calculated using equation 7. Here the μ_v represents the membership value of a Tweet fingerprint ($\Phi(D)$) or a fingerprint of sentiment class $\Phi(A)$.

$$sim\Phi D, A = \sum_{v} \frac{\min(\mu_v(\Phi(D)), \mu_v(\Phi(A))}{k} \tag{7}$$

For Tweets there is an adapted approach developed due to the small size of Tweets. Words need to be as unique as possible to correctly classify them because of the limited number of words, therefore the fingerprint library is not only based on the word frequencies, but also on an adaption of the Inversed Document Frequency (IDF). The IDF is a principle that constructs a score that becomes lower when words appear in more documents, i.e. words occurring in fewer documents are likely to provide us with more detail for characterizing the document. The IDF score for a word is constructed by dividing the total number of documents N by by the number of documents that contains a specific term n_i (Church & Gale, 1999).

$$idf = \log\left(\frac{N}{n_i}\right) \tag{8}$$

The adaption of the IDF score as proposed by Rosa, Batista, et al., (2014) resides in the fact that N becomes the number of classes and that the term n_i becomes the number of classes in which a word is present. This score is then multiplied by the count of the word in that class to compute the final score, in this case the Inversed Class Frequency (ICF) score. An example of the classes and their top-3 words fingerprints with their ICF scores is shown in Table 7: ICF scores and ranking example.

Class	Features	Counter	n_i	ICF	Rank
"Positive"	Love	8	2	1.408	2
	Like	5	3	0	3
	Donation	3	1	1.431	1
"Neutral"	Oil	6	3	0	3
	Drilling	5	2	0.88	1
	Water	1	2	0.176	2
"Negative"	Spill	12	2	2.112	2
	Hate	11	1	5.427	1
	BP	9	3	0	3

Table 7: ICF scores and ranking example

After the top-k list with corresponding ranks for a given class is constructed the same approach as in the original algorithm is applied, the features are fed to the fuzzification function and the membership values are determined based on the ranks. In the original approach the fingerprint of a new tweet is made for classification after fuzzification, but in this case that is not useful as tweet as they can only contain 140 characters and thus only a small amount of words. Instead of comparing the fingerprints with one another a new normalized Twitter Similarity Score (T2S2 score) is introduced (Rosa, Batista, et al., 2014). The T2S2 score provides a normalized value between 0 and 1 and takes into account the size of the Tweet.

Notation	Description
T	Set of words in a Tweet
Φ	Fingerprint
v	A word in a fingerprint
i	Index of a word
μ_Φ	Membership value
$\mu_\Phi(v)$	Membership value of word v
w_i	Word in fingerprint with rank i
j	Number of words in a Tweet

Table 8: Notation T2S2 score

The T2S2 score is computed by taking in the fingerprint stated as Φ and the set of words of a Tweet denoted by T. Again μ_Φ represents the membership degree, a word in the topic fingerprint is denoted as v, $\mu_\Phi(v)$ is thus the membershipvalue of word v in the fingerprint. Only the membership values of words that are apparent in the Tweet itself and the and the fingerprint are considered, hence the notation $v \epsilon (\Phi \cap T)$. These values are then summed and divided by the sum of the membership values of the top-k ranked words in the fingerprint, the words in the fingerprint are stated as w_i. Here i is the rank of the words and the number of words in a Tweet is denoted as j. The T2S2 score basically divides the sum of the membership values of every word that is common in a Tweet and a sentiment class fingerprint by the membership values from the top-k words in a fingerprint.

$$T2S2(\Phi, T) = \frac{\sum_v \mu_\Phi(v) : v \epsilon (\Phi \cap T)}{\sum_{i=0}^{j} \mu_\Phi(w_i)} \tag{9}$$

4.1.4 Evaluation metrics

Besides the algorithms and the working mentioned it is also important to evaluate the results of the algorithms. The evaluation method used is the F1 score which is based on the precision and recall of the model.

Notation	Description
$TP(class)$	True positives of a sentiment class
$FP(class)$	False positives of a sentiment class
$FN(class)$	False Negatives of a sentiment class
$TN(class)$	True negatives of a sentiment class

Table 9: Notation evaluation metrics

The precision value states the number of correctly classified examples, divided by the number of examples labeled as that class (Sokolova & Lapalme, 2009). The precision is calculated for each class and can thus be annotated as $precision(class)$, it takes the number of correctly classified examples which are the true positives (TP) denoted as $TP(class)$. It divides the true positives by the number of examples labeled as that class which are the true positives of a class and the false positives (FP) of a

class, where the false positives are represented as $FP(class)$. Precision thus measures the number of correctly classified instances of a class, i.e. how "useful" the results are, see equation 10 on the next page.

Recall is the number of correctly classified examples divided by the number of examples of that class within the data (Sokolova & Lapalme, 2009), recall is also calculated for each class and is denoted as $recall(class)$. The number of correctly classified examples is again denoted as $TP(class)$. The number of examples of a specific class is represented by the true positives of a class and the false negatives (FP) of a class which is represented as $FN(class)$. Recall measures the number of classified instances in relation to the total amount of instances of that specific class, also the "completeness" of the results. The formula for recall is presented in equation 11.

The F1 score shows the relationship between a data's given labels and those classified by the classifier based on the precision and recall scores (Sokolova & Lapalme, 2009), see equation 12. To represent the results there are two option available, micro and macro averaging. Micro averaging calculates the performance for the individual true positives, false positives, and false negatives of the model. Macro averaging represents the performance of a model by averaging the precision and recall scores of the datasets (Asch, 2013). There is chosen to use Macro averaging for the representation of the performance as there is only one dataset and performance of the system across all classes without bias is used to evaluate the models. Macro averaging gives equal weight to the performance of the classes whereas in micro averaging the large classed dominate small classes (Asch, 2013).

$$precision(class) = \frac{TP(class)}{TP(class) + FP(class)} \tag{10}$$

$$recall(class) = \frac{TP(class)}{TP(class) + FN(class)} \tag{11}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \tag{12}$$

To get the precision and recall scores of the classifier a multiclass confusion matrix is needed that shows the actual and predicted classes of the classifier shown in Table 10 below.

		Predicted classes		
		"Positive"	**"Neutral"**	**"Negative"**
Actual classes	**"Positive"**	TP(pos)	A(pos|neg)	A(pos|neg)
	"Neutral"	A(neu|pos)	TP(neu)	A(neu|neg)
	"Negative"	A(neg|pos)	A(neg|neu)	TP(neg)

TP(pos) = True positives for class positive
A(neu|pos) = Actual class neutral, predicted class positive

Table 10: Multiclass confusion matrix

4.2 Classification models

The classification models are based on the workings of the algorithms as presented in the previous chapters. There are two datasets used within this project because the BP data is unaccompanied by sentiment labels and needs to be manually labeled. The process of manually labeling Tweets can contain faults because it is a human task, therefore there is chosen to use two datasets. The pre-labeled dataset and the BP dataset will both be used for classification of Tweets. If there is any major difference observed between the classification results for BP data and the data from Niek Sanders it shows that there is an error in the manually labeling process. The Niek Sanders dataset is chosen as it is used in academic papers to test sentiment classifiers (Bravo-Marquez, Mendoza, & Poblete, 2013; K. Liu, Li, & Guo, 2010).

4.2.1 Naïve Bayes

The first model starts with acquiring two datasets, one academic dataset from Niek Sanders and the tweets from BP. The dataset from Niek Sanders is used to test if there are any mistakes made within the manual labeling part of the BP Tweet process as the BP data does not contain any labels. Both datasets are used for classification and if the results differ significantly there is assumed that an error occurred when manually labeling. After acquiring both sets the data is explored to see if the quality is good enough and all necessary attributes are included in the datasets. The dataset from Niek Sanders is already labeled, the BP dataset needs to be manually labeled to provide the classifier with training data as it concerns supervised learning. Subsequently the data is pre-processed to make it ready for the classifier. After the pre-processing a Naïve Bayes classifier is build and the data is passed through, the classifier gives an output for both datasets. The results are then evaluated and compared. This cycle is graphically shown in Figure 6: sentiment model 1 on the following page. This model sets a base classifier which shows the usefulness of the Naïve Bayes classifier on Tweet data.

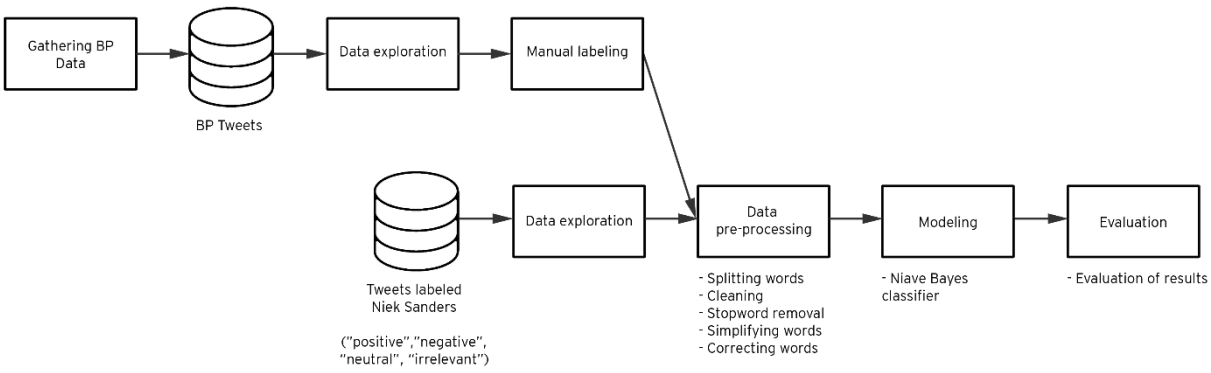


Figure 6: sentiment model 1

4.2.2 Multinomial Naïve Bayes

Following the CRISP-DM cycle there are multiple iterations based on the evaluation of the applied models. The first iteration shows the improvement and comparison of a new classification model that is based on the same theory, namely the Multinomial Naive Bayes algorithm. Only the BP dataset is used to compare the results, the same pre-processing and steps are used as in the previous model. The results of the two Naïve Bayes models are assessed and compared. The second model set-up is shown in Figure 7: Sentiment model 2 on the next page.

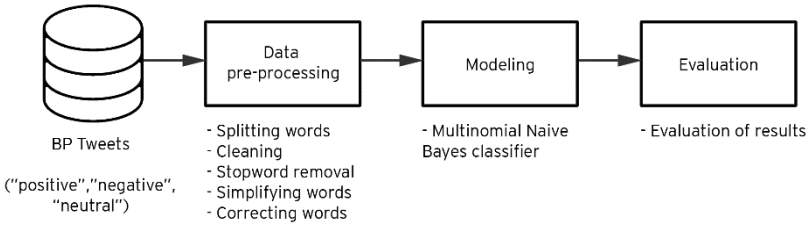


Figure 7: Sentiment model 2

4.2.3 Fuzzy fingerprint

The second iteration concerns the improvement and comparison of a new innovative classification model called fuzzy fingerprints. For this comparison only the BP dataset is needed. The dataset is already explored in the previous cycle and can thus be pre-processed immediately. The results of the Fuzzy Fingerprint model and the previous Naïve Bayes models can now be assessed and compared. The third model set-up is shown in Figure 8: sentiment model 3.

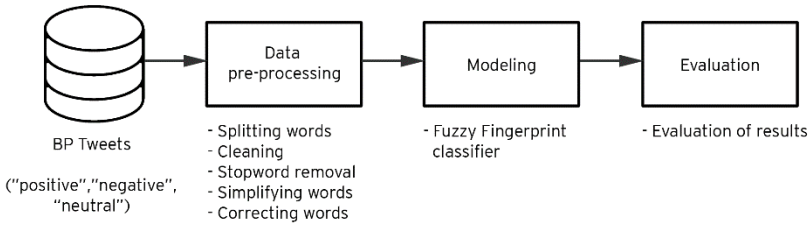


Figure 8: sentiment model 3

4.2.4 Improving pre-processing

The fourth sentiment model concerns the improvements of the pre-processing steps that can increase the performance of the models. Again the BP data is used for the improved pre-processing steps, after the data is ready all previous algorithms are used to classify the tweets more and see what the increases in evaluation scores are. The process is shown in Figure 9: sentiment model 4.

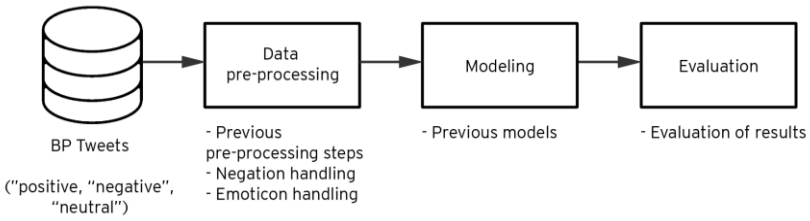


Figure 9: sentiment model 4

4.2.5 IDF penalized Multinomial Naïve Bayes

The fifth and last sentiment model concerns tries to improve the classification results of the Multinomial Naïve Bayes algorithm based on the inversed document frequency penalties as applied in the Fuzzy Fingerprint method. The BP data of the improved pre-processing is used for classification. After the classification procedure the results are evaluated, this process is shown in Figure 10: sentiment model 5 on the next page.

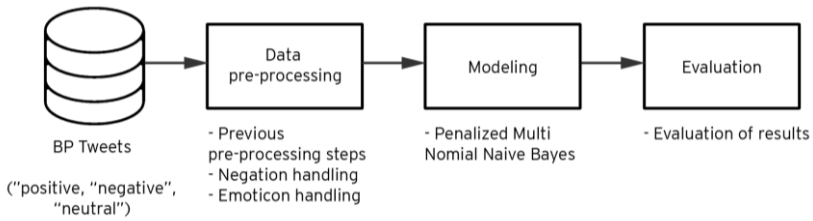


Figure 10: sentiment model 5

4.3 Data acquisition

To perform any data analysis there needs to be data. This data is acquired from Twitter and from the Niek Sanders website. The procedures followed and executed are stated on the following pages.

4.3.1 BP data

To gather the historic Tweets that range from 2009 till 2014 a web-scraping method is applied as the Twitter Application Programming Interface (API) does not provide historical Tweets older than 7 days (Twitter, 2017). Data is gathered by accessing Twitter's website and executing an advanced search query. First there is an advanced search executed towards the timeline of BP's main twitter account @BP_america, all Tweets directed towards this account are retrieved. A total of 8.036 Tweets are found within this first advanced search. The constraints for the advanced search are shown in Table 11: search term 1 and constraints.

Search term	Constraints
Directed towards @BP_america	English language only Between 01-01-2009 and 01-01-2014

Table 11: search term 1 and constraints

Second, there is an advanced search conducted that finds every Tweet that includes the hashtags #britishpetroleum and #beyondpetroleum as these are the definitions of the abbreviation BP. #BP is excluded from the search as #BP results in many Tweets that are unrelated to the oil company BP. This is due to slang and other meanings of the abbreviation BP, for example #BP also stands for #Beautiful People and # Bachelor party. A total of 582 Tweets are found within this second advanced search. The constraints for the advanced search are shown in Table 12: search term 2 and constraints.

Search term	Constraints
#britishpetroleum	English language only Between 01-01-2009 and 01-01-2014

Table 12: search term 2 and constraints

A total of 72 Tweets are found within this third advanced search. The constraints for the advanced search are shown in Table 13: search term 3 and constraints.

Search term	Constraints
#beyondpetroleum	English language only Between 01-01-2009 and 01-01-2014

Table 13: search term 3 and constraints

Besides the Tweets directed towards BP's main twitter account @BP_america there are also Tweets that mention this twitter account. These Tweets are also extracted and a total of 23.531 Tweets are found within this fourth advanced search. The constraints for the advanced search are shown in Table 14: search term 4 and constraints.

Search term	Constraints
Mentioning @BP_america	English language only Between 01-01-2009 and 01-01-2014

Table 14: search term 4 and constraints

Data is extracted from four different advanced search queries, the first one is the main Twitter account of BP. BP has two other accounts: @BP_plc and @BP_press. These accounts are not deemed fit for analysis because the creation date of these accounts lie far beyond the Deepwater Horizon Oil spill and contain fewer Tweets than @BP_america. The hashtag searches are included because there are also Tweets that are not addressed towards specific BP twitter accounts but do provide sentiment expressed regarding the oil spill. Other search terms such as the combination of #BP and #oilspill would provide Tweets that are related to BP and the Deepwater Horizon disaster, unfortunately, these search terms already include a negative sentiment and are thus excluded from analysis to maintain the most objectivity possible. The searches executed provide a lot of Tweets in total, there are probably Tweets that have been collected twice. These double Tweets are removed from the dataset in the chapter of pre-processing. As there are no free alternatives to obtain the data required a web-scraping method is applied. Web scraping is gathering data from a website other than using the website's API (Mitchell, 2016). A program is written that queries the Twitter web server, requests the data and then parses the data to extract information that is required. A Python program is designed to retrieve the Tweets from the advanced search, it relies on the following packages: CSV, Selenium and Time. The CSV package allows the reading and writing of CSV files within Python. Selenium is a package that is used to automate the interaction with a web browser and server, it is used to automatically scroll pages and to access and retrieve information of a website. The time package allows to keep track of time and set certain time requirements within the code.

The web crawler first imports the above-mentioned packages. The path of the google chrome web driver is specified and the web driver is initialized, the link to the Twitter advanced search is provided to the web driver. The web driver then parses the provided URL and opens the webpage in the browser. First the time is set for the pause between loading the page and scrolling down as this might take some time to load, also the height in coordinates of the webpage is stored in a variable to check if the scrolling has reached the end of the page. A while loop is made that uses the selenium package to scroll down to the bottom of the page, the script then pauses to load the entire page. After the pause it checks if the new coordinates are different from the last stored height coordinates, if they are different the loop is executed again until the last scroll height and the current height of the webpage are the same. After the page has reached the bottom the data of the twitter page is analyzed based on the corresponding Java embedding, the selenium packages searches for the embeddings of the names, dates and Tweet messages. These Java embeddings are searched for and the information extracted is stored in a list within Python. The script loops this procedure until all embeddings are found and extracted. The data is then extracted from the lists and written into the three separate .csv files for names, dates and Tweets. Figure 11: Working Python-based web crawler on page 32 provides a graphical sequence of the tasks performed by the web crawler. The web crawler code is included in Appendix: 8.10 Web crawler code.

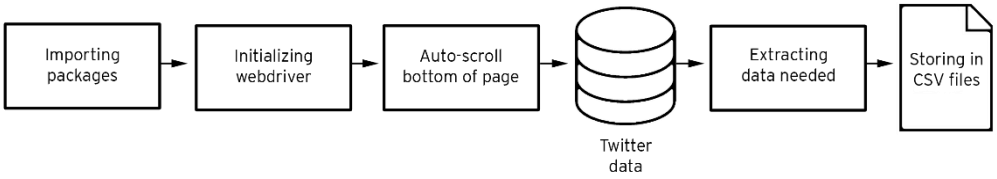


Figure 11: Working Python-based web crawler

4.3.2 Niek Sanders data

The dataset is acquired by running a python script that downloads Tweets via a unique identifier from the Twitter website, this script is downloaded from the Sanalytics website (Sanders, 2011) when the script is completed it stores the Tweets in a CSV file.

4.4 Data Exploration

Within data exploration it is important to look at the characteristics of the data, exploration is needed to ensure that all the data required for analysis is available. It also shows the characteristics such as data size, completeness of the data and other characteristics such as the distribution of the data, the goal is to get a grasp if there are any outliers and errors present (Zuur, Ieno, & Elphick, 2010).

4.4.1 BP data

The data extracted from the Twitter advanced search web pages is stored in three separate .csv files. These separate files are combined into one large excel file with the columns: Name, Date and Tweet. This file contains 31.658 entries ranging from 9 January 2009 till 31 December 2013. Table 15: Dataset BP Tweets format on the next page shows the format of the dataset acquired. The data set is deemed almost suitable for analysis because the intended columns and corresponding data that are needed for the sentiment analysis are present in the dataset, but there are no labels present yet which classify a Tweet as "Positive", "Negative" or "Neutral". These labels need to be assigned to the data before using the data in any model. The number of entries is deemed high enough to build a model on.

Name	Date	Tweet
BP America verified account	21-12-2013	As a new year begins, we thank our #BP employees for all they do to provide #energy for America, 365 days a year
Opus402	21-12-2013	#bp lackey @MorrellGeoff is nothing but a propaganda machine. Paid to lie & proud of it. @BP_America #boycottbp
….	….	….
Betty Blogs	09-01-2009	@MarkMayhew the #gas DOES belong to the Palestinians -not #israel or #BritishPetroleum - amazing what some people do for commodities #murder

Table 15: Dataset BP Tweets format

Within the dataset there are 11.161 unique Twitter accounts that are included in this dataset. Meaning that on average an unique account within the dataset provides around 2.84 Tweets. Figure 12: Tweet amount per date shows the distribution of the Tweet intensity in the dataset. What catches the eye is

that after the Deepwater Horizon oil spill the intensity of Tweets increases and that in the same year the intensity also drops as the oil leak is stopped. It is also noticed that in the beginning of 2009 there are almost no Tweets, this might be due to the recent creation of the BP Twitter accounts. It is observed that within the Tweets there are many special characters apparent that should be removed before building a text classifying model. There are no missing or empty values apparent in the dataset. The properties of the dataset are stated in Table 16: Properties BP dataset.

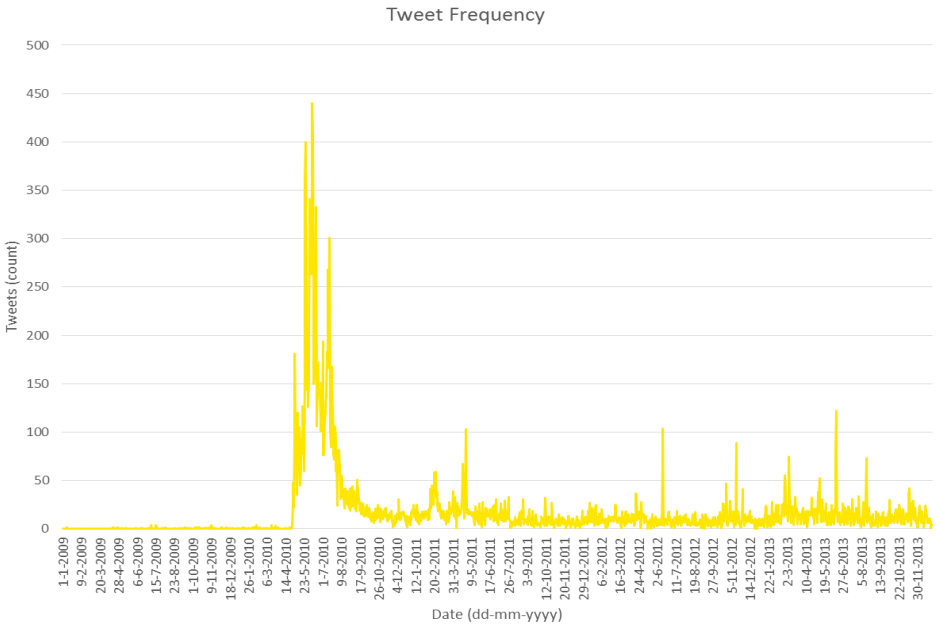


Figure 12: Tweet amount per date

Properties dataset BP	
Amount of Tweets	31.658
Amount of Twitter accounts	11.161
Range	01-01-2009 till 01-01-2014
Missing Values	None
Sentiment Labels available	No

Table 16: Properties BP dataset

To validate that the Twitter web crawler has collected all the right information a check is performed between the information on the Twitter advanced search page and the data stored in the database. Random points are checked over the time period, all entries in the database have the corresponding information in comparison to the Twitter website. The data is deemed good enough to be used for pre-processing because the relevant information is there, the web crawler successfully extracted the proper data and it does not contain major inconsistencies or quality issues.

4.4.2 Niek Sanders data

The Sanders dataset totals 5.386 rows of Tweets within the dataset that are focused on four topics (Apple, Google, Microsoft and Twitter). Each Tweet is manually labeled as either: "Positive", "Negative", "Neutral" or "Irrelevant". The "Irrelevant" labeled Tweets contain Tweets that either do not address the intended topic or are for instance not in the English language.

In total there are 571 "Negative" Tweets, 2.329 "Neutral" ones, 518 "Positive" Tweets and 1.683 irrelevant Tweets. This data is deemed not yet fit for analysis because the format needs to be

transformed so it can be loaded into a python data frame. All the required data is there: topic, classification, date and Tweet. There are inconsistencies within the data that need to address before the actual analysis. There are lines within the dataset that do not correspond to the overall format. These lines are accidentally split from the previous Tweet text and thus need to be aggregated. The format of the acquired data is shown in Table 17: Dataset Sanders Twitter.

Tweet
apple,"positive","126180209501286400","Tue Oct 18 06:18:01 +0000 2011","Lmfao look at the argument I had with Siri !! @ijustine @apple http://t.co/D4VjL7SI"
apple,"positive","126164430546403328","Tue Oct 18 05:15:18 +0000 2011","Incredible: 4 million iPhone 4Ss in 3 days. 135% better than the iPhone 4 http://t.co/1FMJxTMM @apple #iphone4s"
google,"neutral","126489609889783808","Wed Oct 19 02:47:27 +0000 2011","Panorama #icecreamsandwich #googland #google #android #ics"

Table 17: Dataset Sanders Twitter

In total 5.105 rows with Tweets remain. The comma separated values are transformed into separate columns which provide the following dataset, see Table 18: Sample dataset Sanders.

Topic	Sentiment	TweetId	Date	Tweet text
Apple	"Irrelevant"	1,25E+17	Sat Oct 15 05:36:56 +0000 2011	Partner with @apple. I&#39
Google	"Irrelevant"	1,25E+17	Sat Oct 15 12:20:16 +0000 2011	Asi o mas mala la #version5.0 de @google
Apple	"Negative"	1,25E+17	Sat Oct 15 13:49:58 +0000 2011	I wish Siri could Tweet for me . @apple fix that .

Table 18: Sample dataset Sanders

The timeframe in which the Tweets have been gathered ranges from 15 October till 20 October 2011. The amount of Tweets gathered with respect to the date is shown in Figure 13: Tweet amount per date. The most Tweets have been gathered on the 19th of October when the amount totals a little over 2500.

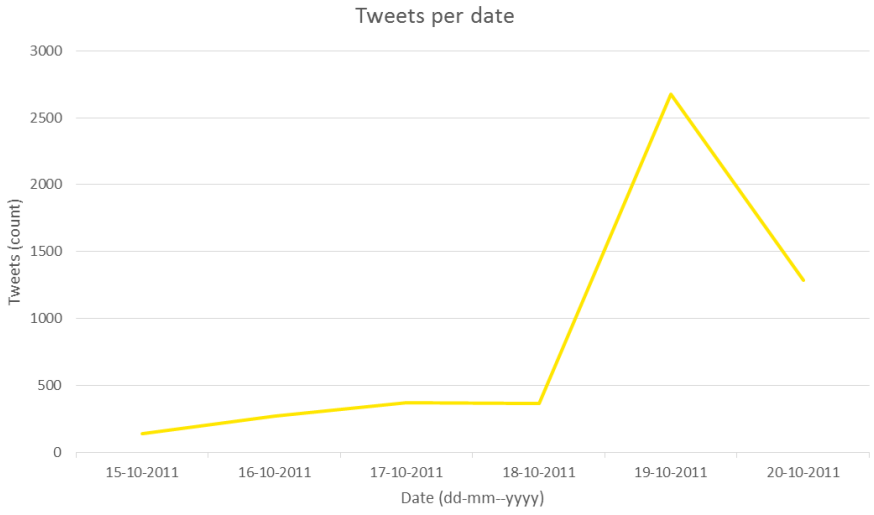


Figure 13: Tweet amount per date

The four categories that are represented within the dataset are quite equally distributed, the apple subject is represented the least in the dataset with a little under 1150 Tweets, Microsoft is the best represented with a little over 1350 Tweets. This is graphically shown in Figure 14: Tweet topic distribution.

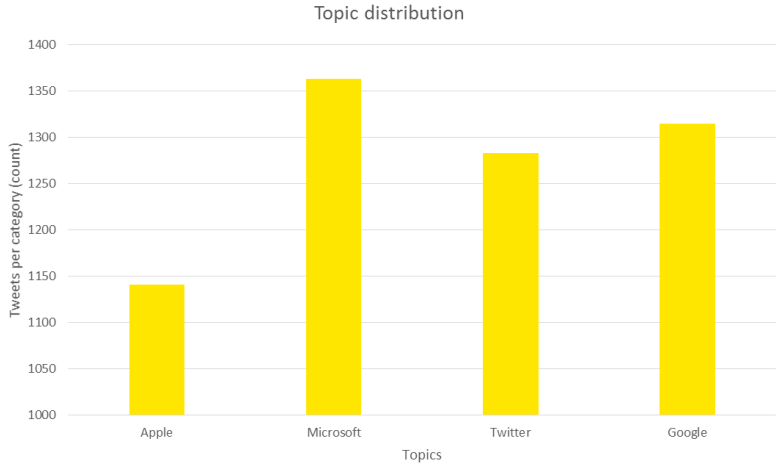


Figure 14: Tweet topic distribution

Besides the distribution of Tweets per category it is interesting to look at the distribution of sentiment classification per topic. Figure 15: Tweet sentiment distribution per topic on the next page shows this distribution. What becomes apparent is that the "irrelevant" classification is present in high volume on every topic except the Apple topic. In all topics the "Neutral" sentiment has the highest count compared to sentiment Tweets that are classified as either "Positive" or "Negative". The Google topic has the most "Positive" classified Tweets, while Twitter has the least. The Google Topic also has the least "Negative" classified Tweets while the Apple topic has the most "Negative" classified Tweets. The data is thus imbalanced and that might influence the performance of the classifier.

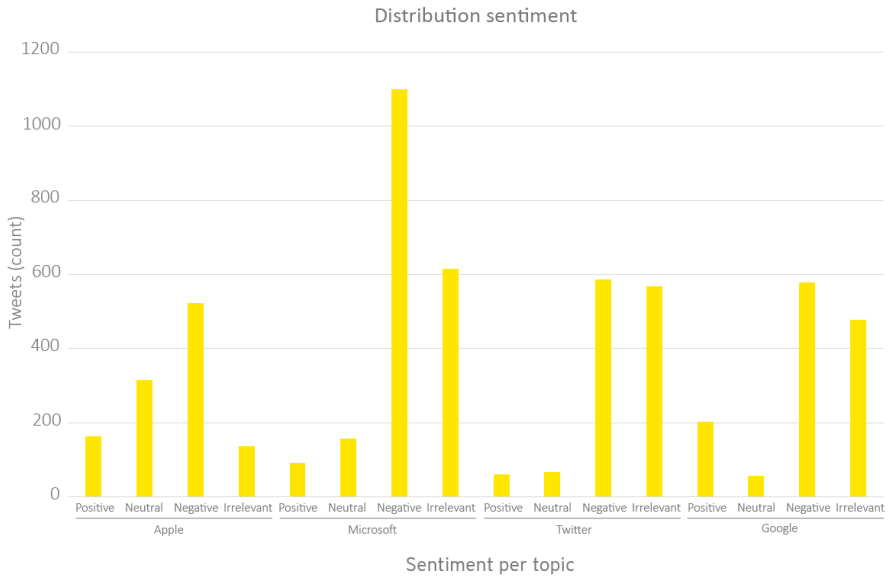


Figure 15: Tweet sentiment distribution per topic

The pre-labeled dataset contains some issues that need to be addressed before pre-processing. First the data contains a column that is not necessary for the sentiment analysis, the column of TweetID will be deleted before pre-processing. Within the Date column there is information within cells that are not required for the sentiment analysis, only the actual date is needed not the actual time of posting

a Tweet. Also the "irrelevant" labeled Tweets need to be deleted so that only the "Positive", "Negative" and "Neutral" labeled Tweets remain. Besides these issues the dataset appears to be good to use for pre-processing and eventually actual classification. The properties of the dataset are stated in Table 19: Properties Niek Sanders dataset.

Properties dataset Niek Sanders	
Amount of Tweets	5.386
Amount of Topics	4
Range	15-10-2011 till 20-10-2011
Missing Values	None
Sentiment Labels available	Yes

Table 19: Properties Niek Sanders dataset

4.5 Data preparation

Before the models can be build there needs to be data preparation to ensure that all the required data and formats are present. For the BP dataset this means that a number of sentiment labels have to be added otherwise the remaining tweets cannot be classified, also the tweets from BP accounts need to be removed as they do not reflect the public opinion and they induce subjective sentiment within the dataset. For the Niek Sanders dataset the Date column of the dataset is transformed to show only the actual date. The irrelevant tagged tweets are removed and also the TweetID column is removed to only store the necessary information.

4.5.1 Manual labeling

Before the BP Tweets can be used for cleaning and later classification there need to be sentiment labels for a part of the Tweets. To make sure that there is a clear structure within the labels a labeling protocol is designed, see Appendix 8.3 Protocol Manual labeling Tweets. The tweets sent into the world by BP accounts have to be removed from the dataset to obtain the most objective sentiment towards BP. The dataset is scanned for any account names that stem from BP by searching for names that include BP and occur frequently. In total there are seven accounts removed that are affiliated with BP. These account names are provided to a search function and the data frame looks for these specific names and removes them from the dataset.

The labeling is done three-fold, by the researcher itself and by two fellow students, all labeled the same amount of tweets and any discrepancies in the assigned labels are checked and discussed. Due to time constraints not all Tweets are labeled but approximately 1/3rd of the Tweets are manually labeled based on this protocol. Around 10% of the labeled tweets differed between the assigned classes, these instances were discussed and the majority vote on the class label is assigned, resulting in a total of 3.296 "Neutral" Tweets, 2.586 "Negative" and 530 "Positive" Tweets. The protocol used states how tweets read should be classified based on the sentiment (or not) expressed in the Tweets. For example, Tweets that contain a negative sentiment towards the subject BP are labeled as "Negative" while Tweets that contain a positive sentiment towards the subject BP are labeled as "Positive", etc. There are cases where it is unclear what sentiment is expressed these Tweets are marked as unresolved and are discussed among the Tweet labeling team. After they are discussed they are either labeled as agreed upon by the majority or removed from the dataset.

5 Modeling and results

Chapter 5 describes the steps taken to classify tweets based on the models from the previous chapter. First the data is made ready for classification via pre-processing, after the data is ready to be used a model is built in the modeling phase subsequently the model is built and executed the results are evaluated. After evaluation other iterations on models are conducted to improve the performance of the process see also chapter 1.6 Research design. All data analysis is constructed with python 3.6.3 as python lists some good packages such as the NLTK package for natural language processing (NLP) and sentiment analysis.

5.1 Data preprocessing

Before data can be used to classify Tweets it is often useful to use pre-processing to remove the data from any inconsistencies and remove information that is not useful for classification (Khan, Qamar, & Bashir, 2016a). The pre-processing steps are based on the pre-processing steps observed within the systematic literature review (see Appendix 8.1 Systematic literature review). After the data is cleaned a classification model can be made. Pre-processing is a critical step in preparing the data before the actual analysis to transform the unstructured data in a format that is better suited for analysis (Vijayarani, Ilamathi, & Nithya, 2015). This unstructured data contains a lot of noise and uninformative parts in the text such as HTML tags or special characters, this kind of data needs to be removed as it influences the speed and performance of the text classification (Haddi, Liu, & Shi, 2013). The sentiment analysis approach adopted processes the raw data obtained from Twitter, which is unstructured, into more suitable text that is free from discrepancies and semantically empty words. In the pre-processing process there are 13 steps executed to receive objective and semantically rich text. Many of these pre-processing steps have been used in other researches read during the systematic literature review (Bravo-Marquez et al., 2013; Khan et al., 2016a; Khan, Qamar, & Bashir, 2016b; Minanović, Gabelica, & Krstić, 2014; Mudinas, Zhang, & Levene, 2012; Sharma & Dey, 2013; Yang, Chen, & Chang, 2015) and form the basis for further data analysis. Figure 16: Pre-processing steps shows the data and the steps taken to pre-process the data.

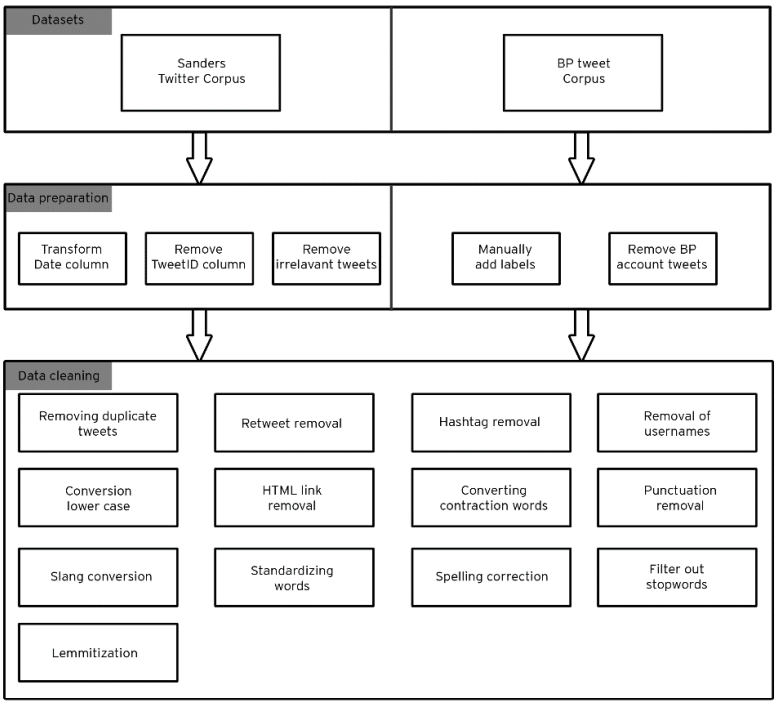


Figure 16: Pre-processing steps cycle 1

Let's now consider a set of Tweets sent into the world, and walk through the process of data cleaning. The code used for the data preprocessing can be found in appendix 8.11 Data cleaning code 1. Suppose that the following data is part of the total dataset (for ease there is chosen to show only some BP Data):

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	#BP releases latest forecast for #energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ #BPstats
GI_JOE234	30-12-2009	"Neutral"	#BP releases latest forecast for #energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ #BPstats
Sjefke76	29-12-2010	"Negative"	@MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, #animals are dying @BP_America. https://bit.ly/ghlkuej
Iceagesid77	01-01-2011	"Negative"	RT @MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, #animals are dying @BP_America. https://bit.ly/ghlkuej

Table 20: Example data preprocessing

5.1.1 Removing duplicates

It is possible that the same Tweets are apparent in the datasets. These duplicate Tweets need to be removed from the datasets to ensure that every sentiment expressed is unique and provides the most objective picture (Go, Bhayani, & Huang, 2009). The datasets are loaded into a python data frame, this data frame stores all the extracted data in the columns: Topic, Sentiment, Date and TweetText. The structure of a data frame in Python allows for finding duplicates by using the .duplicates() extension. The data frame is then checked for identical rows and the first instances are kept, all other duplicates are dropped. This results in the dataset on the following page.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	#BP releases latest forecast for #energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ #BPstats
Sjefke76	29-12-2010	"Negative"	@MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, #animals are dying @BP_America. https://bit.ly/ghlkuej
Iceagesid77	01-01-2011	"Negative"	RT @MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, #animals are dying @BP_America. https://bit.ly/ghlkuej

Table 21: Data after removing duplicates

5.1.2 ReTweet removal

ReTweets are Tweets that stem from account X and are reproduced by account Y with the characters "RT" in front of the original Tweet. ReTweets distort the objective view of the sentiment by doubling the same Tweet and its sentiment towards BP and thus have to be removed from the dataset (Go et al., 2009). As the marking of a ReTweet is indicated with "RT", the data frame searches for strings that contain the text "RT" in the front of the tweet and removes any instances that adhere to this criteria. This search function is capital sensitive and it is thus unlikely that other Tweets than ReTweets

will be removed as the "RT" indicator is a specific symbol for a ReTweet. This results in the following dataset:

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	#BP releases latest forecast for #energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ #BPstats
Sjefke76	29-12-2010	"Negative"	@MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, #animals are dying @BP_America. https://bit.ly/ghlkuej

Table 22: Data after removing reTweets

5.1.3 Tweet hashtag removal

The hashtags and the words attached to it that reside within the Tweets describe a certain topic or theme. These expressions can thus be valuable for the sentiment expressed within a Tweet, consider the hashtag #angry. The hashtags are removed from the Tweet to make sure that the text mining techniques applied for the actual sentiment analysis can read these themes. To search for hashtags the data frame looks for a specific string of text via regular expressions. Regular expressions is a method that can locate specific character strings embedded in text (Thompson, 1968), the data frame then removes any instances of the found regular expression. This results in a Tweet without any hashtags as shown in Table 23: Data after removing hashtags.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	BP releases latest forecast for energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ BPstats
Sjefke76	29-12-2010	"Negative"	.@MiamiLib This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, animals are dying @BP_America. https://bit.ly/ghlkuej

Table 23: Data after removing hashtags

5.1.4 Tweet removal usernames

Usernames are mentioned within Tweets to direct Tweets toward specific accounts. These usernames hold no sentimental value and thus have to be removed to avoid any discrepancies and to reduce the preprocessing time of the code. Usernames start with the @ sign and are followed by the account name, the same method as used within the hashtag cleaning is applied by using regular expressions. The result is a Tweet without any usernames as shown in Table 24: Data after removing usernames.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	BP releases latest forecast for energy future on Jan. 15. Join the live webcast of BP Energy Outlook 2035: http://bit.ly/1gEl9tQ BPstats
Sjefke76	29-12-2010	"Negative"	. This is sfs, I was reday to boycott toooo...maybe I'll get Shell on purpose, animals are dying. https://bit.ly/ghlkuej

Table 24: Data after removing usernames

5.1.5 Tweet Conversion lowercase

All Tweets are converted to lower case letters to avoid any inconsistencies for further data cleaning, this is especially important when using dictionaries to look up words in a string of text. The dictionaries used later in the pre-processing stage are all case sensitive and are all lower case, therefore it is necessary to convert everything to lower cases. The data frame column "Tweet" is a string and is converted to all lowercases by using python's inbuilt functions, the result is shown in table *Table 25: Data after removing capital letters*.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan. 15. join the live webcast of bp energy outlook 2035: http://bit.ly/1gel9tq bpstats
Sjefke76	29-12-2010	"Negative"	. this is sfs, i was reday to boycott toooo...maybe i'll get shell on purpose, animals are dying . https://bit.ly/ghlkuej

Table 25: Data after removing capital letters

5.1.6 Tweet HTML removal

Html tags are a common phenomenon within Tweets as users direct the reader towards website articles or any other website. These websites and their HTML reference need to be removed from the Tweet as they hold no sentimental value that can be read by the text classifier. These HTML references al start with the same letters and characters: https://, using regular expressions these HTML links are removed entirely from the Tweet. The result is a Tweet as shown in Table 26: Tweets after removing HTML links.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan. 15. join the live webcast of bp energy outlook 2035:
Sjefke76	29-12-2010	"Negative"	. this is sfs, i was reday to boycott toooo...maybe i'll get shell on purpose, animals are dying .

Table 26: Tweets after removing HTML links

5.1.7 Converting contracted words

Contraction words are words in a contracted form of two other words. For instance I'm is a contraction word made out of the words I and am. These contracted words cause problems as the punctuation within the Tweet (the next step) has to be removed and contracted words like I'll otherwise become words that possibly hold negative sentiment, in this case ill. Furthermore, these contracted words often consist out of words that are also stop words. These stop words are semantically empty (Fiori, 2014) and are thus removed from the data. A dictionary is made of the most common contraction words in the English language, this list is obtained by a publicly available source (Enchanted learning, 2016). All words in the Tweet dataset are compared to the words in the dictionary by using a loop that iterates over every Tweet, if a word in the Tweet matches a word in the dictionary it is replaced by the dictionary word via a nested list comprehension. The result is a Tweet without any contracted words, see Table 27: Tweets after converting contracted words on page 41.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan. 15. join the live webcast of bp energy outlook 2035:
Sjefke76	29-12-2010	"Negative"	. this is sfs, i was reday to boycott toooo...maybe I will get shell on purpose, animals are dying .

Table 27: Tweets after converting contracted words

5.1.8 Punctuation removal

For This first models punctuation, which also includes emoticons, is presumed to not hold any sentimental value and it thus distorts the data for the text classification. Python has an inbuilt function that differs between characters such as letters and punctuation characters, this function is used in combination with a loop to find any punctuation that resides within the text of a Tweet. When these instances are found they are replaced for a blank space in the Tweet, this is due to the fact that some punctuation connects multiple words and otherwise these words would be joined together. The result is a clean Tweet containing only text as shown in Table 28: Tweets after removing punctuation.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan 15 join the live webcast of bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	this is sfs i was reday to boycott toooo maybe I will get shell on purpose animals are dying

Table 28: Tweets after removing punctuation

5.1.9 Slang conversion

Slang is an important measure to be dealt with within Tweets, especially because Tweets can only contain 140 characters. The Tweet classifier will work better when all sorts of slang are converted into actual readable words. A publicly available database of slang is used to build a dictionary (No slang, 2017). The total dictionary consists of 5.045 slang abbreviations. All words in the Tweet dataset are compared to the words in the dictionary by using a loop that iterates over every Tweet, if a word in the Tweet matches a word in the slang dictionary it is replaced by the dictionary word via a nested list comprehension. The result is a Tweet without any slang, see Table 29: Tweets after converting slang.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan 15 join the live webcast of bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	this is so fucking stupid i was reday to boycott toooo maybe I will get shell on purpose animals are dying

Table 29: Tweets after converting slang

5.1.10 Standardizing words

Only now after the Tweets contain only text, only standard words and contain no slang and contractions the remaining words in the Tweet can be standardized. Word standardization searches for words within the Tweet that contain more than two of the same letters in a sequence and replaces it with only two of these letters. Consider the word of the Tweet: "toooo". To emphasize a certain word a user puts more letters in place than is correct, this needs to be corrected to two vowels to accurately classify the text. Note that this can still contain errors within the Tweet. For instance if the word "date"

is emphasized by spelling it like "dateeeeeee" the computer will return: "datee". This is not the right spelling, but this will be corrected in the next step of the pre-processing phase. Python iterates over the words in the Tweet data and looks for any words that contain more than two of the same letters, it reduces them to only two of these letters and puts the new word back into the original string of text. This is shown in Table 30: Tweet after standardizing words.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan 15 join the live webcast of bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	this is so fucking stupid i was reday to boycott too maybe I will get shell on purpose animals are dying

Table 30: Tweet after standardizing words

5.1.11 Spelling correction

Correcting the spelling within a Tweet is crucial for accurate classification, otherwise semantically rich words will be interpreted wrong by the text classifier (Bakliwal et al., 2012). There is a spellings checker package for python based on the work of Peter Norvig (Norvig, 2016), it is trained on a corpus of around a million words and achieves around 80-90% accuracy which is deemed sufficient for checking the Tweets. All separate words need to be fed to the spellings checker making it a time-consuming job for the computer, misspelled words are corrected and placed back into the string of text. Table 31: Tweets after spellings check shows the corrected Tweet.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast for energy future on jan 15 join the live webcast of bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	this is so fucking stupid i was ready to boycott too maybe I will get shell on purpose animals are dying

Table 31: Tweets after spellings check

5.1.12 Filter out stop words

Stop words are words that are commonly used within sentences but do not contain any sentiment (Fiori, 2014), these are words like: "the", "a", "am", "that", etc. The natural language toolkit (NLTK) for python contains a list of 153 English stop words. Again the words are iterated over via a loop and with a nested list comprehension the words that are found in the Tweet and also within the stop words dictionary are removed. This results in the Tweet as shown in the table below.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releases latest forecast energy future jan 15 join live webcast bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	fucking stupid ready boycott maybe I get shell purpose animals dying

Table 32: Tweet after stop word filtering

5.1.13 Lemmatization

Lemmatization is the process of reducing inflectional forms of a word and to convert it to a common base form of a word (Manning, Raghavan, & Schütze, 2008). Words as: "car", "car's", "cars" and "cars'" are all converted to their basic form: "car". The lemmatizer based on the Wordnet Corpus from

the NLTK toolkit is used to look up words within a dictionary and it analyzes the word based on its morphological forms. It then returns the basic form of that word, which is known as the so-called: Lemma. Without Part of Speech (POS) tagging every word that is fed to the lemmatizer is assumed as a noun. This converts any different spelling of the same base word into the lemma, this improves the classification of the Tweet as the variance of the same word meaning with slightly different spelling is reduced. This process results in the final Tweet that will be used for further processing, see Table 33: Tweets after lemmatization.

Name	Date	Sentiment	Tweet
GI_JOE234	30-12-2009	"Neutral"	bp releas latest forecast energy future jan 15 join live webcast bp energy outlook 2035
Sjefke76	29-12-2010	"Negative"	fuck stupid ready boycott maybe I get shell purpose animals dying

Table 33: Tweets after lemmatization

5.2 Model 1 – Naïve Bayes

The first model for sentiment classification uses the un-preprocessed and pre-processed of both the BP and Sander Twitter datasets. This approach will show two things, if the pre-processing steps have improved the classification of Tweets and if there is a potential error in the manually labeling part. The classification approach adopted is shown in Figure 17: classification approach Naïve Bayes.

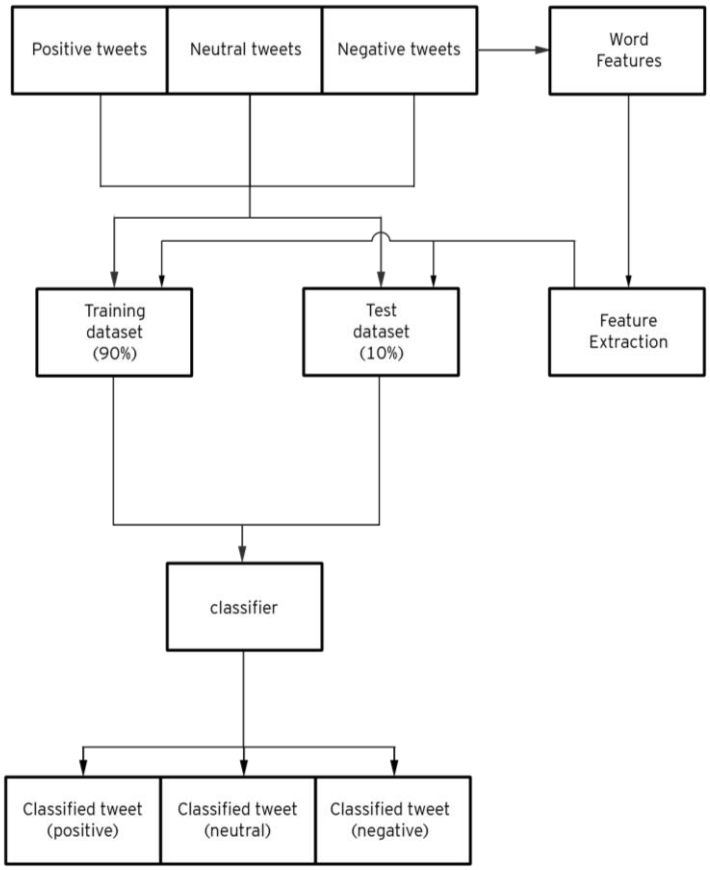


Figure 17: classification approach Naïve Bayes

First the data is imported and stored in a data frame, when the data is stored it is shuffled to avoid any bias when classifying (Lin & Kolcz, 2012). The shuffling of the data ensures that when each classification iteration is performed there is different data in the training and testing sets. Thereafter the Tweets are split by words to obtain a vector that contains all separate words per Tweet, this vector is then accompanied by its sentiment label. This happens in Python by making a loop that splits the words that appear in the column that holds the text of the Tweets. These separate words are then stored in place of the original Tweet text. When these vectors are constructed the feature vectors can be made. First all unique words are extracted from the vectors by a function that looks for all unique words within the column that now holds the separate words that appear in a Tweet, followed by its sentiment label. When this unique word list is made a function that extracts the words features is built. This function converts the words that are used as input into a frequency distribution based on the NLTK package within python. This functions looks at the total count of word appearances and sorts it from highest to lowest count. Now the word feature is ready to be used and the features per Tweet can be extracted. The feature extraction uses the word features of unique words and is sorted based on the total occurrence within the Tweets. The actual Tweets are now fed to a function that looks for the words in the Tweets and matches them with the words within the word feature, when there is a match of words the functions returns a label "TRUE" that identifies words being apparent within a Tweet. When words do not appear within a Tweet the label is "FALSE", This vector of words with corresponding "TRUE" and "FALSE" labels combined with the sentiment labels makes the total feature set of the data. The algorithm can now calculate the occurrences of words within "Positive" or "Negative" Tweets and can try to classify new Tweets based on this information.

Thereafter the data must be divided into a training and a testing set. To reduce bias within the dataset the same proportions of Tweets directed towards a topic must be present in both the training and the testing set. This ensures that although the data is shuffled there is still enough training data available for every topic that is represented in the data. For training and testing a 10-fold cross validation method is used that divides the data into 10 equal parts of data, these parts of data are then separately used for training and testing (Zhang, 1993). The first nine parts are used for training and the remaining part is used for testing, this is done 10 times and the results are averaged. The whole process is repeated 10 times to compute the final precision, recall and F1 score of the classifier.

The code of the approach can be found in appendix 8.12 Classification code 1 Naïve Bayes -BP and 8.12 Classification code 1 Naïve Bayes – Niek Sanders. The results from the classifier are shown in Table 34: Naïve Bayes results classification BP data (un pre-processed data), Table 35: Naïve Bayes results classification BP data (pre-processed data), Table 36: Naïve Bayes results classification Niek Sanders data (un pre-processed data) and Table 37: Naïve Bayes results classification Niek Sanders data (pre-processed data). Note that the precision and recall scores are averaged over all three classes. The full tables can be found in appendix 8.4 Results classifier 1 because of their size.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.67	0.68	0.67	0.67	0.67	0.67	0.68	0.67	0.66	0.67
Recall	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.65	0.65	0.65
F1 score	0.66	0.67	0.66	0.66	0.66	0.66	0.67	0.66	0.66	0.66
Average F1 score	0.66									

Table 34: Naïve Bayes results classification BP data (un pre-processed data)

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.66	0.67	0.68	0.67	0.67	0.67	0.67	0.67	0.67	0.67
Recall	0.62	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.63
F1 score	0.64	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.64	0.65
Average F1 score	0.65									

Table 35: Naïve Bayes results classification BP data (pre-processed data)

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.59	0.59	0.6	0.59	0.59	0.58	0.59	0.59	0.58	0.58
Recall	0.55	0.56	0.56	0.56	0.55	0.54	0.56	0.56	0.55	0.55
F1 score	0.57	0.57	0.58	0.57	0.57	0.56	0.57	0.57	0.57	0.57
Average F1 score	0.57									

Table 36: Naïve Bayes results classification Niek Sanders data (un pre-processed data)

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.57	0.56	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
Recall	0.54	0.54	0.54	0.54	0.55	0.54	0.55	0.54	0.54	0.55
F1 score	0.56	0.55	0.55	0.56	0.56	0.56	0.56	0.56	0.56	0.56
Average F1 score	0.56									

Table 37: Naïve Bayes results classification Niek Sanders data (pre-processed data)

5.2.1 Evaluation classification model 1

At first glance the results show no promising precision, recall and F1 scores. This applies to both the un pre-processed and the preprocessed data of both BP and the Niek Sanders dataset. Although the F-1 scores for the BP dataset are higher (0.09) the scores are still quite low (0.66 for un-preprocessed and 0.65 for pre-processed).

The first thing that stands out is that the un-preprocessed data achieves a slightly higher score in general than the pre-processed data. A first argument why this can be the case is due to the amount of input data that differs, because of the pre-processing some Tweets contain no words anymore. For example the input data for the pre-processed data of Niek Sanders contains 2.940 Tweets in comparison to 3.418 for the un pre-processed data, more training data is likely to result in higher results for precision, recall and thus F1 scores also. A second argument is that with the pre-processing there might be too much information deleted that hinders accurate classification. When looking at the precision and recall scores per prediction class for the datasets (see appendix 8.4 Results classifier 1) there is observed that for the BP data, the low average scores mainly stem from low scores in the "Positive" prediction class. For the Niek Sanders data the low scores stem from the "Positive" and "Negative" prediction classes. In general the "Positive class has low precision scores (BP: 0.47, Sanders: 0,44) and very low recall scores (BP: 0.37, Sanders: 0.29), it is clear that the classifier is not able to correctly predict this class in both cases. In the Niek Sanders dataset also the "Negative" prediction class has both low precision (0.54) and recall scores (0.54). It is important to look at the input data, the number of "Neutral" Tweets outnumbers the "Positive" and "Negative" Tweets by far for both datasets. The "Negative" Tweets accumulate to 3.296 for BP and to 1.967 for Sanders while

the "Negative" and "Positive" Tweets number respectively to 2.586 and 530 for BP and to 513 and 459 for the Sanders dataset. The lower scores are thus also explained by the unequal categorical distribution of the available data, meaning that the data is imbalanced. Another remark is that it might be possible that classified Tweets contain words that are not yet seen within the extracted word features for a class, if this is the case the probability will amount to 0 for the class. This is problematic as other words in the tweet might indicate that a Tweet should be classified within that class. A positive remark is that there is low variance observed between the results of the different runs of the classifier, meaning that the classifier is stable.

Within the preprocessing there are several issues that can be addressed further in the processes. An increase in precision, recall and F1 scores for the "Negative" and "Positive" class might be partly established by handling negation. Negation handling refers to words that alter the polarity of a word or sentence (Indurkhya & Damerau, 2010), this is thus important because it might improve the accuracy of the "Positive" and "Negative" categories. A second improvement relates to the most influential word features, there is observed that through preprocessing valuable information is lost, emoticons such as :(or :) can are used as word features to classify Tweets as "Positive" or "Negative". For the next model it is thus valuable to transform or keep emoticons that hold sentimental value into the dataset. A third improvement is that now all word features are used to classify a Tweet into a category, it might be beneficial for the classifier to only use the first couple of thousand word features, instead of all word features. This might result in higher accuracies but also in reduced computing time.

5.2.2 Conclusion classification model 1

Based on the evaluation the following conclusions are drawn up. The low precision and recall scores on the "Negative" and the "Positive" class have to be improved a lot, it seems unthinkable that this can be achieved only by improving the pre-processing with the data available as there is no more training data available. An improved classification algorithm is needed to deal with the imbalanced data as Twitter data is most likely imbalanced, this assumption is based on the distributions seen in the data and multiple papers read that all use Twitter data and have imbalanced data (Bravo-Marquez et al., 2013; Liang & Dai, 2013; Thakkar & Patel, 2015). To compare the new algorithm to the standard Naïve Bayes algorithm there is chosen to delay the improvements of the pre-processing procedure to a next iteration model to accurately show the difference between the old and new classifier. The improvements in the pre-processing procedure include: POS tagging, negation handling and emoticon inclusion. The goal of using two datasets is to check if the labeling of the BP dataset has gone right without major mistakes. Because there are no major differences observed between the BP dataset and the Niek Sanders dataset the labeling of the BP dataset seems correct with no major errors.

5.3 Model 2- Multinomial Naïve Bayes

The second classification model uses the previous used BP dataset and pre-processing steps as there appears to be no major error in manually labeling. An improved Bayesian classification algorithm is used that tries to classify not only based on the words in a Tweet but also on the word frequencies to achieve better classification results. The results of this new model will be compared to the results of the previous model, not that this model also does not consider any sentence structure or word structure to be important. The Multinomial Naïve Bayes classification model is shown in Figure 18: classification approach Multinomial Naïve Bayes on the next page.

As in the first sentiment model the data is imported and stored in a data frame, when the data is stored it is shuffled to avoid any bias when classifying (Lin & Kolcz, 2012). Then the package which transforms words into a vector with corresponding word counts is initialized. Before this countvectorizer package is used the data is split into 10 parts where 90% is used for training and 10%

is test data. The same 10-fold cross validation method is used as in the previous sentiment model. The data is then stored in an array to make sure that the input is correct for the countvectorizer package. The learning of the countvectorizer is then executed by feeding it the training data. This package then looks for all the unique words in the training data and stores a vector of these words. Based on the learned countvectorizer the data can then be transformed into a matrix that shows the word vocabulary of the training data and the count of the words per Tweet. It does so by using its learned vector and counting the words that appear in the new Tweets that are passed through. After these matrices are constructed they are the classifier is initiated. The classifier is trained on the training data and uses an alpha smoothing of 1 to avoid the zero probability issue (Jurafsky, 2012; Jurafsky & Martin, 2017). The test data is then passed through and the predicted classes are stored for comparison with the actual classes.

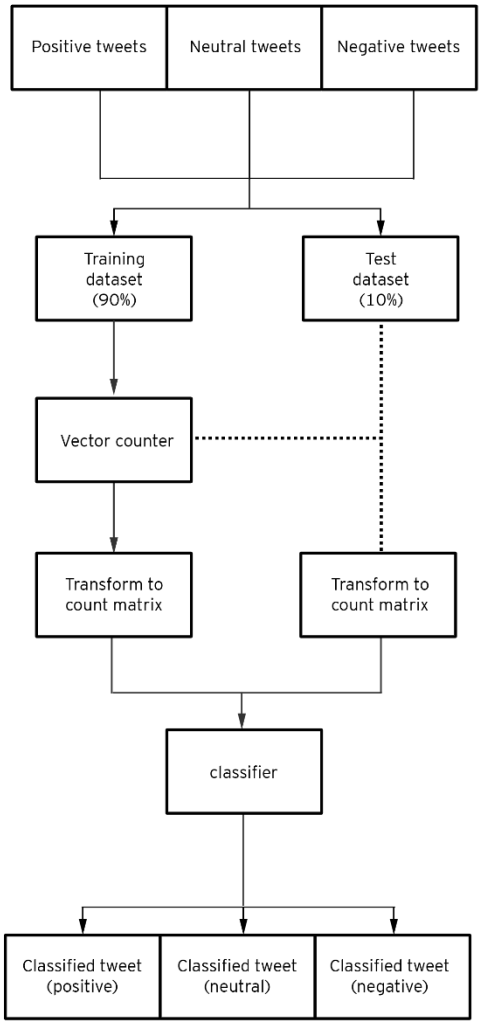


Figure 18: classification approach Multinomial Naïve Bayes

The code of the approach can be found in appendix 8.14 Classification code 2 Multinomial Naïve Bayes - BP The results from the classifier are shown in Table 38: Multinomial Naïve Bayes results classification BP data (un *pre-processed* data*)* and Table 39: Multinomial Naïve Bayes results classification BP data (pre-processed data*)*. Note that the precision and recall scores are again averaged over all three classes. The full tables can be found in appendix 8.5 Results classifier 2 because of their size. The evaluation method is the same as in the first classification model where the F1, recall and precision scores are used.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.75	0.74	0.76	0.74	0.75	0.76	0.76	0.75	0.75	0.74
Recall	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
F1 score	0.66	0.65	0.66	0.66	0.66	0.67	0.66	0.66	0.66	0.66
Average F1 score	0.66									

Table 38: Multinomial Naïve Bayes results classification BP data (un pre-processed data)

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.75	0.74	0.74	0.74	0.74	0.74	0.74	0.75	0.74	0.76
Recall	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.65	0.65	0.66
F1 score	0.69	0.69	0.69	0.69	0.69	0.69	0.70	0.70	0.70	0.70
Average F1 score	0.69									

Table 39: Multinomial Naïve Bayes results classification BP data (pre-processed data)

5.3.1 Evaluation classification model 2

The results of this new algorithm are quite the same as within the standard Naïve Bayes sentiment cycle. The F1, precision and recall scores have gone up slightly for the pre-processed data than for the un pre-processed data, now the pre-processed data achieves better results than the not pre-processed data.

Within the different classes there is observed that for the un pre-processed data the precision scores for the "Positive" class are high (0.74) but the recall scores are very low (0.16). Within the pre-processed data the precision is of this class is a bit lower (0.7) but the recall scores are higher (0.34). Still the issue remains that because of the imbalanced data there are no accurate predictions for the "Positive" class. In general the scores have thus gone up slightly. This new classifier performs better in terms of recall than the standard Naïve Bayes algorithm and thus is better at classifying the Tweet classes out the total amount of instances of that specific class. It appears difficult for the Bayesian classifier to classify the "Positive" class as there are not a lot of data points.

5.3.2 Conclusion classification model 2

As the Algorithm does not majorly improve the results and especially the most problematic class ("Positive"), it seems that Naïve Bayes is not sufficient for predicting the classes of imbalanced Tweets. More training data could partly solve this problem, but this is not available and it seems unreasonable to assume that only improved pre-processing could improve the results for the "Positive" class majorly. It seems hard to find unique words that belong to the "Positive" class because of the imbalanced data, thus a new algorithm needs to be implemented that has the potential to better deal with imbalanced data.

5.4 Model 3- Fuzzy fingerprints

The third sentiment classification model uses an innovative and new approach for text classification named Fuzzy Fingerprints (FFP). This approach has shown good results when trying to classify textual data (Homem & Carvalho, 2011; Rosa, Batista, et al., 2014). The original FFP approach is developed for classifying text that is longer than Tweets (Homem & Carvalho, 2011), but there are some adaptions to the original algorithm that can make it suitable for classifying the topics of Tweets (Rosa,

Batista, et al., 2014; Rosa, Carvalho, & Batista, 2014), see chapter 4.2.3. This approach appears to work well when dealing with imbalanced data regarding topic classification in Twitter Tweets, the results that stem from the FFP approach for classifying Twitter topics are good (F-1 score: 0.844) (Rosa, Batista, et al., 2014). It seems reasonable to assume that this method might thus work well when trying to classify sentiment in Twitter Tweets with imbalanced data as is the situation for this project. To my knowledge there is no research conducted towards the application of FFP to classify Twitter sentiment, it is thus a novel experiment that shows the usefulness of the FFP method for sentiment analysis on Tweets. The working of the classifier is graphically shown in Figure 19: classification approach fuzzy fingerprints.

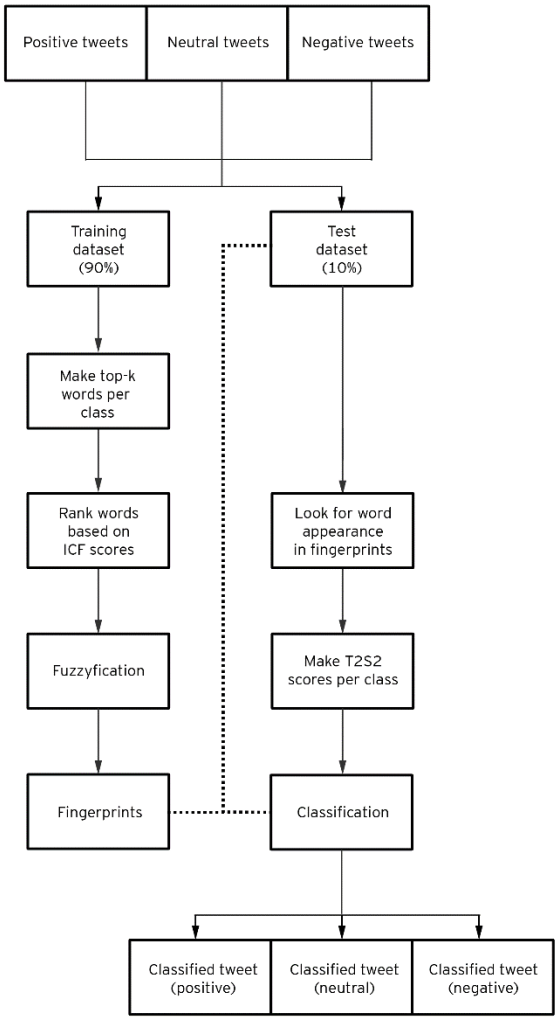


Figure 19: classification approach fuzzy fingerprints

First the data is imported and stored in a data frame, after the data is stored the data is split into the three sentiment classes: "Positive", "Neutral" and "Negative". After the data is split into the classes it is shuffled to avoid bias when performing new classification runs on the model. Then the data is split into 10 folds where 9 parts will be used for training and 1 part will be used for testing, the 10-fold cross-validation. When the data is split into the 10 folds the first loop is initiated that runs over all the three classes. Within this loop first the test and training data per class is initialized and the unique words per class are identified, also all the words apparent in the training data are stored in a vector. Based on these two vectors the word counts are constructed and the appearance of a word in the number of classes. When these measures have been constructed the top-k words based on word frequency for a class are stored in a data frame with its corresponding word counts and class

appearances. Based on this information the ICF scores are calculated and the order of the top-k words is rearranged based on the highest ICF scores, subsequently the ranks are assigned to the words. Thereafter the fuzzification functions are applied to the words and the $\mu(1)$ and $\mu(2)$ scores are computed and stored in the data frame, this is the final step for computing the fingerprint per class. When the fingerprints are constructed and stored the test data is deconstructed into words per Tweet. The words that are in a Tweet are then stored in a data frame, based on the words in the Tweet and the words in the fingerprints the $\mu(1)$ and $\mu(2)$ scores per tweet are calculated and summed. The total $\mu(1)$ and $\mu(2)$ scores are stored per class for the Tweet and the T2S2 scores are calculated. Then classification is executed based on the highest T2S2 score for a class per tweet and the results are stored. The results are then used to construct a confusion matrix per class and these matrices are used to calculate the precision and recall scores.

The code for the Fuzzy Fingerprint approach can be found in appendix 8.15 Classification code 3. The results from the classifier are shown in tables Table 40: Fuzzy Fingerprint $\mu 1(i)$ results classification BP data (un pre-processed data) till Table 43: Fuzzy Fingerprint $\mu 2(i)$ results classification BP data (pre-processed data) on the next page. The precision and recall scores are averaged over all three classes, the full tables are added in appendix 8.6 Results classifier 3. Again the precision, recall and F-1 scores are used to evaluate the classifier.

5.4.1 Evaluation classification model 3

The results that stem from the Fuzzy Fingerprint approach are not as good as expected. The high F-1 scores (0.844) as noted in Rosa, Batista, et al., 2014 are not reached. The highest achieved F-1 scores for the un pre-processed data are 0.604 for $\mu(1)$ at k=3000 and 0.605 for $\mu(2)$ at k=900. The F-1 scores for the pre-processed data are slightly better, 0.648 for $\mu(1)$ at k=900 and 0.635 for $\mu(2)$ at k=900. The precision and recall scores for the un pre-processed data are quite low, 0.585 precision and 0.625 recall for $\mu(1)$ and 0.585 precision and 0.635 recall for $\mu(2)$. When looking at the scores of $\mu(1)$ per class for the best average scores it is noted that the precision for the "Positive" score is very low (0.149) and that the recall scores for the "Neutral" and "Negative" are also quite low, 0.583 and 0.465 respectively. For $\mu(2)$ the precision for "Positive" is again low (0.182) and the recall scores for "Neutral" and "Negative" are again quite low with 0.574 and 0.572 respectively.

For the pre-processed data the scores are slightly higher. 0.612 and 0.690 precision and recall for $\mu(1)$ and 0.598 precision and 0.677 recall for $\mu(2)$. For the pre-processed $\mu(1)$ scores the problematic precision from the "Positive" class has gone up to 0.296 which is still low. The recall scores for the "Neutral" and "Negative" have gone up to mediocre scores 0.641 and 0.699. for $\mu(2)$ the same applies, precision for "Positive" increased somewhat to 0.288 and the recall scores increased to 0.598 for "Neutral" and to 0.730 for "Negative". The $\mu(2)$ fuzzification function performs better for the un pre-processed data but performs worse for the pre-processed data. In total it can be stated that the classifier thus predicts too many Tweets as "Positive" and performs mediocre at predicting the remaining classes, it performs best on the pre-processed data with the simple $\mu(1)$ fuzzification function.

5.4.2 Conclusion classification model 3

The new algorithm does not provide better results than the Multinomial Naïve Bayes algorithm, it does provide better results than the standard Naïve Bayes model. Notable is that the Fuzzy Fingerprint model achieves higher recall score for predicting the "Positive" class where the other two models do a terrible job considering recall for the "Positive" class. But this algorithm performs slightly less on the other two sentiment classes. It appears that there are too many words that appear (i.e. too few unique words per sentiment class) in all classes and that Tweets cannot be classified accurately. The next step to improve the results is to look at the pre-processing and try to make words more class specific.

Top-k	Precision	Recall	F1-score
5	0.412	0.414	0.413
10	0.433	0.468	0.450
15	0.449	0.501	0.473
20	0.468	0.530	0.497
25	0.498	0.557	0.526
30	0.507	0.558	0.531
50	0.515	0.552	0.533
100	0.525	0.590	0.555
200	0.553	0.593	0.572
300	0.560	0.579	0.570
400	0.563	0.586	0.574
500	0.568	0.599	0.583
600	0.573	0.609	0.590
700	0.574	0.611	0.592
800	0.572	0.602	0.587
900	0.577	0.624	0.600
1000	0.574	0.618	0.595
2000	0.579	**0.625**	0.601
3000	**0.585**	0.624	**0.604**
4000	0.574	0.578	0.576
5000	0.577	0.601	0.588

Table 40: Fuzzy Fingerprint $\mu 1(i)$ results classification BP data (un pre-processed data)

Top-k	Precision	Recall	F1-score
5	0.402	0.420	0.411
10	0.413	0.436	0.425
15	0.434	0.466	0.449
20	0.445	0.488	0.466
25	0.473	0.521	0.496
30	0.468	0.511	0.488
50	0.497	0.462	0.479
100	0.509	0.499	0.504
200	0.553	0.462	0.503
300	0.560	0.477	0.515
400	0.563	0.594	0.578
500	0.568	0.573	0.581
600	0.573	0.626	0.598
700	0.574	0.630	0.601
800	0.572	0.625	0.597
900	0.577	**0.635**	**0.605**
1000	0.574	**0.635**	0.603
2000	0.579	0.613	0.595
3000	**0.585**	0.604	0.594
4000	0.573	0.598	0.586
5000	0.577	0.612	0.594

Table 41: Fuzzy Fingerprint $\mu 2(i)$ results classification BP data (un pre-processed data)

Top-k	Precision	Recall	F1-score
5	0.422	0.473	0.446
10	0.474	0.506	0.489
15	0.491	0.536	0.513
20	0.507	0.563	0.534
25	0.507	0.574	0.538
30	0.511	0.581	0.544
50	0.511	0.581	0.551
100	0.543	0.622	0.580
200	0.571	0.652	0.608
300	0.574	0.654	0.611
400	0.585	0.668	0.624
500	0.587	0.671	0.626
600	0.578	0.651	0.613
700	0.594	0.674	0.631
800	0.601	0.682	0.639
900	**0.612**	**0.690**	**0.648**
1000	0.607	0.688	0.645
2000	0.592	0.654	0.622
3000	0.578	0.589	0.584
4000	0.571	0.562	0.566
5000	0.572	0.566	0.568

Table 42: Fuzzy Fingerprint $\mu 1(i)$ results classification BP data (pre-processed data)

Top-k	Precision	Recall	F1-score
5	0.439	0.478	0.458
10	0.461	0.482	0.471
15	0.477	0.508	0.492
20	0.487	0.531	0.508
25	0.489	0.547	0.516
30	0.492	0.550	0.519
50	0.491	0.547	0.517
100	0.519	0.580	0.548
200	0.549	0.617	0.581
300	0.567	0.643	0.603
400	0.577	0.659	0.616
500	0.583	0.665	0.621
600	0.594	**0.677**	0.633
700	0.581	0.657	0.617
800	0.594	0.676	0.632
900	**0.598**	**0.677**	**0.635**
1000	0.596	0.675	0.633
2000	0.588	0.664	0.624
3000	0.587	0.656	0.620
4000	0.586	0.656	0.619
5000	0.590	0.647	0.617

Table 43: Fuzzy Fingerprint $\mu 2(i)$ results classification BP data (pre-processed data)

5.5 Model 4- Improved pre-processing

The fourth classification model is based on an improvement in the pre-processing phase. Based on the previous models it seems that the words within the sentiment classes are not unique enough, the prediction classes show that the "Positive" sentiment class proves most troublesome. It is hard to predict accurately as the data for the "Positive" class is scarce. An improvement that could help is to use the emoticons in the Tweets to make Tweets more unique for a sentiment class, also the handling of negation could prove beneficial (Garg & Meher, 2015; K. Liu et al., 2010). The code for the improved pre-processing can be found in appendix 8.16 Improved pre-processing. Let's consider some sample Tweets shown below.

Name	Date	Sentiment	Tweet
Goof_81	01-01-2011	"Positive"	So happy about the donation from BP, it brings a smile to my face :) .
Ridderknorbert	23-11-2011	"Negative"	Not fun at all BP, look what you did to the ecosystem of the Gulf area :(..

Table 44: Tweet examples

The previous pre-processing steps as described in chapter 5.1 Data preprocessing remain the same, the addition lies in the handling om negation and emoticons as this could provide more detailed information about the features that appear in a Tweet and the sentiment class it belongs to. First an emoticons dictionary is build based on a database of common text emoticons (PC.net, n.d.), a total of 87 common emoticons are used. It is noticed that in the Tweets some emoticons are attached to punctuation characters and these need to be split to accurately convert emoticons. These patterns are stated and by using regular expressions these characters are split. Then a loop is initiated that looks for the emoticons in the Tweets and replaces all the positive associated emoticons with the text "HAPPY" and all the negativity associated emoticons with the word "SAD". The result is shown in table Table 45: Tweets after converting emoticons

Name	Date	Sentiment	Tweet
Goof_81	01-01-2011	"Positive"	So happy about the donation from BP, it brings a smile to my face HAPPY .
Ridderknorbert	23-11-2011	"Negative"	Not fun at all BP, look what you did to the ecosystem of the Gulf area SAD ..

Table 45: Tweets after converting emoticons

For the negation handling the same approach adopted by Garg & Meher, 2015 is used that is based on the work of Narayanan, Arora, & Bhatia, 2013. First the Tweets are split into words and the words are passed through a Part Of Speech (POS) tagger which labels the words based on its part of speech such as: noun, verb, adverb, etc. When the words are tagged the words are iterated over to look for negation words that alter the sentimental value of a word. When such a word is found the next word is a noun, adjective, verb, adverb or determiner the word is edited to have the string "not_" in front of the word. The result from handling negation is shown in Table 46: Tweets after handling negation.

Name	Date	Sentiment	Tweet
Goof_81	01-01-2011	"Positive"	So happy about the donation from BP, it brings a smile to my face HAPPY .
Ridderknorbert	23-11-2011	"Negative"	Not not_fun at all BP, look what you did to the ecosystem of the Gulf area SAD ..

Table 46: Tweets after handling negation

5.5.1 Evaluation classification model 4

The results from the fourth model are shown in tables Table 47: Naïve Bayes results improved *pre-processing* till Table 50: Fuzzy Fingerprint $\mu 2(i)$ results improved pre-processing. The precision and recall scores are averaged over all three classes, the full tables are added in appendix 8.7 Results classifier 4. Again the precision, recall and F-1 scores are used to evaluate the classifier.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.68	0.69	0.67	0.68	0.68	0.69	0.67	0.69	0.68	0.69
Recall	0.63	0.63	0.63	0.64	0.63	0.64	0.63	0.63	0.64	0.64
F1 score	0.66	0.66	0.65	0.66	0.65	0.66	0.65	0.66	0.65	0.66
Average F1 score	0.66									

Table 47: Naïve Bayes results improved pre-processing

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.75	0.75	0.74	0.75	0.75	0.74	0.74	0.75	0.75	0.74
Recall	0.65	0.65	0.65	0.65	0.66	0.65	0.65	0.66	0.65	0.65
F1 score	0.69	0.70	0.69	0.70	0.70	0.69	0.69	0.70	0.70	0.69
Average F1 score	0.70									

Table 48: Multinomial Naïve Bayes results improved pre-processing

Top-k	Precision	Recall	F1-score
5	0.439	0.478	0.457
10	0.487	0.518	0.502
15	0.495	0.540	0.516
20	0.507	0.567	0.535
25	0.506	0.569	0.536
30	0.514	0.580	0.545
50	0.518	0.590	0.552
100	0.535	0.613	0.571
200	0.572	0.647	0.607
300	0.574	0.650	0.609
400	0.617	0.684	0.649
500	0.569	0.670	0.615
600	0.589	0.671	0.628
700	0.600	0.684	0.639
800	0.599	0.681	0.638
900	0.603	0.682	0.640
1000	**0.606**	**0.684**	**0.642**
2000	0.59157	0.655	0.622
3000	0.582	0.593	0.587
4000	0.577	0.569	0.573
5000	0.574	0.564	0.569

Table 49: Fuzzy Fingerprint $\mu 1(i)$ results improved pre-processing

Top-k	Precision	Recall	F1-score
5	0.440	0.478	0.458
10	0.462	0.483	0.472
15	0.479	0.508	0.493
20	0.490	0.538	0.513
25	0.491	0.544	0.516
30	0.493	0.548	0.519
50	0.496	0.550	0.521
100	0.521	0.581	0.849
200	0.552	0.617	0.583
300	0.562	0.636	0.597
400	0.578	0.660	0.616
500	0.583	0.663	0.620
600	0.584	0.664	0.621
700	0.592	**0.674**	0.630
800	0.592	0.673	0.630
900	0.594	0.672	**0.631**
1000	**0.595**	0.673	**0.631**
2000	0.588	0.660	0.622
3000	0.589	0.660	0.623
4000	0.587	0.652	0.618
5000	0.591	0.649	0.619

Table 50: Fuzzy Fingerprint $\mu 2(i)$ results improved pre-processing

The results from the previous models built and the results from the models based on the new pre-processing do not differ much. For the standard Naïve Bayes the total F-1 score goes up by 0.01 point, from 0.65 to 0.66. The same goes for the multinomial Naïve Bayes model, the score goes up from 0.69 to 0.70. There are no major increases or decreases in the precision and recall scores per class for the Bayesian methods. The results per class are situated in appendix 8.7 Results classifier 4. The Fuzzy Fingerprint algorithm performs a little worse than the model that does not use negation and emoticons. The highest achieved score stems from the $\mu 1(i)$ fuzzification function and achieves an F-1 score of 0.642 compared to 0.648 in model 3. For the $\mu 2(i)$ fuzzification function the F-1 score peaks at 0.631 compared to 0.635 in model 3.

5.5.2 Conclusion classification model 4

The improved pre-processing does not provide a significant increase in scores for all the three models. The Bayesian models perform slightly better, but the Fuzzy fingerprint method shows a slight decrease in performance. There are no major differences observed between the precision and recall scores in the sentiment classes. In conclusion the improved pre-processing provides a little better results but no major increase is noticed, thus the new pre-processing is only a very minor improvement for the Bayesian models.

5.6 Model 5- Penalized Multinomial Naïve Bayes

As the improved pre-processing only increased the F-1 scores by 0.01 another model is build that combines techniques learned from the Fuzzy Fingerprint method and the best performing algorithm. The last classification model tries to improve the Multinomial Naïve Bayes algorithm by penalizing the conditional probabilities based on the Inversed Document Frequency (IDF). It seems a good approach to penalize words that occur in more than one class as the uniqueness of a word for classification decreases, and thus the potential of correct classification based on that word. There are some other researches that use this method to increase the results of the Multinomial Naïve Bayes algorithm (Mohod, Dhote, & Thakare, 2015; Rennie, Shih, Teevan, & Karger, 2003; Win, Thu, & Tun, 2017) it thus seems a good approach to try and implement this method. The working of the adapted Multinomial Naïve Bayes algorithm is shown in Figure 20: Adapted multinomial Naïve Bayes approach on the next page. The code for this approach can be found in Appendix 8.17 Classification code IDF Multinomial Naïve Bayes.

Again the data is imported and stored in a dataframe, when it is stored the data is split into its three sentiment classes. The data is thereafter shuffled to avoid bias when running the algorithm multiple times. The data is then split for 10-fold cross validation as in the previous models. The first loop is initiated and first the prior probabilities are computed based on the total amount of instances of a sentiment class apparent in the training data. The method then extracts all unique words that are available in the training data of the three sentiment classes and stores them. Then the conditional probability for each unique word is calculated for each sentiment class by looking at the number of times it appears in the training data and the vocabulary size (see chapter 4.2.2). Then the number of times the unique word appears in a sentiment class (IDF) is calculated and stored, the probabilities are then penalized by this IDF measure. If a word appears only in 1 class the conditional probability is multiplied by 1, if it appears in two classes it is multiplied by $2/3^{rd}$ and if it appears in all three classes it is multiplied by $1/3^{rd}$. The results are then stored in a dataframe which serves as a probability library, there are in total three libraries one for each sentiment class. Then the test Tweets are initiated and split into separate words, for each test Tweet the words are looked up in the libraries and the conditional probabilities are summed. If a word does not appear in the library the la place smoothing of 1 is used (see chapter 4.2.2) to avoid 0 probabilities. When all words are passed through the library the likelihood is calculated for each sentiment class. Classification is then based on the highest likelihood value for a sentiment class.

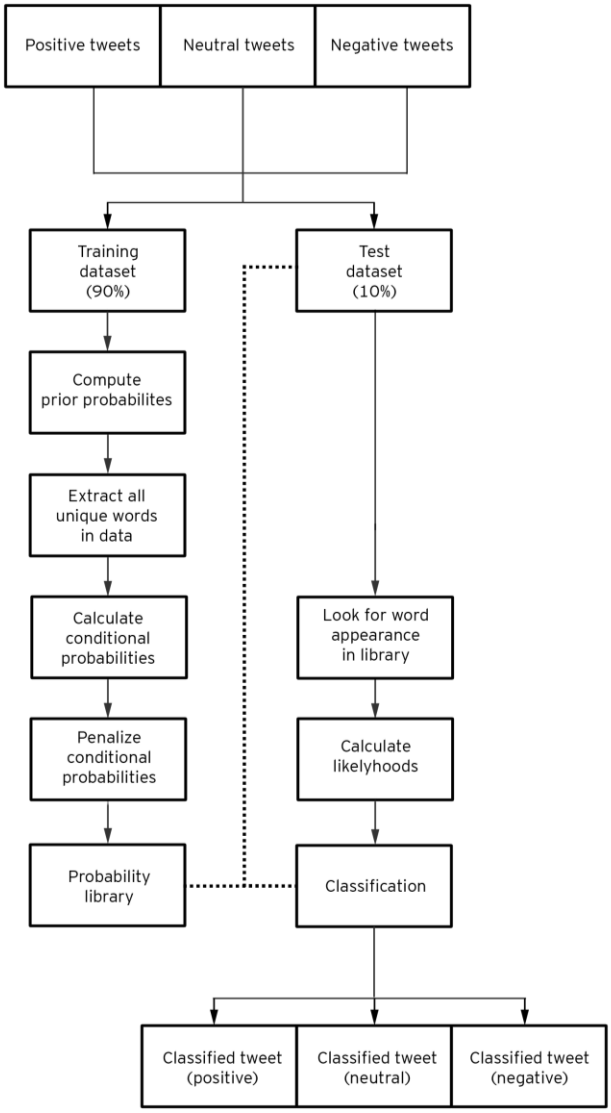


Figure 20: Adapted multinomial Naïve Bayes approach

5.6.1 Evaluation classification model 5

The results from the last classifier are good and establish a major increase in F-1 score (0.17) compared to the standard Multinomial Naïve Bayes model. Precision and recall scores have gone up significantly, about 0.2 and 0.15 respectively. The full table in included in Appendix 8.8, it is observed that the most troublesome class "Positive" is still the worst performing class. Precision scores for the "Positive" class are on average around 0.99 but the recall scores are lower, between 0.4 and 0.55 per run. The precision and recall scores for the other two sentiment classes are high and thus provide good results. The result are shown in Table 51: Adapted Multinomial Naïve Bayes results on page 56.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Precision	0.95	0.91	0.94	0.96	0.95	0.93	0.93	0.95	0.95	0.96
Recall	0.79	0.81	0.78	0.79	0.80	0.79	0.79	0.80	0.83	0.82
F1 score	0.86	0.86	0.85	0.86	0.87	0.85	0.85	0.87	0.89	0.89
Average F1 score	0.87									

Table 51: Adapted Multinomial Naïve Bayes results

5.6.2 Conclusion classification model 5

In conclusion this classifier works very well and handles the imbalanced data better than any of the other classifiers used. It provides better precision and recall scores for all the classes and especially for the "Positive" sentiment class. The penalties used to make unique words weigh more in the classification process is thus a valid and useful improvement to the Multinomial Naïve Bayes classification process.

6 Influence analysis

Chapter 6 researches the influence of the earlier established greenwashing events identified and the sentiment expressed on Twitter. A sentiment trend is constructed by using the best performing classification method. Here it is explained how this sentiment trend is constructed and the final sub research question is answered: What is the influence of greenwashing practices on the public opinion?

6.1 Sentiment trend

The sentiment trend that is used to measure the influence of the greenwashing events on the opinion expressed on Twitter is made out of two parts, the positive Tweets and the Negative Tweets. To classify the Tweets the best performing classifier is used, this is the adapted Multinomial Naïve Bayes classifier (see chapter 5.6). The improved pre-processing is used to clean all the Tweets and then the manually labeled Tweets are used as training data, the remaining Tweets are used as test data and the sentiment is predicted. After the prediction of the sentiment is completed the sentiment labels are added to the data and a sentiment trend can be constructed. First the "Positive" and "Negative" predicted sentiment classes are visually represented in Figure 21: "Positive" and "Negative" predicted sentiment. This figure is also included in Appendix 8.9 for readability.

Important to note is that the predicted sentiment is dependent on the results of the classifier used. The penalized Multinomial Naïve Bayes model achieves an F-1 score of 0.86 with an average precision of 0.94 and recall of 0.80. Meaning that it does a good job at predicting the correct classes out of the total number of Tweets in a class, meaning that the "usefulness" of the method is good. It performs a little less at predicting the correct sentiment classes in relation to the total amount of instances of that specific class, the "completeness". Meaning that the method is good at predicting the true positives but it also predicts a some number of false negatives. The results of the classification are thus by no means perfect, but good enough to construct a reliable sentiment trend.

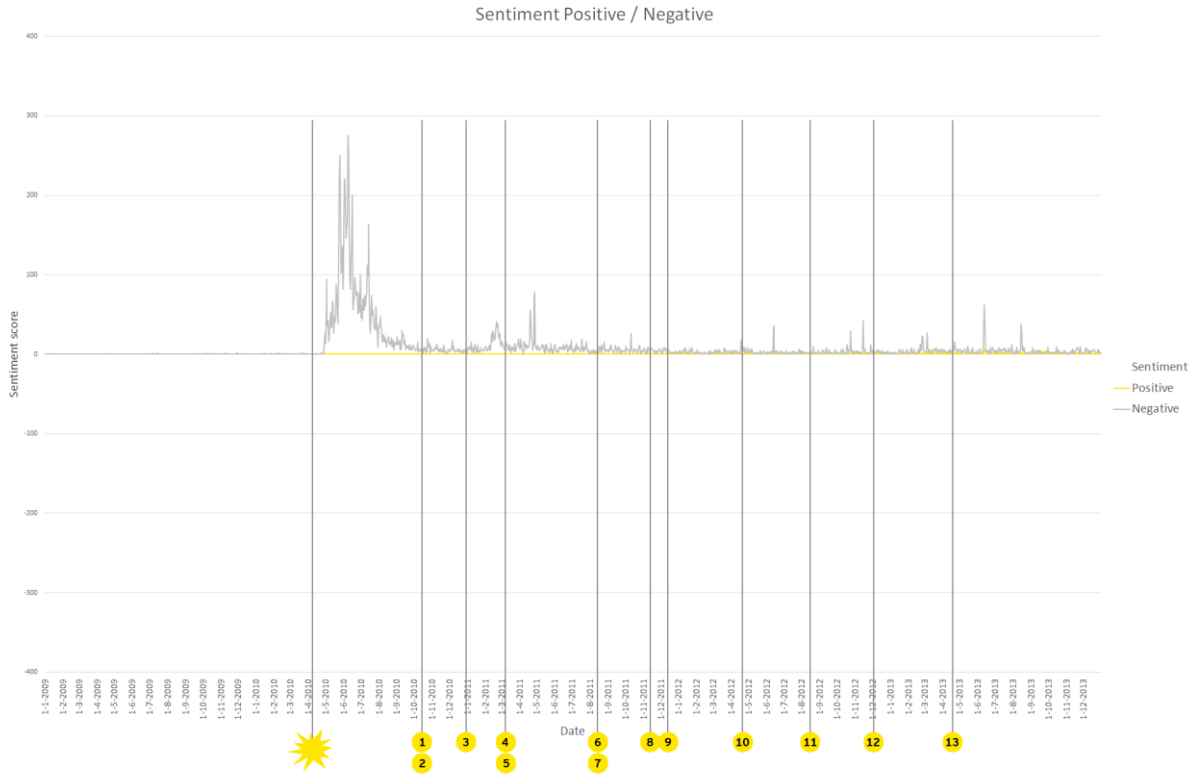


Figure 21: "Positive" and "Negative" predicted sentiment

The precision and recall scores for the "Positive" and "Negative" classes are important to consider when making a statement of the reliability of the predicted sentiment trend. For the penalized Multinomial Naïve Bayes classifier the precision and recall scores are around 0.99 and 0.45 for the "Positive" and 0.91 and 0.99 for the "Negative" class, these figures stem from appendix 8.8 Results classifier 5. The classifier does a reasonable job at predicting the true positives for the "Positive" and "Negative" class, but also predicts a lot of false negatives for the "Positive" class. Meaning that the amount of actual "Positive" classes is probably somewhat higher.

The sentiment trend is constructed by assigning a value of 1 for every "Positive" classified Tweet, a 0 for every "Neutral" predicted Tweet and -1 for every "Negative" prediction. The scores of the dates are then summed and plotted in a graph, see Figure 22: Predicted sentiment trend (this figure is also included in Appendix 8.9). The relationship between the events identified and the occurring drops and peaks are fist visually identified, thereafter the tweets that correspond to the drops and peaks are checked in the data to see if they relate to the identified greenwashing event or any other events.

What immediately becomes clear when visually checking the sentiment trend is that there is not much Twitter activity before the oil spill, then the sentiment drops significantly as the Deepwater Horizon oil spill occurred. There are also some drops observed a few weeks after some commercials aired, see for instance the drop in the sentiment trend after commercial 3. When checking the Tweets around those dates it is observed that the commercials are sometimes mentioned, but the majority of the Tweets do not specifically mention the commercials. Meaning that besides the commercials there are other influences that increase the "Negative" sentiment expressed on Twitter. This becomes increasingly clear when multiple drops are checked, there are references to other events such as the Whitehouse approves BP drilling in the Gulf again which increases the "Negative" sentiment expressed on Twitter. There is thus no clear relationship observed between the identified greenwashing events and the sentiment trend.

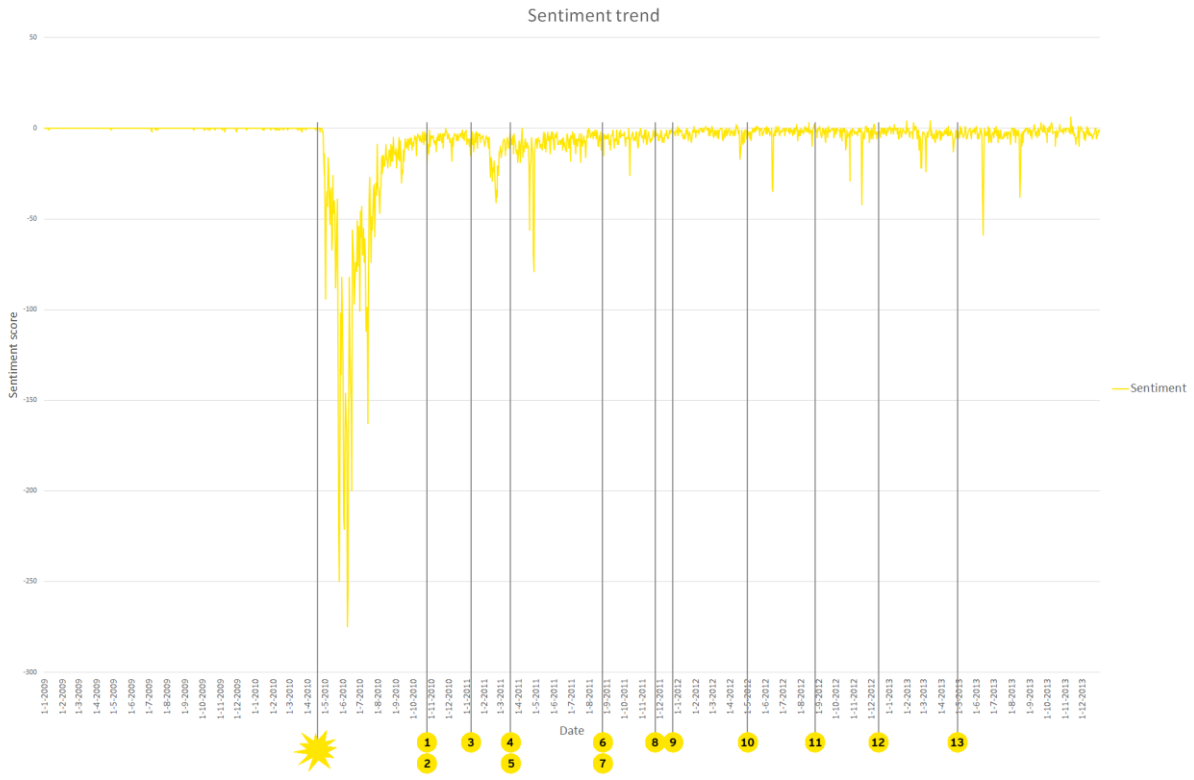


Figure 22: Predicted sentiment trend

6.2 Sentiment relation to stock prices

Multiple types of research investigated the relationship between Twitter sentiment and stock prices (Mittal & Goel, 2012; Ranco, Aleksovski, Caldarelli, Grčar, & Mozetič, 2015; Si et al., 2013) for predicting stock prices. These studies show that there are relationships between the stock prices and for instance moods expressed on Twitter (Mittal & Goel, 2012; Si et al., 2013) and dependence of stock price returns and Twitter sentiment about companies (Ranco et al., 2015). Although this relationship is of no major concern for the thesis it is interesting to analyze as stock data is publicly available. First the data needs to be gathered and validated before a relationship can be investigated.

6.2.1 Data acquisition

BP stock data from the American ADS New York stock exchange is downloaded from the website of BP (BP, 2017), this data is used for comparing the sentiment trend to the prices of BP stocks. The data on the BP website is set to the range of 01-01-2009 until 01-01-2014.

6.2.2 Data understanding

The data extracted from BP's website contains the following items: date, stock price opening, stock price closing, stock price high and stock price low. The excel file contains 1.254 entries with stock price data from 2 January 2009 till 31 December 2013. Table 52: BP stock prices dataset shows the dataset retrieved from the website (BP, 2017). This data is ready for analysis because the information needed is (stock prices and dates) are present within the dataset.

Date	Close**	Open**	High**	Low**
31 Dec 2013	48.61	48.38	48.65	48.31
30 Dec 2013	48.2	48.2	48.33	48
….	….	….	….	….
….	….	….	….	….
02 Jan 2009	48.45	47	48.62	46.96

** figures in USD

Table 52: BP stock prices dataset

The BP stock price data has 1.254 entries ranging from 2 January 2009 to 31 December 2013. The stock prices are checked for any outliers by plotting the median and the corresponding quartiles of the stock prices. A data point which lies outside of the 1.5 interquartile range can be considered an outlier (Dawson, 2011), in this case there are many data points outside this range as can be seen in Figure 23: Boxplots BP stock data. These points are not considered outliers because the stock market is a highly volatile one (Kang, Ratti, & Yoon, 2015), the data points outside the whiskers of the plot are tightly grouped and there are no data points which lie way beyond these groups.

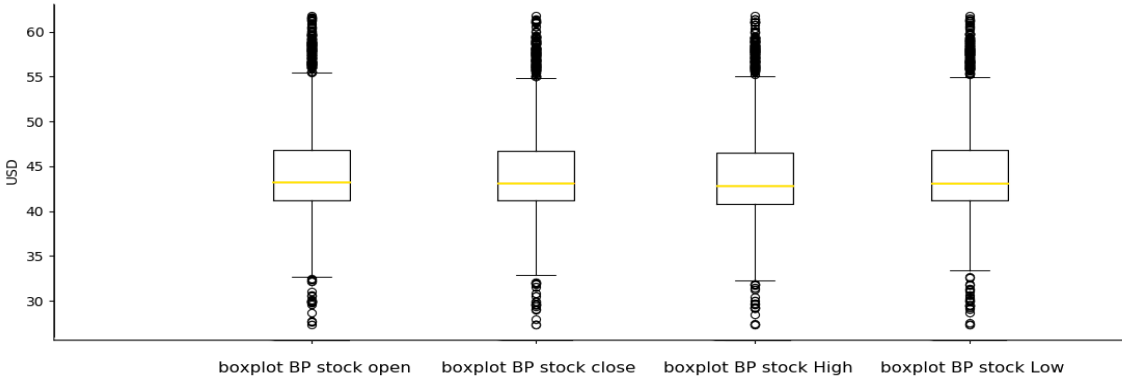


Figure 23: Boxplots BP stock data

6.2.3 Data Validity

The BP stock data is checked for validity by accessing the data provided by Nasdaq (Nasdaq, 2017) and there are no differences to be observed between randomly checked data points of the two parties. It is thus assumed that the stock prices are presented correct, furthermore the dataset shows no missing values or any other inconsistencies. It is thus judged to be proper for analysis and comparison.

6.2.4 Relationship sentiment and stock prices

From the data acquired the opening stock price is used because it includes orders that have accumulated overnight due to after hour trading and investor valuation changes (Investopedia, 2018), in contradiction to the closing price. The relationship between stock prices and the sentiment trend is shown in Figure 24: Relationship sentiment trend and stock price (see also appendix 8.9). The relationship between the stock prices and the sentiment trend is first visually checked, when there appears to be a relationship between the peaks and drops in both trends the correlation is measured. There is similarity observed between the major sentiment drop after the oil spill and the drop in the stock price. Besides the one similar drop there is no clear relationship to be found visually between other sentiment drops and peaks that correspond to drops and peaks in stock prices. Several large drops such as the drop after greenwashing event 3 to 5 or 13 show no related drops in the stock price. Also when sentiment appears to be relatively stable between greenwashing events 9 and 10 there is a slight increase in stock prices instead of a stable stock price trend. This concludes that there is no clear relation between the stock price and the data used in this project.

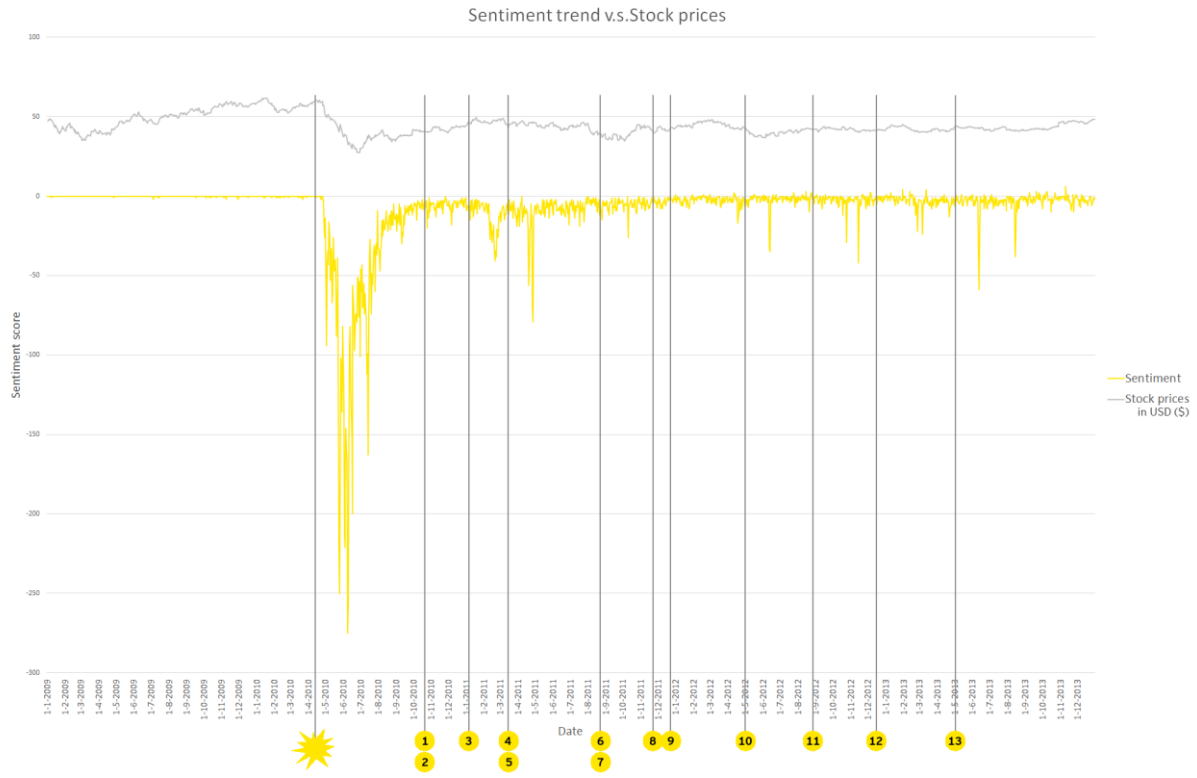


Figure 24: Relationship sentiment trend and stock price

7 Conclusion

The last chapter presents the conclusions of the project. Also it reflects on the results, the methods used and the limitations of the project. First a summary is given that briefly describes the project and its most important aspects. Then a discussion is presented that reflects on the arguments that underline the outcomes of the project, but also the opposing view. Subsequently the actions that can be taken in future work are elaborated on and the limitations of the project are discussed. The last part of this chapter presents the recommendations for the company.

7.1 Summary

The energy sector is a good example of an area where a lot of greenwashing practices are used to create better company and sector pictures in the minds of the public. As these practices hinder more eco-friendly investments and innovations it is important to research the effects of Greenwashing on the public. As there is almost no knowledge about the effects of Greenwashing, this thesis tries to research those effects. In this case the public opinion is measured by using the opinions expressed on Twitter. A sentiment analysis is conducted to show the relationship between identified greenwashing practices and the sentiment expressed.

First there needs to be some proof about events that are considered greenwashing. The first research question relates to the identification of these events: How can greenwashing events be identified that BP used concerning Deepwater horizon? In this case the concept of framing is used to analyze the commercials that are aired by BP after the Deepwater Horizon oil spill. The analysis shows that there are in total 13 commercials aired and classified as greenwashing due to the deceiving character of these commercials. These commercials do not represent the full facts available, episodic and didactic frames are used to "teach" the audience and leave them with an incomplete and narrow view toward the effects of the oil spill.

The second research question regards the data and the attributes needed to measure the influence of greenwashing practices on the public opinion. Data is gathered from the Twitter webpage ranging from before and after the oil spill, a range from 01-01-2009 till 01-01-2014 is used to capture the sentiment before and after the oil spill. The data extracted includes the Tweet account, the actual Tweet, and the date. Sentiment labels need to be added to perform sentiment analysis and finally make the sentiment trend,

As the BP Tweets do not have any sentiment labels the third research question relates to the ground-truth that is used for the classification of sentiment: How can a ground-truth be established that supports the sentiment expressed by the public opinion? In this case a manual labeling process is designed and a protocol is made. Three people are labeling around 1/3rd of the Tweets based on the protocol and the results are compared. The differences in labeled Tweets are then discussed and corrected based on the majority vote. This process ensures accurate sentiment labels for the extracted Tweets. These Tweets with accompanying labels are then used as the training and test data for the classifiers.

When the Tweets are accompanied with sentiment labels it needs to be processed in some way to classify the remaining Tweets. The fourth question relates to how this sentiment expressed can be measured. In total there are four methods used, all based on supervised learning: the standard Naïve Bayes algorithm, the Multinomial Naïve Bayes algorithm, the Fuzzy Fingerprint Algorithm and the penalized Multinomial Naïve Bayes algorithm. Although the final results that stem from the penalized Multinomial Naïve Bayes algorithm are good for all sentiment classes, the results of the other methods show that these methods have trouble dealing with imbalanced data because they perform badly for

the "Positive" sentiment class. Besides the imbalanced data the structure of a tweet also appears problematic for the other methods as it is hard to find sentiment class specific words.

The final sub question relates to the influence of the identified greenwashing practices on the public opinion. The results of the final model made are not fully reliable (F-1 score of 0.86) but good enough to make accurate claims, the real sentiment trend is probably a little more positive. The trend shows almost no real relationship to the identified greenwashing practices and increases or decreases in sentiment besides the massive drop directly after the oil spill. Because there are no fluctuations present in the sentiment trend near the identified greenwashing events it is concluded that these events have no clear relationship with the sentiment expressed on Twitter.

After the established sub questions the main research question can be answered: How can sentiment analysis be used to measure the influence of BP's greenwashing practices on the public opinion? Sentiment analysis can partially be deployed for measuring greenwashing practices if the results of the classifier are more accurate. It can only aid partially as the sentiment trend is never able to state any causal relationship due to the hard task of making a classifier 100% accurate and the influence of events that are outside of the scope of a sentiment analysis. In this project the maximum reached F-1 score is 0.86 out of 1 and is thus not fully reliable but reliable enough to state claims. Besides sentiment expressed online there is also sentiment which is not expressed online and that information is excluded from the analysis, making sentiment analysis only a substantiated part of the proof about the effectiveness of greenwashing events on the public opinion. What sentiment analysis can do is show increases or decreases in online sentiment in relation to certain identified events with relatively low effort as most of this data is publicly available. It appears that using supervised machine learning methods to establish such a sentiment trend is not the ideal way because the training of the algorithms is topic specific and very time-consuming. A more general approach to construct a sentiment trend might prove more useful and is less time consuming when applied for multiple problems. In conclusion, sentiment analysis can aid the discussion of the effectiveness of greenwashing practices as it can provide a relationship between greenwashing events and online opinions, but it can never show any causal relation.

7.2 Discussion

This thesis is by no means a project that tried to establish any causal relationship as the sentiment expressed online on Twitter does not fully represent the public opinion. It is known that there is a lot of publicly available information on Twitter but it represents only a part of the available opinions as there are also other platforms such as Facebook, (micro)blogs etc. that hold sentimental value that can be taken into account. In this case only the data from one social media platform is used as the structure of different platforms differs and might require multiple methods to analyze them, there is chosen to work on only one method for one platform due to the limited time of the project. Online opinions can be extracted from the internet and are thus a cheap option to measure, opinions that are not expressed online can only be gathered by questionnaires and interviews which are more time consuming and expensive. The objective was to use this easily accessible information and try to find a relationship between the sentiment expressed and the identified greenwashing events.

The results of the project are quite good and are in line with the set expectation, only the recall measure of the "Positive" sentiment class performs less than expected. In general it can be said that based on the literature read that all classifiers except the adapted Multinomial Naïve Bayes are expected to provide better results, many papers have used cleaner and more equally distributed data which partly explain their better performances as this project deals with imbalanced data. Also many papers state accuracy only but in this paper the F-1 score is used as it provides more insight into the classified results as it takes precision and recall into account. There are several mechanisms that attribute to the mediocre results of these classifiers, the first is the limited availability of "Positive" Tweets that make it hard to accurately identify unique features for this sentiment class. More training

data for this class needs to be available to accurately predict the sentiment classes, the distribution between the sentiment classes should preferably be as equal as possible to avoid imbalanced data. Second, the overall amount of training data should be increased as +- 6500 Tweets that can serve as training data is too few to precisely learn the algorithms. Thirdly the "Neutral" sentiment class holds the most data points, classification is easier if these neutral Tweets are deleted from the dataset and the classification problem becomes a binary problem. Therefore it seems reasonable to learn an algorithm to detect these neutral tweets and delete them before the Tweets with actual sentiment are used for learning an algorithm. Based on the results and the methods used it seems a good approach to also explore unsupervised learning methods or lexical methods for sentiment analysis to avoid a labor-intensive labeling process which hinders the applicability of the supervised learning methods due to the limited availability of data and it topic specific required data.

The use of sentiment analysis can show the relationship between sentiment trends and identified greenwashing events if the results are accurate over all sentiment classes. Sentiment analysis can only partially show the relationship between greenwashing events and sentiment expressed on Twitter as the identification of greenwashing events is subjective in some cases. In this case TV commercials are used as the greenwashing events. TV commercials are aired and can be viewed by many people through the TV and internet, but there are also other events that can be greenwashed. Consider the greenwashing of a product by stating vague green terms such as eco-friendly and ambiguous information on the packaging of a product. It is much harder to measure the influence of these kinds of greenwashing practices as they are not as public as the TV commercials. There might be some expressed sentiment towards these practices expressed online, but unless these practices really catch the eyes of a lot of consumers the expectation is that it will not be measurable with data from social media platforms. It seems that sentiment analysis is mainly able to aid in measuring the effectiveness of greenwashing events that are able to publicly be viewed by many. Also it not always 100% clear if misstated information or the way information is presented is intentionally set-up this way (see chapter 2.2.1). It is thus also not 100% clear if events identified are actually greenwashing or well-intended mistakes. Furthermore the sentiment trend might show a relationship but it is never able to state any causal relationship because there might be other factors outside the identified greenwashing events influencing the sentiment expressed. Thus sentiment analysis can only partly aid the discussion about the effectiveness of greenwashing events.

7.3 Future work

In this thesis four different methods are used of which two are quite common approaches for sentiment analysis, namely the standard Naïve Bayes and the Multinomial Naïve Bayes methods. The Fuzzy Fingerprints method has, to my knowledge, not been used for sentiment analysis and is thus a novel approach. Although the FFP method does not perform the best it uses some interesting procedures that may be applied to other methods as well. The FFP method uses the top-k most frequent words and penalizes their rank based on their appearances in other sentiment classes, a similar approach was therefore adopted for the Multinomial Naïve Bayes method. Here the probabilities calculated are also be penalized based on the inversed document frequency to make a more accurate classification. If tackling one of the assumptions of the Bayesian methods provides such increasingly better results it seems reasonable to explore more options to increase the classification of the Bayesian methods even further. Rennie et al., 2003 show some of these poor assumptions of the Bayesian methods such as the favor of the Bayesian method to favor classes with more data points and provide possibilities on how to resolve them. It is believed that these adaptions might provide even better results, especially for imbalanced data.

In the adapted Multinomial Naïve Bayes algorithm there is now chosen to use a simple metric for the penalties for the conditional probabilities. As there are three sentiment classes the penalties are 1, 2/3rd and 1/3rd of the original conditional probabilities based on the IDF scores. These penalties might

not be optimal, experiments need to be conducted with different penalty values to seek the optimal values.

Another issue that becomes apparent is what happens when words in a Tweet are not apparent in the training data for the Bayesian methods or in the fingerprint for the FFP method. In the Naïve Bayes methods the probability amounts to 0 which is problematic as the whole probability for a sentiment class then amounts to 0 (see chapter 4.1.1 and 4.1.2). A solution is to use alpha smoothing for words that do not appear in the training data which prevents this issue. In the FFP method the T2S2 scores become lower as words in a Tweet do not appear in a sentiment class. A solution that can be implemented for all methods could be to make a dictionary of synonyms, if a word does not appear in the training data or fingerprint its synonyms are used for classification instead. This might reduce the chances of having 0 probabilities as the feature range increases. This could be of great help especially for Twitter data as there not many characters in a Tweet and missing one word in the training data could majorly influence the outcome of the classification. Important to note is that the computation time will probably go up and that there must be some available resource containing words and its synonyms for implementing this into the methods.

7.4 Limitations

No project is without limitations. The most important limitation of this project is the fact that only Twitter data is used to measure the online opinion. More opinions are available from other social media platforms but also opinions that are not expressed online should be taken into account. Due to time limitations of the project there is chosen to only use Twitter data, to get the full picture these other opinions need to be analyzed as well.

Another limitation that stems from Twitter itself is that Tweets are gathered from all around the world, there is now filtered on English language only. The intention of the project was to analyze American sentiment only as other languages besides English fall outside the scope of this thesis. There is way in the Twitter advanced search to search for specific GEO-locations, but unfortunately not many Twitter users provide that information. This means that the Tweets analyzed can stem from every country of the world if they are written in English.

In this project there is chosen to use a specific use case regarding the Deepwater Horizon oil spill. The limitation of this project is that the severity of the case might influence the sentiment expressed on Twitter. After the oil spill it is likely that the overall sentiment has dropped and that it is thus hard to see any sentiment changes around the time of the greenwashing events. It might be better if the case is a little more subtle in term of environmental damage and severity to measure any increases and decreases in the sentiment expressed online.

7.5 Recommendations

The recommendations are written for the company specific. Within the company there is an interest to perform sentiment analysis for advisory related assignments such as: company reputation management, social media monitoring, advertisement impact measuring and measuring customer opinion and satisfaction.

Ernst and Young (EY) is interested in an application of sentiment analysis that can be applied to multiple cases and domains to reduce the amount of work that has to be done before sentiment analysis can be applied. This project has focused on one specific use case and uses machine learning algorithms that require labeled domain specific data for the learning of the algorithms. The company

is better suited with algorithms that can be deployed more broadly for more use cases without having to conduct a whole labor intensive labeling cycle for the training data. Fortunately there are other methods available for sentiment analysis such as unsupervised methods and lexical methods which are less domain specific and do not require the manual labeling of data. This also relates to the analysis of sentiment on other social media platforms, it is likely that an advisory assignment contains more than just the analysis of one social media domain. The algorithms used in this paper all have to be adjusted considerably before they can ben functionally applied to other social media domains due to the characteristics of the short Tweets, other social media platforms have different formats and outputs. Again a less domain specific approach, on that is for instance based on more general words that hold sentiment and are stored in a dictionary, can be applied more easily to other social media domains. Lexical methods might be more useful for application across multiple domains as they are based on pre-established dictionaries. It is assumed to be easier to adapt the lexical methods to multiple platforms than revising whole supervised machine learning methods and the data that is needed for learning the algorithms.

If EY considers to use one of the algorithms used in this paper then there are some recommendations for each of the methods and its application. The adapted Multinomial Naïve Bayes methods performs better than the other Bayesian methods and it is thus recommended to use this algorithm based on the results. The penalized multinomial method is applicable for media with relative short texts such as Twitter or Facebook, longer textual documents such as books or whole blogs that need analysis are not suitable for a Naïve Bayes approach. This stems from the fact that longer text has many words and the Naïve Bayes approach will most likely favor longer documents because they the probability of a word occurring in a longer text increases. The Fuzzy Fingerprint method can be used for the application of either short and longer texts (with some adjustment see chapter 4.1.3). If the objective is to use one classifier which requires minimal rework for different assignments than it seems useful to apply this method because of the multiple use cases that it has, although results are not as good as of the adapted Multinomial Naïve Bayes method.

The third recommendation relates to the use of sentiment analysis as a tool that can aid the company in identifying how the public thinks about a company or event. EY must consider that a sentiment trend that is established by an algorithm is by no means the whole story. This project shows that there may be relationships found between events that have happened and sentiment expressed but that there are also other influential events that can distort the sentiment trend. Sentiment analysis can be used to gather some first insights and then later on extend the investigation towards some increases or decreases in the sentiment trend. But as mentioned there are also other opinions which are not expressed online, these are also important to consider but expensive to gather. It is thus advised to first gather some preliminary results about the public opinion via sentiment analysis, this sentiment trend shows the peaks and drops within sentiment expressed online. This can then be used as a starting point for a more elaborate investigation of the sentiment regarding a company, the peaks and drops can be used as investigation points to determine events that contribute to increases and decreases in sentiment.

Reference list

Aggarwal, P., & Kadyan, A. (2014). Green Washing: The Darker Side of CSR. *International Journal of Innovative Research and Practices*, *2*(5), 22–35.

Asch, V. Van. (2013). Macro-and micro-averaged evaluation measures, 1–27. https://doi.org/10.3386/w19544

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. *IADIS European Conference Data Mining*, (January), 182–185. Retrieved from http://recipp.ipp.pt/handle/10400.22/136

Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., & Varma, V. (2012). Mining Sentiments from Tweets. *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, (July), 11–18. Retrieved from http://www.aclweb.org/anthology/W12-3704

Barrage, L., Chyn, E., & Hastings, J. (2014). Advertising As Insurance or Commitment? Evidence From the.

Beder, S., Greer, J., Bruno, K., The, G., Behind, R., & Environmentalism, C. (1998). Manipulating Public Knowledge, *7*(1), 1–6. https://doi.org/10.1007/BF02913282

Beyer, J., Trannum, H. C., Bakke, T., Hodson, P. V, & Collier, T. K. (2016). Environmental effects of the Deepwater Horizon oil spill: A review. *Marine Pollution Bulletin Journal*, *110*(1), 28–51. https://doi.org/10.1016/j.marpolbul.2016.06.027

BP. (2017). BP stock data. Retrieved from http://tools.bp.com/investor-tools/share-price-history/new-york

Braun, A. (2015). In the Wake of the Deepwater Horizon Oil Spill, Gulf Dolphins Found Sick and Dying in Larger Numbers Than Ever Before. Retrieved October 12, 2017, from https://response.restoration.noaa.gov/about/media/wake-deepwater-horizon-oil-spill-gulf-dolphins-found-sick-and-dying-larger-numbers-ever.

Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2013). Combining strengths, emotions and polarities for boosting Twitter sentiment analysis. *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '13*, 1–9. https://doi.org/10.1145/2502069.2502071

Brennan, K. (2013). *A Stakeholder Analysis of the BP Oil Spill and the Compensation Mechanisms Used to Minimize Damage*. UNIVERISTY OF SOUTH FLORIDA. Retrieved from http://www.usf.edu/business/documents/undergraduate/honors/thesis-brennan-katelyn.pdf

Burke, K. (1984). *ATTITUDES TOWARD HISTORY*. University of California Press.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). Crisp-Dm 1.0. *CRISP-DM Consortium*, 76. https://doi.org/10.1109/ICETET.2008.239

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Expert Systems with Applications Feature selection for text classification with Naïve Bayes. *Expert Systems With Applications*, *36*(3), 5432–5435. https://doi.org/10.1016/j.eswa.2008.06.054

Cherry, M. A., & Sneirson, J. F. (2012). Chevron, Greenwashing, and the Myth of'Green Oil Companies'. *Journal of Energy, Climate, and the Environment*, *3*.

Cherry, M., & Sneirson, J. (2011). *Beyond Profit : Rethinking Corporate Social Responsibility and Greenwashing After the BP Oil Disaster*. *Tulane Law Review* (Vol. 85). https://doi.org/10.2139/ssrn.1670149

Church, K., & Gale, W. (1999). Inverse document frequency (idf): A measure of deviations from poisson. *Natural Language Processing Using Very Large Corpora*, 121–130. https://doi.org/10.1007/978-94-017-2390-9_18

Collins, C., Steg, L., & Koning, M. (2007). Customers' Values, Beliefs on Sustainable Corporate Performance, and Buying Behavio. *Psychology & Marketing*, *24*(6), 555–567.

https://doi.org/10.1002/mar

Dahl, R. (2010). Greenwashing. *Environmental Health Perspectives*, *118*(6), 246–253. https://doi.org/10.1289/ehp.118-a246

Dawson, R. (2011). How Significant Is A Boxplot Outlier? *Journal of Statistics Education*, *19*(2), 1–13. Retrieved from www.amstat.org/publications/jse/v19n2/dawson.pdf

Delmas, M., & Burbano, V. C. (2011). The drivers of greenwashing. *California Management Review*, 38. https://doi.org/10.1016/0737-6782(88)90039-2

Delmas, M., & Toffel, M. (2008). Services, industry evolution, and the copetitive strategies of product firms. *Academy of Management Journal*, *51*(2), 315–334. https://doi.org/10.1002/smj

Enchanted learning. (2016). List of contractions.

Entman, R. (1993). Framing: Toward Clarification of A Fractured Paradigm. *Journal of Communication*, *43*(January), 51–58. https://doi.org/10.1111/j.1460-2466.1993.tb01304.x

Entman, R. (2004). *Projections of power: Framing news, public opinion, and US foreign policy*. University of Chicago Press.

Fiori, A. (2014). *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding: Revolutionizing Knowledge Understanding*. IGI Global.

Frick, W. (2014). Study: green advertising helped BP recover from the Deepwater horizon spill. *Harvard Business Review*, 2–5.

Furlow, N. E. (2010). Greenwashing in the New Millennium. *Journal of Applied Business and Economics*, *10*(6), 22–25.

Garg, S. K., & Meher, R. K. (2015). *Naive Bayes model with improved negation handling and n-gram method for Sentiment classification*. *Department of Computer Science and Engineering National Institute of Technology Rourkela*.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Processing*, *150*(12), 1–6. https://doi.org/10.1016/j.sedgeo.2006.07.004

Goffman, E. (1974). *Frame analysis: an essay on the organization of experience*. Harvard University Press.

Goguen, D. (2015). BP Oil Spill Lawsuits and Legal Issues. Retrieved October 12, 2017, from https://www.nolo.com/legal-encyclopedia/bp-oil-spill-lawsuits-legal-33442.html

Gross, K. (2008). Framing Persuasive Appeals : Episodic and Thematic Framing , Emotional Response , and Policy Opinion. *Political Psychology*, *29*(2).

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*(December 2014), 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Homem, N., & Carvalho, J. P. (2011). Authorship identification and author fuzzy "fingerprints." *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 180–185. https://doi.org/10.1109/NAFIPS.2011.5751998

Horiuchi, R., Schuchard, R., Shea, L., & Townsend, S. (2009). Understanding and Preventing Greenwash: A Business Guide. *Futerra Sustainability Communications*, (July), 1–41. Retrieved from http://www.bsr.org/reports/Understanding__Preventing_Greenwash.pdf

Huffington Post. (2013). History Of BP Ads Since The Gulf. Retrieved from http://www.huffingtonpost.com/2013/04/16/bp-oil-spill-ads-since-deepwater-horizon_n_3093185.html?slideshow=true#gallery/292273/2

Hurley, P. J. (2011). *A Concise Introduction to Logic*. Cengage Learning.

Husted, B., & De Jesus Salazar, J. (2006). Taking Friedman Seriously: Maximizing Profits and Social Performance. *Journal of Management Studies*, *43*(1), 75–91.

Indurkhya, N., & Damerau, F. (2010). *Handbook of Natural Language Processing*. CRC Press.

Investopedia. (2018). Opening price. Retrieved from https://www.investopedia.com/terms/o/openingprice.asp

Ispot. (2017). BP TV Commercials. Retrieved from https://www.ispot.tv/brands/dJG/bp

Jurafsky, D. (2012). *Multinomial Naive Bayes: a worked example*. Retrieved from https://www.youtube.com/watch?v=km2LoOpdB3A&t=31s

Jurafsky, D., & Martin, J. (2017). Naive Bayes and Sentiment Classification. *Speech and Language Processing Stanford University*. Retrieved from https://web.stanford.edu/~jurafsky/slp3/6.pdf

Kahneman, D. (2003). Maps of bounded rationality: Psychology for behavioral economics. *American Economic Review*, *93*(5), 1449–1475. https://doi.org/10.1257/000282803322655392

Kang, W., Ratti, R. A., & Yoon, K. H. (2015). The impact of oil price shocks on the stock market return and volatility relationship. *Journal of International Financial Markets, Institutions and Money*, *34*, 41–54. https://doi.org/10.1016/j.intfin.2014.11.002

Khan, F. H., Qamar, U., & Bashir, S. (2016a). A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. *Knowledge and Information Systems*, *51*(3), 851–872. https://doi.org/10.1007/s10115-016-0993-1

Khan, F. H., Qamar, U., & Bashir, S. (2016b). Multi-Objective Model Selection (MOMS)-based Semi-Supervised Framework for Sentiment Analysis. *Cognitive Computation*, *8*(4), 614–628. https://doi.org/10.1007/s12559-016-9386-8

Kibriya, A., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multimomial Niave bayes fir text categirization revisited. *Australian Conference on Artificial Intelligence (*, *3339*, 488–499.

Laufer, W. S. (2003). Social Accountability and Corporate Greenwashing. *Journal of Business Ethics*, *43*(3), 253–261. https://doi.org/10.1023/A:1022962719299

Lecy, J. D., & Beatty, K. E. (2012). Representative Literature Reviews Using Constrained Snowball Sampling and Citation Network Analysis. *SSRN Electronic Journal*, 1–15. https://doi.org/10.2139/ssrn.1992601

Liang, P.-W., & Dai, B.-R. (2013). Opinion Mining on Social Media Data. *2013 IEEE 14th International Conference on Mobile Data Management*, 91–96. https://doi.org/10.1109/MDM.2013.73

Lin, J., & Kolcz, A. (2012). Large-scale machine learning at twitter. *Proceedings of the 2012 International Conference on Management of Data - SIGMOD '12*, 793. https://doi.org/10.1145/2213836.2213958

Liu, B., Blasch, E., Chen, Y., Shen, D., & Chen, G. (2013). Scalable Sentiment Classification for Big Data Analysis Using Na ¨ ıve Bayes Classifier. *2013 IEEE International Conference on Big Data*, 99–104. https://doi.org/10.1109/BigData.2013.6691740

Liu, K., Li, W., & Guo, M. (2010). Emoticon Smoothed Language Models for Twitter Sentiment Analysis, 1678–1684.

Lustgarten, A., & Knutson, R. (2010). Reports at BP over years find history of problems. *Washington Post*.

Lydenberg, S. (2002). Envisioning socially responsible investing. *The Journal of Corporate Citizenship*, (7), 57–77.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. NY: Cambridge University Press New York.

Marujo, L., Carvalho, J. P., Gershman, A., Carbonell, J., Neto, J. P., & Matos, D. M. de. (2015). Textual Event Detection Using Fuzzy Fingerprints. *Intelligent Systems'2014*, *322*, pp 825-836. https://doi.org/10.1007/978-3-319-11313-5

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 41–48. https://doi.org/10.1.1.46.1529

Mcwilliams, A., & Siegel, D. (2001). Corporate Social Responsibility : A Theory of the Firm

Perspective Author (s): Abagail McWilliams and Donald Siegel Source : The Academy of Management Review , Vol . 26 , No . 1 (Jan ., 2001), pp . 117-127 Published by : Academy of Management Stable U. *The Academy of Management Review*, *26*(1), 117–127. https://doi.org/+

McWilliams, A., Siegel, D. S., & Wright, P. M. (2006). Corporate Social Responsibility: Strategic Implications. *Journal of Management Studies*, *43*(January), 1–18. https://doi.org/10.1111/j.1467-6486.2006.00580.x

Mendelssohn, I. A., Andersen, G. L., Baltz, D. M., Caffey, R. H., Carman, K. R., Fleeger, J. W., … Rozas, L. P. (2012). Oil Impacts on Coastal Wetlands: Implications for the Mississippi River Delta Ecosystem after the Deepwater Horizon Oil Spill. *BioScience*, *62*(6), 562–574. https://doi.org/10.1525/bio.2012.62.6.7

Michie, D., Spiegelhalter, D. J., Taylor, C. C., Michie, E. D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine Learning, Neural and Statistical Classification. *Ellis Horwood Series in Artificial Intelligence*, *37*(4), xiv, 289 . https://doi.org/10.2307/1269742

Milne, M., & Patten, D. (2002). Securing organizational legitimacy. An experimental decision case examining the impact of environmental disclosures. *Accounting, Auditing & Accountability Journal*, *15*(3), 372–405. https://doi.org/10.1108/MBE-09-2016-0047

Minanović, A., Gabelica, H., & Krstić, Ž. (2014). Big data and sentiment analysis using KNIME: Online reviews vs. social media. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, (May), 1464–1468. https://doi.org/10.1109/MIPRO.2014.6859797

Mitchell, L. D., & Ramey, W. D. (2011). Look How Green I Am! An Individual-Level Explanation for Greenwashing. *Journal of Applied Business and …*, *12*(6), 40–45. Retrieved from http://www.na-businesspress.com/JABE/mitchelll_abstract.html

Mitchell, R. (2016). *Web Scraping with Python: collecting data from the moder*. *Sante Publique* (Vol. 28). O'Reilly Media, Inc. https://doi.org/10.1017/CBO9781107415324.004

Mittal, a, & Goel, a. (2012). Stock Prediction Using Twitter Sentiment Analysis. *Tomx.Inf.Elte.Hu*, (June). https://doi.org/10.1109/WI-IAT.2013.48

Mohod, S. W., Dhote, C. A., & Thakare, V. M. (2015). Modified Approach of Multinomial Naïve Bayes for Text Document Classification, 196–200. https://doi.org/10.090592/IJCSC.2015.614

Mudinas, A., Zhang, D., & Levene, M. (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis. *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '12*, 1–8. https://doi.org/10.1145/2346676.2346681

Nadeau, L., Kaplan, M., Sands, M., Moore, K., & Goodhue, C. (2014). Assessing the impacts of the Deepwater Horizon oil spill on tourism in the Gulf of Mexico region. https://doi.org/10.1016/j.egypro.2016.11.209

Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and Accurate Sentiment Classification Using an Enhanced Naive Bayes Model. *Intelligent Data Engineering and Automated Learning – IDEAL*, 194–201.

Nasdaq. (2017). BP p.l.c. Common Stock Historical Stock Prices. Retrieved from http://www.nasdaq.com/symbol/bp/historical

No slang. (2017). Internet & text slang dictionary.

Norvig, P. (2016). How to write a spelling corrector.

OCEANA. (2013). *Oil Spills and Tourism : They Don't Mix*. Retrieved from http://rsbl.royalsocietypublishing.org/lookup/doi/10.1098/rsbl.2015.0596

Oppolecchia, E. L. K. C. (2010). The Greenwashing Deluge: Who Will Rise Above the Waters of Deceptive Advertising? *University of Miami Law Review*, *1*, 1–54. Retrieved from https://litigation-essentials-lexisnexis-com.ezproxy.lib.swin.edu.au/webcd/app?action=DocumentDisplay&crawlid=1&doctype=cite&docid=64+U.+Miami+L.+Rev.+1353&srctype=smi&srcid=3B15&key=af3bc6033a672e75d0fb

fc683e8b68d7

PC.net. (n.d.). Emoticons Fun Text-Based Expressions. Retrieved from https://pc.net/emoticons/

Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects. Retrieved September 25, 2017, from http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Plec, E., & Pettenger, M. (2012). Greenwashing consumption: The didactic framing of ExxonMobil's energy solutions. *Environmental Communication*, *6*(4), 459–476. https://doi.org/10.1080/17524032.2012.720270

Putman, N. F., Abreu-Grobois, F. A., Iturbe-Darkistade, I., Putman, E. M., Richards, P. M., & Verley, P. (2015). Deepwater Horizon oil spill impacts on sea turtles could span the Atlantic. *Biology Letters*, *11*(12), 20150596. https://doi.org/10.1098/rsbl.2015.0596

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The effects of twitter sentiment on stock price returns. *PLoS ONE*, *10*(9), 1–21. https://doi.org/10.1371/journal.pone.0138441

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)-2003)*, *20*(1973), 616–623. https://doi.org/10.1186/1477-3155-8-16

Reuters. (2012). Two years after BP oil spill, tourists back in U.S. Gulf. Retrieved from https://www.reuters.com/article/usa-bpspill-tourism/two-years-after-bp-oil-spill-tourists-back-in-u-s-gulf-idUSL1E8GP15X20120527

Rosa, H., Batista, F., & Carvalho, J. P. (2014). Twitter topic fuzzy fingerprints. *IEEE International Conference on Fuzzy Systems*, 776–783. https://doi.org/10.1109/FUZZ-IEEE.2014.6891781

Rosa, H., Carvalho, J. P., & Batista, F. (2014). Detecting a Tweet's Topic within a Large Number of Portuguese Twitter Trends. *3 Rd Symposium on Languages, Applications and Technologies (SLATE'14)*, 185–199. https://doi.org/10.4230/OASIcs.SLATE.2014.185

Sanders, N. (2011). Twitter Sentiment Corpus. Retrieved from http://www.sananalytics.com/lab/twitter-sentiment/

Scheufele, D. A., & Tewksbury, D. (2007). Framing, agenda setting, and priming: The evolution of three media effects models. *Journal of Communication*, *57*(1), 9–20. https://doi.org/10.1111/j.1460-2466.2006.00326.x

Sharma, A., & Dey, S. (2013). A boosted SVM based sentiment analysis approach for online opinionated text. *Proceedings of the 2013 Research in Adaptive and Convergent Systems on - RACS '13*, 28–34. https://doi.org/10.1145/2513228.2513311

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013). *Topic based Twitter Sentiment for Stock Prediction*. *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference* (Vol. 2). Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84907356594&partnerID=tZOtx3y1

Siegel, D., & Vitaliano, D. F. (2007). An empirical analysis of the strategic use of corporate social responsibility. *Journal of Economics & Management Strategy*, *16*(3), 773–792. https://doi.org/10.1111/j.1530-9134.2007.00157.x

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Stein, B. (2005). Fuzzy-fingerprints for text-based information retrieval. *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 05)*, 572–579. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Fuzzy-Fingerprints+for+Text-Based+Information+Retrieval#0

Talib, R., Hanif, M. K., Ayesha, S., & Fatima, F. (2016). Text Mining : Techniques , Applications and Issues, *7*(11), 414–418.

TerraChoice. (2010). *The Sins of Greenwashing - Home and Family Edition*. *Underwriters Laboratories*. Retrieved from http://sinsofgreenwashing.com/index.html

Thakkar, H., & Patel, D. (2015). Approaches for Sentiment Analysis on Twitter: A State-of-Art study. Retrieved from http://arxiv.org/abs/1512.01043

The Guardian. (2012). BP oil spill seriously harmed deep-sea corals, scientists warn. Retrieved from https://www.theguardian.com/environment/2012/mar/26/bp-oil-spill-deepwater-horizon

Thompson, K. (1968). Programming Techniques: Regular expression search algorithm. *Communications of the ACM*, *11*(6), 419–422. https://doi.org/10.1145/363347.363387

Twitter. (2017). The Search API. Retrieved October 24, 2017, from https://developer.twitter.com/en/docs/tweets/search/overview/basic-search

Vauday, J. (2013). The Greenwashing Machine : is CSR more than Communication The Greenwashing Machine : Is CSR More Than Communication ?

Vijayarani, S., Ilamathi, J., & Nithya, M. (2015). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16. Retrieved from http://www.ijcscn.com/Documents/Volumes/vol5issue1/ijcscn2015050102.pdf

Vos, J. (2009). Actions speak louder than words: greenwashing in corporate America. *Notre Dame JL Ethics & Pub. Pol'y*, *23*(2), 673. Retrieved from http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/ndlep23&section=30

Webb, T. (2011). BP's Gulf oil spill compensation fund is not independent, rules Louisiana court. Retrieved October 12, 2017, from https://www.theguardian.com/business/2011/feb/03/bp-gulf-spill-fund-not-independent-rules-court

Welford, R. (2013). *Hijacking Environmentalism: Corporate Responses to Sustainable Development*. Routledge.

Win, T., Thu, P., & Tun, Z. A. R. (2017). Sentiment orientation system of automotive reviews using multinomial naive bayes classifier at document level. *International Journal of Advances in Electronics and Computer Science*, (7), 11–15.

Yang, C. S., Chen, C. H., & Chang, P. C. (2015). Harnessing consumer reviews for marketing intelligence: a domain-adapted sentiment classification approach. *Information Systems and E-Business Management*, *13*(3), 403–419. https://doi.org/10.1007/s10257-014-0266-z

Zhang, H. (2004). The Optimality of Naive Bayes. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference FLAIRS 2004*, *1*(2), 1–6. https://doi.org/10.1016/j.patrec.2005.12.001

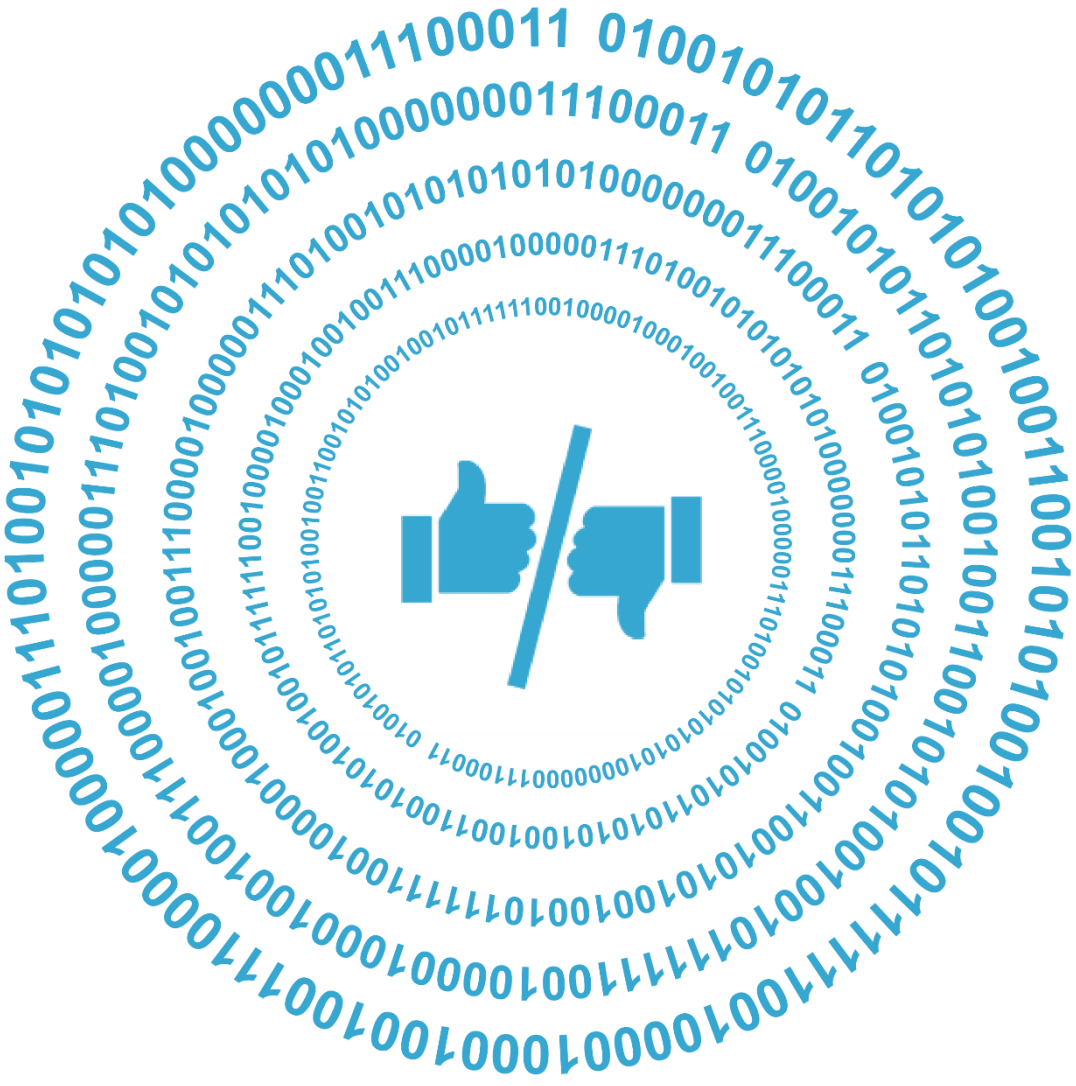
Zhang, P. (1993). Model Selection Via Multifold Cross Validation. *The Annals of Statistics*, *21*(1), 299–313.

Zuur, A. F., Ieno, E. N., & Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, *1*(1), 3–14. https://doi.org/10.1111/j.2041-210X.2009.00001.x

8 Appendices

8.1 Systematic literature review

"techniques for sentiment analysis"



Version: 2.1

Professor: Dr. A.M. Wilbik
Author: Dian J. Kroes
Student ID: 0753153
Date: 27-09-2017

Table of contents

1. Introduction 74

2. Research area 75

3. Search protocol 78

4. Analysis 82

5. Discussion and conclusions 92

6. References 94

7. Appendix 92

Appendix 1: Literature long list 92

Appendix 2: Literature short list 95

8.1.1 Introduction

This systematic literature review is made as a part of the master thesis for pursuing the masters title at the university of technology Eindhoven. This thesis researches the effects of "greenwashing" practices via sentiment analysis on the public opinion expressed on twitter. Sentiment analysis is a hot topic nowadays for companies that want to analyze textual data and derive opinions from this data. All sorts of data are available, data from ERP (enterprise resource planning) systems but also online data from Twitter and for instance product reviews. The application of sentiment analysis on this information can lead to interesting insights that might help companies to quickly determine the opinion of customers toward their company or towards products and attributes. Sentiment analysis can also be used to show trends and fluctuations in sentiment towards specific topics. To make a good overview of the methods that are available for executing sentiment analysis this systematic literature follows a systematic Kitchenham cycle. This literature review provides the different analysis methods that are applied nowadays within the sentiment analysis field and their characteristics such as performance and to which data the analysis is applied to.

The report is structured as follows. First the subject and the general research area are described in chapter 2. Thereafter the search protocol and search strategy are elaborated on in chapter 3. This results in an article pool, from this article pool a selection process is applied which results in the literature short-list for this systematic literature review. Subsequently, an analysis is executed and the results are made clear in a final analysis table. This table shows the clusters within the articles read. Finally, the conclusion and discussion mark the end of the literature review.

8.1.2 Research area

This part of the literature review aims to shed some light on the motivation of the research area, its application and what data science is. A short overview of the development of the data science field and what makes it so important for businesses these days is given. Thereafter a definition of the field and its parts is given to increase the understanding of sentiment analysis in its domain, last but not least the distinction between small and big data is elaborated on. The goal of this systematic literature review is to explore the current knowledge base of techniques available for sentiment analysis on big data and to gain insight into the different techniques can execute such an analysis.

8.1.2.1 Motivation research area

This research is conducted as a part of the master thesis and investigates the influence of "greenwashing" practices on the public opinion via sentiment analysis, which is a part of text mining. Greenwashing is the spreading of incomplete or false information toward the public regarding the company's environmental responsibility (Furlow, 2010). The practice of greenwashing leads the public to believe that a company or its products are more environmental responsible than that they actually are. Investigating the influence of such practices on the public sentiment is important because sustainability and ecological issues are currently a major societal issue and influence not only business decisions but also consumer decisions to buy products or not (Collins, Steg & Koning, 2007; Schwarz, Beloff & Beaver, 2002). It is therefore important to gain insight in the effects of these greenwashing practices. Such analysis has become possible because the introduction of social media resulted in a vast increase in data that contains a public sentiment, there are large quantities of this data available. The increased awareness for the environment and the large amounts of data available creates the opportunity to research the effects of greenwashing practices on the public opinion via sentiment analysis.

8.1.2.2 The data science field

Everyday 2.5 Quintillion bytes of data are generated (IBM, 2013), this amount equals almost 30.000 Gigabytes per second. Predictions even estimate that the world will produce a staggering 50.000 Gigabytes of data every second in 2018 (Walker, 2015). This amount of data production calls for new approaches to analyze and interpret data in a systematic manner. The term "*data science*" has been around since the 1960's, but in that time data science was just another word for data processing (Sundaresan, 2016). As time progressed and the computer era arose, data science evolved from being part of the statistical field to a field of its own. In the early 2000's data science was being formalized by Silicon Valley companies, eventually businesses caught on to this development as with most things that are used in Silicon Valley. This sparked the emergence of new forms of data and extensive use of analytics to achieve competitive advantage through fact-based decision making (Press, 2014). Ultimately this resulted in data science becoming an important field for businesses and the establishment of the data science journal in 2003 (Davenport, 2006).

But what is data science exactly? And what lies within its domain?

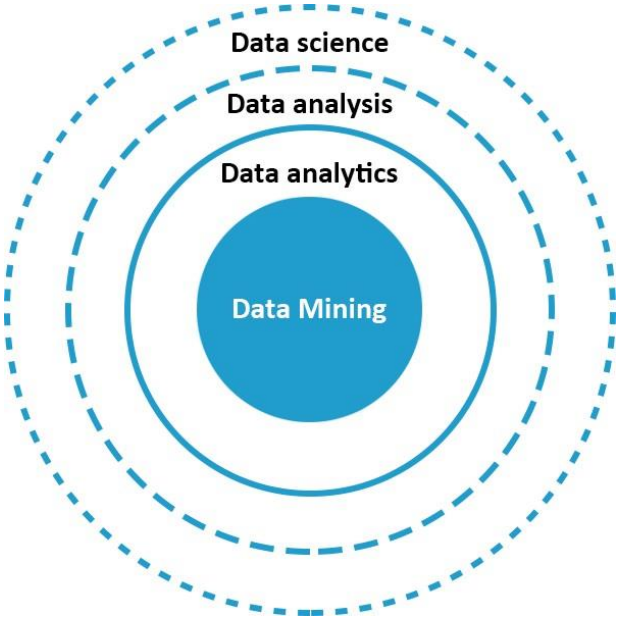


Figure 1: Domain data science

The data science domain consists of multiple levels and at the core of this domain lies data mining. *Data science* itself deals with all kinds of data, it involves principles and processes for understanding phenomena via automated analysis of data with the ultimate goal being improving decision making (Provost & Fawcett, 2013). *Data analysis* are the human activities that come along with trying to gain insight into the secrets that a dataset may hold. *Data analytics* concerns the automation of these insights and uses procedures such as queries and data aggregation to automate these insights. Last but not least is *data mining*, which are the actual tools and techniques used to analyze the data and extract hidden patterns in the data set. The topic of this systematic literature review lies within the data mining domain, in fact it is a part of the text mining research field. Sentiment analysis is also commonly referred to as "*opinion mining*", it is a technique that derives opinions toward a specific topic from data. The text mining research field uses different data mining techniques such as classification, and clustering approaches to derive specific information on a systematic basis from data sources which are too vast to be analyzed by humans (Talib, Hanif, Ayesha & Fatima, 2016).

8.1.2.3 Data groups

There are two groups of data: "Small Data" and "Big Data". The definition which classifies if data is either small data or big data is ambiguous, not only the volume of the data matters. The distinction between the groups is important because different analysis techniques are available for the different data groups. Although the definitions of small and big data differ quite extensively there are some common features which are commonly agreed upon (Madden, 2012). Data features are also called the 5 V's of data, these V's stand for Volume, Velocity, Variety, Veracity and Value (Demchenko, Grosso, de Laat, & Membrey, 2013). Small data relates mainly to data that is relatively small in volume and is produced within companies itself such as data stemming from Enterprise Resource Planning (ERP) systems. Big data comes from many sources, can take different forms, is produced in large quantities and is produced at phenomenal speeds. Therefore, big data is quite different than traditional data and might thus require different and more powerful analytical approaches and methods to analyze than small data. A small table is presented in table 1 that summarizes the differences between small and big data.

Category	Big data	Small data
Data source	Data emerged from outside a company/enterprise. Also stems from nontraditional data sources: - Social media - Sensor and log data - Device data - Video, images, etc	Data emerged from within a company/enterprise, stems from traditional sources: - Enterprise Resource Planning data (ERP) - Customer Relationships Management data (CRM) - Financial data
Volume	- Terabytes (10^{12}) - > Terabytes	- Gigabytes (10^{9}) - Terabytes (10^{12})
Velocity	- Often real time - Requires immediate response	- Batch - Near real time - Does not always require immediate response
Variety	- Structured - Unstructured - Multi-structured	- Structured - Unstructured
Veracity	- Lower data consistency - Lower data trustworthiness	- High data consistency - High data trustworthiness
Value	- Complex, advanced or predictive business analysis and insights	- business intelligence, analysis and reporting

Table 1: Data groups and its characteristics (Banafa, 2015; Demchenko, Grosso, de Laat, & Membrey, 2013)

This systematic literature review focusses on big data because the intended data for the sentiment analysis concerns data extracted from a social media platform, is unstructured and has most likely a lower data consistency and trustworthiness.

8.1.3 Search protocol

The search protocol of a systematic literature review is based on scientific databases that are available and a search strategy that depicts how the search for literature will be conducted. First the research question that this systematic literature review intends to answer is provided, thereafter the rationale for this literature review is discussed. Subsequently the search strategy and the selection criteria for literature are elaborated on.

8.1.3.1 Research question

The intent of this literature review is to provide an in depth overview of techniques that are available to perform sentiment analysis on big data. Therefore, the research question of this systematic literature review is the following:

Which techniques are currently used for performing sentiment analysis on big data?

8.1.3.2 Rationale for the review

This systematic literature review is made in advance of the graduation project of the author that is pursuing a master's degree in industrial engineering. This literature review should provide the reader and the author with a thorough understanding of the work done within the scope of this literature review. This review will identify work that is already done by other researchers regarding the topic of sentiment analysis on big data and it will provide an overview of the available techniques to execute such an analysis.

8.1.3.3 Search strategy

The search strategy consists of search terms that are derived from the research question and the resources used to find literature (Kitchenham, 2007). To ensure that the literature searched covers a wide spectrum of the available content 4 highly rated scientific databases are used. Each scientific database stores different content and are therefore used complementary to one another. Two databases focused towards computing and two multidisciplinary databases are used to ensure that the whole research area is covered.

1. ACM digital library

This database stores articles and other publications such as proceedings of conferences published by ACM (Association for Computing Machinery). ACM digital library focusses on computing proceedings (14.312 from 219 publishers), high impact computing research journals (45), computing magazines published by ACM (8) and books published by ACM and Morgan & Claypool (161.559 by 6034 publishers).
Source: ACM digital library

2. IEEE-Xplore

The IEEE-Xplore scientific database has a strong focus on computer science, engineering and electronics. It covers high impact journals and magazines (287), various books (2.383), conference publications (7.302) and other materials such as courses.
Source: IEEE-Xplore digital library

3. Springer

Springer Science + Business media is a widely known scientific database that publishes peer-reviewed journals (3.359), books (228.286) and other scientific documents with a broad focus on science, technology and medicine.
Source: Springer Link

4. Web of Science

Web of Science by Thomson Reuters is arguably the most widely known scientific database available today, it covers multidisciplinary range of research areas. It publishes conference proceedings (+160.000), notable scientific journals (13.200) and selected books (+60.000).
Source: Web of Science

Web of Science is used to search all high impact journals because of their high journal count, ACM digital library and IEEE-Xplore are used to find content covered in proceedings of conferences due to their high proceedings count. In total these four databases should cover most of the research area on sentiment analysis on big data.

8.1.3.4 Search Terms

The search terms used to derive relevant literature from the scientific databases are derived from the research question and the goal of this systematic literature review. The research question is divided into facets, these facets are then used to draw up a list of synonyms and alternative spellings to include in the search query (Kitchenham, 2007) see table 2. These search terms are then constructed into a search query by using search operators. This followed an iterative process where synonyms, operators and other criteria are used in different combinations to ensure that the documents found are in line with the goal of the systematic literature review and the research question. There is chosen to search only in the titles of scientific articles as searches in abstracts and other search field yield results that do not correspond with the goals of this literature review.

The research question "*Which techniques are currently used for performing sentiment analysis on big data?*" is divided into the different facets. These facets, their synonyms and other selection criteria are combined into a search query. This search query might differ slightly among the different scientific databases because each database requires a different syntax.

Number	Facet	Synonyms and alternative spellings	Source
1	Sentiment	Opinion	Thesaurus (http://www.thesaurus.com) Opinion mining and sentiment analysis (Pang & Lee, 2008)
2	Analysis	None	-
3	Big data	None	-

Table 2: research question facets

8.1.3.5 Selection criteria

To ensure that all documents found by the search query are in line with the research goal and the research question there must be selection criteria in place. These selection criteria provide a first scope to reduce the amount of documents found and provide better alignment with the literature review goal. The goal of this systematic literature review is to explore the current knowledge base of techniques available for sentiment analysis on big data and to gain insight into the different techniques can execute such an analysis.

1. Timeframe

To explore the current knowledge base of techniques available for sentiment analysis, a time frame selection criterion is used. Sentiment analysis gained popularity around 2001 (Pang & Lee, 2008) and has evolved extensively because of the development of more refined machine learning methods. Innovation in machine learning and the development of methods is ongoing practice and the research field evolves quite fast. As a first indication a timeframe of 10 years was set, this provided a too vast amount of available literature. The timeframe has been reduced to 5 years because of the vast amount of literature produced and the evolvement of the research field.

2. Accessibility

To fully understand the content of the article there are two criteria to be met. First the article must be published in English, otherwise language barriers might arise. Some articles are written in English but might contain other languages as well because the data that is used is not English.
Articles are excluded if this second language prevents the full understanding of the paper, otherwise this is considered not to be a problem. Second, the article is only included when the full text is available.

3. Goal of article

The goal of the documents found should be to execute and describe a sentiment analysis on textual big data according to one certain method. Papers must thus describe a practical approach to execute a sentiment analysis according to one specific method, and extract these sentiments based on textual big data.

8.1.3.6 Search results

Table 3 gives an overview of the search queries used and their results, in total 61 documents are found. A closer inspection to the found documents revealed that there are 8 documents also found in another search engine i.e. overlap, these are deleted from the literature long list resulting in 53 unique search results. ACM digital library results are limited to conference proceedings only because of the limited amount of journals compared to Springer and Web of Science. The same reasoning applies to IEEE-Xplore. Springer and Web of Science are both used for articles only because of their strong focus on scientific journals. Different search engines require a different search syntax. The main objective is to keep the search terms (“Sentiment”, “Opinion”, “Analysis”, “Big Data”) the same. Executing the exact same query in all used databases returns sometimes no hits at all, therefore slight variations of these search queries are deemed necessary to extract all relevant documents from the databases. The different search queries that are used are found via an iterative process of testing different queries in the database and looking at the results returned by the database. These results are quickly scanned through based on their titles and abstracts to reach a decision of the search query applied is deemed correct.

The literature long list contains 53 scientific documents (see appendix 1), selection criteria are applied after analyzing all the documents abstracts, methods and conclusions. The literature short

	ACM Digital library	IEEE-Xplore	Springer	Web Of Science
Search query	*Search in title:* *Matches all* "Sentiment" OR "Opinion" "analysis" + Search in title: *Matches any* "Big Data"	Search in Metadata document title: "Sentiment" OR "Opinion" NEAR "Analysis" AND "Big Data"	With all the words: ("Sentiment" OR "Opinion") NEAR "Analysis" AND "Big Data" Search in title: "Sentiment" "Opinion" "Analysis" "Big Data"	Search in title: ("Sentiment" OR "Opinion") NEAR "Analysis" AND "Big Data"
Publication years	2012-2017	2012-2017	2012-2017	2012-2017
Language	English	English	English	English
Filter	Proceedings only	Conference publications	Articles only	Citation indices only
Results	17	10	21	13
Unique results	53			

list that will be read and analyzed thoroughly is made after applying the selection criteria from paragraph 3.5, this results in 16 documents deemed fit for in depth review.

Table 3: Search query results

8.1.4 Analysis

The results that are found according to chapter 8.1.3 are thoroughly read and analyzed, in this chapter the results of the analysis are presented.

To accurately answer the research question and adhere to the goal of this systematic literature review, several analyses are executed. The research question concerns techniques that are used for performing sentiment analysis on big data. Several analyses are performed that all answer parts of this question

- 8.1.4.1 analyses the methods used for sentiment analysis and categorizes them, providing the distribution of methods used for sentiment analysis.
- 8.1.4.2 analyses the data sources that are used within the papers to see if there is any difference in data domain and technique used.
- 8.1.4.3 analyses what kind of preprocessing procedure is used because this can influence the performance of the sentiment analysis.
- 8.1.4.4 analyses how sentiment is extracted from text, this can happen on multiple levels and can be of influence on performance.
- 8.1.4.5 analyses how the ground truth is established on which the performance measures are based.
- 8.1.4.6 analyses the performance measures used to calculate the accuracy of the methods, these are used to compare their functioning.

These analyses are combined into one model that shows all articles and their analyzed parts. This model gives a complete overview of the analysis done and allows the identification of similarities and differences among applied methods.

8.1.4.1 Techniques for sentiment analysis

Different papers discuss the application and techniques used in sentiment analysis (Maynard & Funk, 2011; Ravi & Ravi, 2015), these papers provide an overview of sentiment classification. Ravi & Ravi (2015) argue that the techniques for sentiment analysis stem from Natural Language Processing (NLP), Information Retrieval (IR) and (un)structured Data Mining (DM). They argue that sentiment analysis can be used to perform different tasks, for these tasks there are different techniques available. These tasks are: subjectivity or sentiment classification, opinion spam detection, review usefulness detection, lexicon creation and aspect extraction. Ravi and Ravi (2015) introduce five sentiment analysis categories that relate to the tasks executed; machine learning approaches, lexicon based approached, hybrid approaches, ontology based and non-ontology based approaches. The most important tasks and categories and their relations of these tasks and categories are shown in figure 2. Lexicon creation and aspect extraction are referred to as lesser used tasks for sentiment analysis.

The earlier mentioned three main categories are also advocated by other researchers (Maynard & Funk, 2011). It thus seems viable to use the three main approaches for sentiment analysis as mentioned earlier for classifying the found literature. In this systematic literature review three approaches are adopted by the papers analyzed; machine learning approaches, lexicon based approaches and hybrid approaches. All these approaches determine the polarity of a text or word and compute a total sentiment score. In general lexicon approaches compute these scores based on a predefined dictionary of words, machine learning approaches compute these scores based on a model that is trained on labeled training data. The hybrid approach combines the previous two methods. There are no ontology and non-ontology approaches found in the article pool, these are thus excluded

from the analysis. This relates to the findings in the previous section that ontology based and non-ontology based approaches are less used as sentiment analysis tasks and are thus less likely to appear in the literature found. First the main differences between the three main approaches for sentiment analysis are discussed, definitions of the approaches are used to categorize the obtained literature.

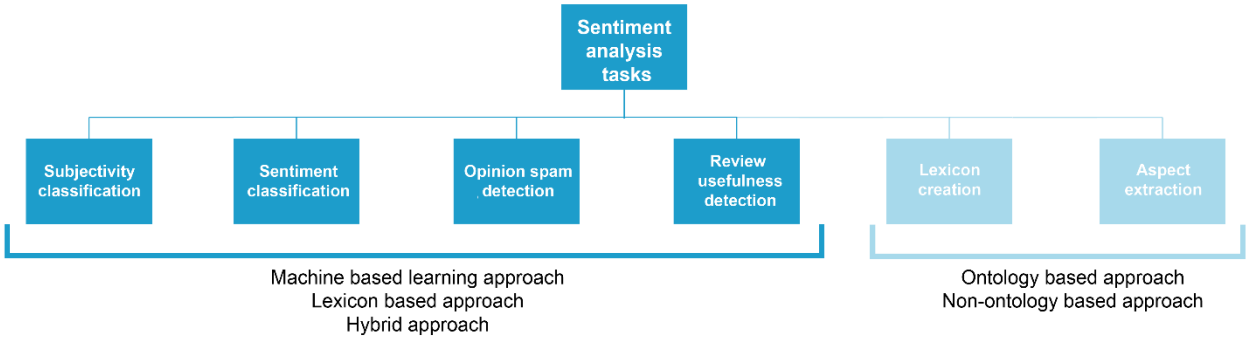


Figure 2: Sentiment analysis tasks and approaches (adapted from: Ravi & Ravi, 2015)

Machine learning based approaches

Machine Learning Approaches (ML) apply ML algorithms such as: Naive Bayes (NB), Support Vector Machines (SVM), Neural Networks (NN), Bayesian Network (BN) and Maximum Entropy (ME) (Medhat et al, 2014). These machine learning approaches use linguistic features to classify the sentiment of a text. ML approaches can be further divided into supervised and unsupervised learning processes. Supervised learning processes require labeled data for training of the algorithm applied and classify target data based on the training set (Hassan Khan, Qamar & Bashir, 2015). Unsupervised methods are used when there is no labeled training data available, the categories of objects are unknown. Unsupervised learning aims to assign these class labels (Khan, Qamar & Bashir, 2016).

Lexicon based approaches

Lexicon based approaches rely on a so called sentiment lexicon, which is basically a collection of sentiment terms. Lexicon based approaches can be divided into two sub-categories: dictionary based and corpus based. The dictionary approach depends on finding specific seed words (or their synonyms and antonyms) in the dictionary (Khan, Qamar & Bashir, 2016). The corpus based approach starts with a list of opinion words, it then finds other opinion words in a large corpus to help derive context specific opinion words. Based on these opinion words the polarity, or semantic orientation, of each word or phrase in the text is computed.

Hybrid approaches

Hybrid approaches combine ML and Lexicon approaches; the most common form of the hybrid approach is using a lexicon approach as the basis and machine learning is used to increase accuracy of the sentiment classification (Mudinas, Zhang & Levene, 2012; Pawar, Shrishrimal & Deshmukh, 2015).

The articles read and analyzed are categorized within these three main categories based on the method applied within the paper. This categorization is shown in figure 3. In general, all three classes are represented within the article pool but there is a clear favor for either a ML or hybrid approach. This finding is in line with multiple researchers that state that many of today's approaches are either use ML techniques or hybrid approaches (Mudinas, Zhang & Levene, 2012; Pawar, Shrishrimal & Deshmukh, 2015). Possible explanations for this are the following: pure lexicon based approaches

give poor results in real-life application due to the maintenance of sentiment dictionaries for different domains (Sharma & Dey, 2013). According to Mudinas et al. (2012), ML approaches achieve very high accuracy by treating sentiment detection as a "simple" classification problem which require no expensive dictionary. They also argue that hybrid approaches are favorable because the combination of ML and lexicons created more structured and readable results which are less sensitive to writing styles in text.

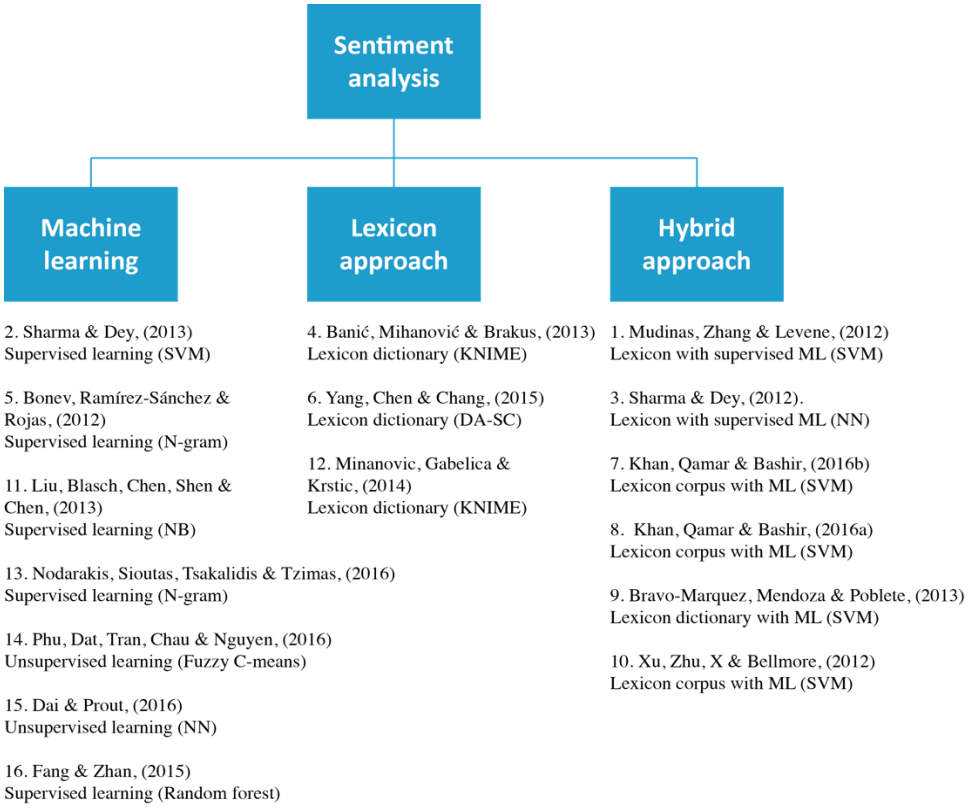


Figure 3: Categorization of papers

Note: Multiple papers (Fang & Zhan, 2015; Khan, Qamar & Bashir, 2016a; Khan, Qamar & Bashir, 2016b; Nodarakis, Sioutas, Tsakalidis & Tzimas, 2016; Sharma & Dey, 2012a; Sharma & Dey, 2013; Yang, Chen & Chang, 2015) use multiple methods to compare their methods performance to other methods. Only the method that is designed and executed in the paper is used to classify the paper into a category. An example is Sharma & Dey (2012a), they use a Neural Network method to classify sentiment from text, they compare their method to other methods such as lexicon approaches. This paper is classified within the machine learning category because their developed method concerns a neural network which falls under the machine learning category.

8.1.4.2 Data sources used for sentiment analysis

For sentiment analysis different data sources are used, to see if there is any difference in applied method for sentiment analysis based on the data used it is of importance to analyze the data sources. All papers use some kind of data but it differs along the platform that is used to extract the data from, this can for instance be Tweet data from Twitter or product data from Amazon. For different data domains there could be different sentiment classification techniques performed because of their functioning in relation to the data used (Singh, Paul & Kumar, 2014). Consider differences in techniques applied on structured data from ERP systems or unstructured data such as Twitter feed. Some papers (Bravo-Marquez, Mendoza & Poblete, 2013; Dai & Prout, 2016; Khan, Qamar & Bashir,

2016a; Khan, Qamar & Bashir, 2016b; Minanovic, Gabelica & Krstic, 2014; Mudinas, Zhang, & Levene, 2012; Sharma & Dey, 2013; Yang, Chen & Chang, 2015) use multiple datasets to build and test their model, therefore the count of total data domains is higher than the amount of papers in the short list. The papers analyzed apply their sentiment analysis to different domains, these are categorized into four categories; movie and video reviews, hotel reviews, product reviews and twitter feed.

Movie and video reviews, count: 11
7 papers use movie reviews to test the functioning of their sentiment analysis techniques, movie and video reviews are collected from popular websites such as IMDB and Cornell University. These datasets range in between: 2.000 – 400.000 entities.

Hotel reviews, count: 2
2 papers in total have applied their sentiment analysis techniques to hotel reviews. These databases range consist of 2.000 and 105.000 entities.

Product reviews, count: 15
A total of 5 papers use product reviews as their database. Product reviews concern reviews for products such as cars and computers, but also financial and health products. These databases range between: 7.000 and 1.230.000 entities

Twitter feed, count: 8
Twitter databases consist of Tweet extracted from Twitter using the Twitter API. These databases range between: 3500 and 2.2000.000 entities.

8.1.4.3 Preprocessing

Raw data that is gathered often needs some preprocessing steps before actual analysis can be performed (Haddi, Liu & Shi, 2013). Pre-processing is cleaning and preparing the textual data for the classification of the sentiment, which should increase the accuracy of the analysis. There are several pre-processing techniques available for sentiment analysis: tokenization, stop word removal, Parts Of Speech (POS) tagging, feature extraction/selection and stemming (Ravi & Ravi, 2015). All of these preprocessing techniques are used to reduce the extreme noise that is apparent in textual data and to remove irrelevant parts in the text. The papers in the literature shortlist are analyzed and 10 different preprocessing techniques are used; POS tagging, tokenization, stop word removal, stemming, lemmatization, feature selection, punctuation removal, URL and hashtag removal, word segmentation and a number of characters filter. Below a small explanation of the different techniques is given.

POS tagging, count 5:
POS tagging determines what part of speech a word is such as: nouns, verbs, etc. It creates a so-called "POS-tag"for this word that is stored. Note that words can have multiple POS-tags based on the context of the sentence that the word is in. Based on the POS tag, words that belong to a certain category can be deleted or kept as they contain either low or high sentiment value (Voutilainen, 2003).

Tokenization, count: 3
Tokenization is breaking up the textual document in parts of sentences or words and punctuation. Tokenization tries to explore the words in a sentence or sentences in a document and separates this from punctuation (Vijayarani, Ilamathi & Nithya, 2015).

Stopword removal, count: 7
There are many words in a document or text that appear frequently but are meaningless when it comes to sentiment value. They are just used to join words together, making it easy to read but they have no contribution to the content/context of the document. Common stop words are for example: and, are, this, that, etc. (Vijayarani, Ilamathi & Nithya, 2015).

Stemming, count: 4
Stemming conflates the forms of different words into its common representation as the stem of the word e.g. "running" is transformed to its stem "run". The problem with stemming is that it can reduce words with different stems into the same stem (over-stemming), or two words with the same stem are transformed into two different stems (under-stemming) (Vijayarani, Ilamathi & Nithya, 2015). Stemming is done by a heuristic process that removes the ends of the words and using that as a stem form of that word (Schütze, 2008).

Lemmatization, count: 2
Closely related to stemming is lemmatization, the difference resides in the fact that stemming operates on a single word without taking the context into account. Lemmatization also transforms words into a base form of the word, this process is based on a vocabulary and morphological analysis of words (Schütze, 2008).

Feature selection, count: 3
Feature selection is a process where the text corpus is removed of any unnecessary features, i.e. words that do not hold any sentiment, that are apparent in the corpus. This happens before the sentiment classifier is trained to allow for better model fit and better faster classification (Sharma & Dey, 2012).

Punctuation erasure, count: 1
Removing punctuation to reduce invaluable information as punctuation holds no sentiment value.

URL and hashtag removal, count: 2
Removing URL's and hashtags from the text to reduce invaluable information. Note that some researchers do include hashtag as a source of information for sentiment classification (Minanovic, Gabelica & Krstic, 2014; Nodarakis, Sioutas, Tsakalidis & Tzimas, 2016).

Word segmentation, count: 1
Segmenting words in a string that might have been misspelled into different usable words that appear in a dictionary.

Number of characters filter, count: 1
Selecting and removing words based on the number of characters.

Based on the pre-processing methods mentioned within the articles several groups of pre-processing can be distinguished. These main groups of pre-processing techniques provide a clearer view of which pre-processing steps are taken for each article than analyzing the specific pre-processing methods. The following main groups are identified within the analyzed literature:

Group 1: Preparing data before cleaning
(Tokenization, POS Tagging)
This group of pre-processing methods is used to categorize textual data into categories that are used for selecting which data to keep or to remove in the data cleaning process.

Group 2: correcting words
(Word segmentation)
Checks the texts for words that are uncommon and tries to determine the correct word.

Group 3: Simplifying words
(Stemming and Lemmatization)
Uses pre-processing techniques to derive base forms of words to simplify the sentiment classification task.

Group 4: Data cleaning
(Stop word removal, feature selection, punctuation erasure, URL and hashtag removal, number of character filter)
This group of pre-processing methods is used to clean the data based to remove any invaluable data that can slow down the classification algorithm and reduce its accuracy.

It is observed that some articles mention no pre-processing technique at all (Banić, Mihanović & Brakus, 2013; Liu, Blasch, Chen, Shen & Chen, 2013). There are many articles that do not consider preparing the data before cleaning by means of tokenization or using POS tagging, these articles immediately start cleaning their data. There is big difference observed in the papers analyzed about the way the process of the sentiment analysis is conducted. Some papers clearly state a workflow in which they argue their pre-processing approach, others consider these steps less important and spend little attention to stating their workflow.

8.1.4.4 Sentiment extraction

There are three main levels on which sentiment analysis can be executed; document-level, sentence-level and aspect-level (Medhat, Hassan & Korashy, 2014). The three main levels of sentiment analysis are thoroughly described by Feldman (2013) in his article: techniques and applications for sentiment analysis, that provides an overview of the hot topic that is sentiment analysis is. The following descriptions are used for determining the sentiment extraction method of the papers read, these descriptions are based on the above-mentioned article of Feldman.

Document-level sentiment analysis

Document-level sentiment analysis tries to extract the sentiment from the whole document applied, a restriction here is that the document concerns only one topic. Within document level sentiment analysis there are two main approaches for executing the analysis: supervised and unsupervised learning (see 4.1.1). Within the supervised approach training data is labeled as positive or negative, a model then learns the pattern and classifies the new input based on the previous learned pattern. Unsupervised learning determines the semantic orientation of a document by selecting phrases and calculating their semantic orientation, an average then determines if the document is classified into a positive or negative class according to a certain threshold.

4.1.1 Sentence-level sentiment analysis

Sentence-level sentiment analysis extracts the sentiment from all sentences within the document, an assumption here is that all sentences contain a single opinion. With this type of sentiment analysis a more fine-grained view of different opinions within a document can be extracted. First the polarity of the sentence is determined i.e. if the sentence is subjective or objective, only subjective sentences will be analyzed. Thereafter these sentences are classified as positive or negative. Most of these approaches are based on unsupervised learning (Feldman, 2013).

4.1.2 Aspect-based sentiment analysis

Document and sentence level sentiment analysis work best when the analyzed content concerns one subject or entity. Aspect-based sentiment analysis tries to recognize all sentiment expressions in a text and the aspects to which they refer. A common approach is to use a corpus and to extract all noun phrases of a text, the noun phrases that have a frequency above some predetermined threshold are kept for analysis (Hu & Liu, 2004).

Sentiment analysis can be executed on multiple levels and can be of influence on performance and the choice of method. (document level, sentence level and aspect level). The papers are analyzed to extract their sentiment level approach. The distinction between the three main levels of sentiment analysis by Medhat et al. (2014) seems valid since all three methods are used by the papers in this literature review.

Document-level sentiment analysis, count: 3
In total six papers use a document-level approach to sentiment analysis, there is no consistency in data type used for a document-level approach.

Sentence-level sentiment analysis, count: 6
Six papers adopt a sentence-level sentiment approach, the main data type used for this approach is twitter data, but there is also product review data used.

Aspect-based sentiment analysis, count: 7
Aspect-based analysis is executed seven times, the main data type used for this approach is movie and hotel reviews, also some product review data is used.

8.1.4.5 Ground truth

The ground truth in sentiment analysis is used to determine the performance of the models applied. The ground truth states the polarity of a document, sentence or aspect based on an assigned label. Without these labels there is no way of measuring the performance of the adopted models. Within the papers analyzed there are a few methods adopted to establish this ground truth.

Star rating, count: 4
Four papers use a scale rating (1 to 5) based on stars assigned to product or movie reviews. The main method adopted is to classify ratings higher than 3 as positive reviews, lower than 3 as negative and ratings equal to 3 as neutral. All of these ratings are given by the person that made the review.

Corpus labels, count: 6
A total of six papers use a pre-labeled corpus to determine their ground-truth, the data is thus pre-labeled with the polarity of the Tweet or product review. These papers assume that the corpus they have used is valid and that the method applied by the makers to determine the polarity of the content of the corpus is also valid.

Manually labeled, count: 2
A mere two papers manually adjust their data with the polarity labels. This can be done by independent judges that do not participate in the research itself. This is tedious and time consuming work, and is thus not widely adopted.

Not stated, count: 4
There are four papers that do not specifically state how their ground truth is established, some papers use a corpus others do not. It can be assumed that the corpuses are pre-labeled but this cannot be validated.

8.1.4.6 Performance measures

To assess the performance of the sentiment analysis methods applied in the papers, most papers use some sort of statistical measure to assess its accuracy. These measures are used to compare the effectiveness of the sentiment analysis method and provide an overview of the best and worst methods in the field. The measures used in the papers are all based on the accuracy of the model, they classify their predictions into four categories: true positives (TP) are correctly predicted positive sentiments, true negatives (TN) are correctly predicted negative sentiments, false positives (FP) are positive predicted sentiments but are negative sentiments in reality and false negatives (FN) are negative predicted sentiments but are positive sentiments in reality. The papers analyzed use the overall accuracy, the F1 score and sometimes they provide a ROC curve to further demonstrate the performance of their method.

$$overall\ accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FN}$$

$$recall = \frac{TP}{TP + FP}$$

Within the papers read there is quite some difference in methods applied to test the performance of the sentiment analysis. Most are based on accuracy or F1 scores as mentioned above, but the number of test sets used to give a performance measure differs. Some papers test the performance of their sentiment analysis method on multiple test sets (Bravo-Marquez, Mendoza & Poblete, 2013; Dai & Prout, 2016; Khan, Qamar & Bashir, 2016a; Khan, Qamar & Bashir, 2016b; Mudinas, Zhang, & Levene, 2012; Sharma & Dey, 2013; Yang, Chen & Chang, 2015), when this is the case the outcomes of the performance measures for the different test sets are averaged.

Also some papers (Fang & Zhan, 2015; Khan, Qamar & Bashir, 2016a; Khan, Qamar & Bashir, 2016b; Nodarakis, Sioutas, Tsakalidis & Tzimas, 2016; Sharma & Dey, 2012a; Sharma & Dey, 2013; Yang, Chen & Chang, 2015) compare the accuracy of their method in comparison to other used methods, only the performance measure of the subjective method applied in the paper is given. For instance, Sharma & Dey (2013) compare their boosted SVM sentiment analysis with other SVM sentiment analysis methods. In this case only the performance measure for the boosted SVM is given.

Three papers state no performance measure, Banić, Mihanović & Brakus (2013) state that their results are not fully reliable, Xu, Zhu & Bellmore (2012) provide only the cross validation error on their training set and Minanovic, Gabelica & Krstic, (2014) provide no measure for their methods performance.

8.1.4.7 Cluster Analysis

The previous chapters explained the form of the table. Table 4 on the next page shows which sentiment techniques are used in which articles and all other previous analyzed parameters. The articles are ordered differently to better visualize the clusters that became apparent from the analysis. Data used, performance measures and sentiment approaches such as machine learning or lexicon based approaches are used as abbreviations to reduce the width of the table, which increases readability. There is an "X" marked within one or two of the sentiment approaches if a paper uses that specific sentiment approach, two "X" are placed when a paper uses a hybrid approach. These "X" then correspond to the hybrid combination of sentiment approaches used.

Authors	Data used	Pre-processing	performance measures	Ground truth	performance	Extraction	Machine learning						Lexicon	
							SVM	N-gram	NB	FC	NN	RF	DT	CP
1. Mudinas, Zhang & Levene, (2012)	P & M/V	G1	A	S	82%*	Aspect	X						X	
2. Sharma & Dey, (2013)	H & M/V	G1. G3, G4	A	S	93%*	Aspect	X	Cluster 1						
3. Sharma & Dey, (2012)	M/V	G1, G3, G4	A	C	95%*	Aspect	X						X	
7. Khan, Qamar & Bashir, (2016b)	P & M/V	G1, G4	A & F1	C	86%* / 81%**	Aspect	X							X
8. Khan, Qamar & Bashir, (2016a)	P & M/V	G1, G3, G4	A & F1	N/A	82%* / 81%**	Aspect	X		Cluster 2					X
4. Banić, Mihanović & Brakus, (2013)	H	N/A	N/A	N/A	-	Aspect							X	
12. Minanovic, Gabelica & Krstic, (2014)	P & T	G4	N/A	N/A	-	Aspect	cluster 3						X	
10. Xu, Zhu, X & Bellmore, (2012)	T	G3, G4	N/A	C	-	Sentence	X							X
9. Bravo-Marquez, Mendoza & Poblete, (2013)	T	G4	A & F1	C	82%* / 82%**	Sentence	X		Cluster 4				X	
13. Nodarakis, Sioutas, Tsakalidis & Tzimas, (2016)	T	G4	F1	ML	77%**	Sentence		X						
15. Dai & Prout, (2016)	T	G4	A	N/A	78%*	Sentence					X			
14. Phu, Dat, Tran, Chau & Nguyen, (2016)	M/V	G2, G4	A	C	60%*	Document				X				
5. Bonev, Ramírez-Sánchez & Rojas, (2012)	P	G3	A	ML	82%*	Sentence		X						
6. Yang, Chen & Chang, (2015)	P	G1, G4	A & F1	S	81%* / 85%**	Document							X	
11. Liu, Blasch, Chen, Shen & Chen, (2013)	M/V	N/A	A	S	82%*	Document			X					
16. Fang & Zhan, (2015)	P	G1, G4	F1	C	85%**	Sentence						X		

Data type used:	Legend:
Movie/Video reviews	M/V
Hotel reviews	H
Product reviews	P
Twitter feed	T

Classification methode:	Legend:
Accuracy	A
F1-scores	F1
Not applicable	N/A

Ground truth	Legend:
Based on star rating	S
Labeled corpus	C
manually labeled	ML
Not stated	N/A

Performance	Legend:
Are all rounded digits	* = Accuracy
	** = F1

Algorithm used	Legend:
Supp. Vect. machine	SVM
N-gram	N-gram
Naive baynes	NB
Fuzzy c-means	FC
Neural networks	NN
Random forest	RF
Dictionary based	DT

Pre-processing	Legend:
Group 1 - Preparing data before cleaning	G1
Group 2 - Correcting words	G2
Group 3 - Simplifying words	G3
Group 4 - Data cleaning	G4
N/A	Not explicitly mentioned

Table 4: Analysis cluster model

Clusters in the article pool become apparent when combining the data from the previous chapters, 4 different clusters are identified. The largest cluster contains 5 articles, one cluster contains 4 articles and two clusters contain 3 articles. The largest cluster contains all articles that use an aspect-based sentiment approach and all of these articles use movie and video review data as input. They all provide at least the accuracy performance measure and achieve overall high-performance. This cluster also mainly uses the following pre-processing steps: preparing the data before cleaning, simplifying words and data cleaning. The second cluster contains all Not Available (NA) performance measures, this cluster contains 3 articles. What is interesting to notice is that all of these articles use at least a lexicon based approach, which might explain the fact that they do not provide performance measures. Two of the three articles makes use of KNIME for their execution of sentiment analysis. There is a variety of input data used for the execution of these analysis and also the way to determine sentiment by extraction method. The third cluster contains all articles that use Twitter data (T) as their input, all these articles use a sentence-level sentiment approach to determine the opinion within a Tweet. The overall accuracy of these methods varies from Not Available (NA) to high accuracy scores of 82%. Overall the accuracy is lower compared to other clusters. These articles mainly use data cleaning as their only pre-processing step. The fourth cluster consists of 3 papers that have the lowest performance of the whole article pool, these performances are all below an 80% threshold. Interesting to note is that these articles all use a purely machine-learning based sentiment algorithm.

8.1.4.8 Sub-conclusion

From the above-stated analysis chapter we can derive two main components, the first is the structuring towards a classification tree model where three major approaches for sentiment analysis are defined. What is noticed is that there are other methods for sentiment analysis such as deep learning which are not used within the article pool found. The articles analyzed use mainly supervised learning techniques that require previous labeled training data for classifying. The second is the analysis of several dimensions which are converted in an analysis table (table 4) that show relations between these analyzed dimensions. This clearly shows that there are clusters which relate to the aspect to which a sentiment process is applied and its output. Overall there are 4 major clusters identified.

8.1.5 Discussion and conclusions

The goal of this systematic literature review was to explore the current knowledge base of techniques available for sentiment analysis. This review uses articles from the last 5 years due to quick developments in the sentiment analysis field and the vast amount of literature on this topic that is available. The literature review analyses common components from the articles that made the cut to the literature shortlist. There is a wide variety of methods used and some of these provide very good accuracies and some none at all. Clusters in the articles are defined in the analysis table and this shows that there are some similarities between the read papers. The highest accuracies are achieved by aspect-based sentiment analysis and these all include movie and video review data, these papers mainly use three groups of data pre-processing that partly explains the higher accuracies. From all the articles that could not provide a performance, two third used KNIME as sentiment analysis tool, also the approaches used are all (partly) lexicon based. For every article that uses Twitter data is input there is a sentence-level sentiment analysis approach used. Although this approach is adopted it does mostly not achieve high accuracy, this might also be due to the low usage of pre-processing steps. The last cluster shows the lowest achieved accuracies within the article pool and all these sentiment methods are based on pure machine-learning.

A discussion point concerns the use of the term "big data" in the search strategy. As mentioned big data is an ambiguous term as there are many definitions and it is not exactly clear what is meant by it. This also shows in the articles read, some interpret "big data" as so much data that it cannot be analyzed by a standalone computer, others define it as data which is highly available and can stem from multiple sources. The majority of the articles uses the last definition and uses input data that is not extremely large, there are two papers that use so much data that they use an online cloud computing approach using Hadoop. This insight concludes that for graduation there needs to be more research towards what "big data" is and which papers remain relevant for graduation.

Overall this systematic literature review provides a theoretical basis on which the master thesis will be build. It became clear that there are multiple methods available for sentiment analysis and that certain methods are used for specific data. When starting graduating and determining and designing the method to be used, this systematic literature review serves as a good starting point.

8.1.6 References

Banafa, A. (2015). Small Data vs. Big Data : Back to the basics. Retrieved September 07, 2017, from https://datafloq.com/read/small-data-vs-big-data-back-to-the-basic/706

Banić, L., Mihanović, A., & Brakus, M. (2013). Using big data and sentiment analysis in product evaluation. In *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on* (pp. 1149-1154). IEEE.

Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2013). Combining strengths, emotions and polarities for boosting Twitter sentiment analysis. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining* (p. 2). ACM.

Collins, C. M., Steg, L., & Koning, M. A. (2007). Customers' values, beliefs on sustainable corporate performance, and buying behavior. *Psychology & Marketing*, *24*(6), 555-577.

Dai, X., & Prout, B. (2016). Unlock big data emotions: Weighted word embeddings for sentiment classification. In *Big Data (Big Data), 2016 IEEE International Conference on* (pp. 3833-3838). IEEE.

Davenport, T. H. (2006). Competing on analytics. *Harvard business review*, *84*(1), 98.

Demchenko, Y., Grosso, P., de Laat, C., & Membrey, P. (2013). Addressing big data issues in Scientific Data Infrastructure. In 2013 International Conference on Collaboration Technologies and Systems (CTS) (pp. 48–55).

Fang, X., & Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, *2*(1), 5.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, *56*(4), 82-89.

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University press.

Furlow, N. E. (2010). Greenwashing in the New Millennium. *Journal of Applied Business and Economics*, *10*(6), 22–25.

Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science, 17*, 26-32.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177). ACM.

Khan, F., Qamar, U., & Bashir, S. (2015). Building normalized SentiMI to enhance semi-supervised sentiment analysis. *Journal of Intelligent & Fuzzy Systems*, *29*(5), 1805-1816.

IBM (2013). *What will we make of this moment?: 2013 IBM annual report*. Armonk, NY: International Business Machines Corporation.

Khan, F. H., Qamar, U., & Bashir, S. (2016a). A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. *Knowledge and Information Systems*.

Khan, F. H., Qamar, U., & Bashir, S. (2016b). Multi-objective model selection (MOMS)-based semi-supervised framework for sentiment analysis. *Cognitive Computation*, *8*(4), 614-628.

Kitchenham, B. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

Liu, B., Blasch, E., Chen, Y., Shen, D., & Chen, G. (2013). Scalable sentiment classification for Big Data analysis using Naive Bayes Classifier. *2013 IEEE International Conference on Big Data*.

Madden, S. (2012). From databases to big data. *IEEE Internet Computing*, *16*(3), 4-6.

Maynard, D., & Funk, A. (2011). Automatic detection of political opinions in Tweets. In *Extended Semantic Web Conference* (pp. 88-99). Springer Berlin Heidelberg.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093-1113.

Minanovic, A., Gabelica, H., & Krstic, Z. (2014). Big data and sentiment analysis using KNIME: Online reviews vs. social media. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on* (pp. 1464-1468). IEEE.

Mudinas, A., Zhang, D., & Levene, M. (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining* (p. 5). ACM.

Nodarakis, N., Sioutas, S., Tsakalidis, A., & Tzimas, G. (2016). MR-SAT: a MapReduce algorithm for big data sentiment analysis on Twitter. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies* (Vol. 1, pp. 140-147).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1-135.

Pawar, K., Shrishrimal, P., & Deshmukh, R. (2015). Twitter sentiment analysis: A review. *International Journal of Scientific & Engineering Research*, *6*(4), 9.

Press, G. (2014). A Very Short History Of Data Science. Retrieved February 14, 2017, from http://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#3371f95669fd

Provost, F., & Fawcett, T. (2013). *Data science for business*. Sebastopol, CA: O'Reilly.

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, *89*, 14-46.

Schütze, H. (2008). Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*.

Schwarz, J., Beloff, B., & Beaver, E. (2002). Use sustainability metrics to guide decision-making. *Chemical Engineering Progress*, *98*(7), 58-63.

Sharma, A., & Dey, S. (2012a). An artificial neural network based approach for sentiment analysis of opinionated text. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium* (pp. 37-42). ACM.

Sharma, A., & Dey, S. (2012b). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM research in applied computation symposium* (pp. 1-7). ACM.

Sharma, A., & Dey, S. (2013). A boosted SVM based sentiment analysis approach for online opinionated text. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems* (pp. 28-34). ACM.

Singh, S. K., Paul, S., & Kumar, D. (2014). Sentiment Analysis Approaches on Different Data set Domain: Survey. *International Journal of Database Theory and Application*, *7*(5), 39-50.

Sundaresan, N. (2016). The History of Data Science. Retrieved February 14, 2017, from http://www.huffingtonpost.com/quora/the-history-of-data-scien_b_10116442.html

Talib, R., Hanif, M. K., Ayesha, S., & Fatima, F. (2016). Text Mining : Techniques , Applications and Issues, *7*(11), 414–418.

Vijayarani, S., Ilamathi, J., & Nithya, M. (2015). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Voutilainen, A. (2003). Part-of-speech tagging. *The Oxford handbook of computational linguistics*, 219-232.

Walker, B. (2015). 2.5 Quintillion Bytes of Data Created Daily. Retrieved February 13, 2017, from http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

Xu, J. M., Zhu, X., & Bellmore, A. (2012). Fast learning for sentiment analysis on bullying. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining* (p. 10). ACM.

Yang, C. S., Chen, C. H., & Chang, P. C. (2015). Harnessing consumer reviews for marketing intelligence: a domain-adapted sentiment classification approach. *Information Systems and e-Business Management*, *13*(3), 403-419.

8.1.7 Appendix

8.1.7.1 Appendix 1: Literature long list

Springer

NO:	Authors	title	Year	Published in
1	Erik Tromp, Mykola Pechenizkiy, Mohamed Medhat Gaber	Expressive modeling for trusted big data analytics: techniques and applications in sentiment analysis	2017	Big Data Analytics
2	Umberto Grandi, Andrea Loreggia, Francesca RossiVijay Saraswat	A Borda count for collective sentiment analysis	2016	Annals of Mathematics and Artificial Intelligence
3	Jaspreet Singh, Gurvinder Singh, Rajinder Singh	A review of sentiment analysis techniques for opinionated web text	2016	CSI transactions on ICT
4	Gaowei Zhang, Lingyu Xu, Yunlan Xue	Model and forecast stock market behavior integrating investor sentiment analysis and transaction data	2017	Cluster Computing
5	Xing Fang, Justin Zhan	Sentiment analysis using product review data	2015	Journal of Big Data
6	Vo Ngoc Phu, Nguyen Duy Dat, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, Tuan A. Nguyen	Fuzzy C-means for english sentiment classification in a distributed system	2016	Applied Intelligence
7	Farhan Hassan Khan, Usman Qamar, Saba Bashir	A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet	2016	Knowledge and Information Systems
8	Chin-Sheng Yang, Cheng-Hsiung Chen, Pei-Chann Chang	Harnessing consumer reviews for marketing intelligence: a domain-adapted sentiment classification approach	2015	Information Systems and e-Business Management
9	Farhan Hassan Khan, Usman Qamar, Saba Bashir	Multi-Objective Model Selection (MOMS)-based Semi-Supervised Framework for Sentiment Analysis	2016	Cognitive Computation
10	Mehmet Kaya, Shannon Conley	Comparison of sentiment lexicon development techniques for event prediction	2016	Social Network Analysis and Mining
11	Xinzhi Wang, Hui Zhang, Shengcheng Yuan, Jiayue Wang, Yang Zhou	Sentiment processing of social media information from both wireless and wired network	2016	Journal on Wireless Communications and Networking
12	Li Yang, Xinyu Geng, Haode Liao	A web sentiment analysis method on fuzzy clustering for mobile social media users	2016	Journal on Wireless Communications and Networking
13	Yanyan Zhao, Bing Qin, Ting Liu, Duyu Tang	Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog	2016	Multimedia Tools and Applications
14	Fei Jiang, Yi-Qun Liu, Huan-Bo Luan, Jia-Shen Sun, Xuan Zhu, Min Zhang, Shao-Ping Ma	Microblog Sentiment Analysis with Emoticon Space Model	2015	Journal of Computer Science and Technology
15	Nan Cao, Lu Lu, Yu-Ru Lin, Fei Wang, Zhen Wen	SocialHelix: visual analysis of sentiment divergence in social media	2015	Journal of Visualization
16	Martin Haselmayer, Marcelo Jenny	Sentiment analysis of political communication: combining a dictionary approach with crowdcoding	2016	Quality & Quantity
17	Fu-Chen Yang, Anthony J.T. Lee, Sz-Chen Kuo	Mining Health Social Media with Sentiment Analysis	2016	Journal of Medical Systems
18	Hui Yuan, Wei Xu, Qian Li, Raymond Lau	Topic sentiment mining for sales performance prediction in e-commerce	2017	Annals of Operations Research
19	Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, Alexander Serebrenik	On negative results when using sentiment analysis tools for software engineering research	2017	Empirical Software Engineering
20	Guangyou Zhou, Zhiyuan Zhu, Tingting He, Xiaohua Tony Hu	Cross-lingual sentiment classification with stacked autoencoders	2016	Knowledge and Information Systems
21	Dong Wang, Jiexun Li, Kaiquan Xu, Yizhen Wu	Sentiment community detection: exploring sentiments and relationships in social networks	2017	Electronic Commerce Research

ACM digital library

NO:	Authors	title	Year	Published in
1	Kim Yoosin, Kim Taeyun, Park Miri, Kang Suna, Choi Yiseul, Choi Sanghyun	Sentiment Analysis of Consumer Opinion in Blogs: A Case Study in Ramen Market	2015	BigDAS '15: Proceedings of the 2015 International Conference on Big Data Applications and Services
2	Boyan Bonev, Gema Ramírez-Sánchez, Sergio Ortiz Rojas	Opinum: statistical sentiment analysis for opinion classification	2012	WASSA '12: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis
3	Felipe Bravo-Marquez, Marcelo Mendoza, Barbara Poblete	Combining strengths, emotions and polarities for boosting Twitter sentiment analysis	2013	WISDOM '13: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining
4	Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore	Fast learning for sentiment analysis on bullying	2012	WISDOM '12: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining
5	Jianbo Yuan, Sean Mcdonough, Quanzeng You, Jiebo Luo	Sentribute: image sentiment analysis from a mid-level perspective	2013	WISDOM '13: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining
6	Andrius Mudinas, Dell Zhang, Mark Levene	Combining lexicon and learning based approaches for concept-level sentiment analysis	2012	WISDOM '12: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining
7	Sarvesh Ranade, Jayant Gupta, Vasudeva Varma, Radhika Mamidi	Online debate summarization using topic directed sentiment analysis	2013	WISDOM '13: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining
8	Gábor Berend, Veronika Vincze	How to evaluate opinionated keyphrase extraction?	2012	WASSA '12: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis
9	Anuj Sharma, Shubhamoy Dey	A boosted SVM based sentiment analysis approach for online opinionated text	2013	RACS '13: Proceedings of the 2013 Research in Adaptive and Convergent Systems
10	Anuj Sharma, Shubhamoy Dey	An artificial neural network based approach for sentiment analysis of opinionated text	2012	RACS '12: Proceedings of the 2012 ACM Research in Applied Computation Symposium
11	Darlan Santana Farias, Ivone Penque Matsuno, Ricardo Marcondes Marcacini, Solange Oliveira Rezende	Opinion-meter: A Framework for Aspect-Based Sentiment Analysis	2016	Webmedia '16: Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web
12	Jyoti, Seema Rao	A Survey on Sentiment Analysis and Opinion Mining	2016	AICTC '16: Proceedings of the International Conference on Advances in Information Communication Technology & Computing
13	Florencia Gamboa, Marcos A. Malamud, Ana G. Maguitman, Carlos I. Chesñevar	Citymis OpTree: Intelligent Citizen Management Using Sentiment Analysis and Opinion Trees	2016	ICEGOV '15-16: Proceedings of the 9th International Conference oin Theory and Practice of Electronic Governance
14	Amir Zadeh	Micro-opinion Sentiment Intensity Analysis and Summarization in Online Videos	2015	ICMI '15: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction
15	Lu Lin, Jianxin Li, Richong Zhang, Weiren Yu, Chenggen Sun	Opinion Mining and Sentiment Analysis in Social Networks: A Retweeting Structure-Aware Approach	2014	UCC '14: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing
16	Mondher Bouazizi, Tomoaki Ohtsuki	Opinion Mining in Twitter How to Make Use of Sarcasm to Enhance Sentiment Analysis	2015	ASONAM '15: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining
17	Jiangjiao Duan, Jianping Zeng, Banghui Luo	Identification of Opinion Leaders Based on User Clustering and Sentiment Analysis	2014	WI-IAT '14: Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)

Web of Knowledge

NO:	Authors	title	Year	Published in
1	Fang Youjia, Chen Xin, Song Zheng, Wang, Tianzi, Cao Yang	Modelling Propagation of Public Opinions on Microblogging Big Data Using Sentiment Analysis and Compartmental Models	2017	International Journal On Semantic Web and Information systems Volume:13 Issue: 1
2	Nodarakis Nikolaos, Sioutas, Spyros, Tsakalidis Athanasios, Tzimas Giannis	MR-SAT: A MapReduce Algorithm for Big Data Sentiment Analysis on Twitter	2016	Proceedings Of The 12th International Conference On Web Information Systems And Technologies
3	Povoda, Lukas; Burget, Radim; Dutta, Malay Kishore	Sentiment Analysis Based on Support Vector Machine and Big Data	2016	39th International Conference on Telecommunications And Signal Processing (TSP)
4	Tan Yi-Fei, Lam Hai-Shuan, Azlan Asyraf, Soo Wooi-King	Sentiment Analysis for Telco Popularity on Twitter Big Data Using a Novel Malaysian Dictionary	2016	Advances In Digital Technologies
5	Bohlouli Mahdi, Dalter Jens, Dornhoefer Mareike, Zenkert Johannes, Fathi Madjid	Knowledge discovery from social media using big data-provided sentiment analysis (SoMABiT)	2015	Journal Of Information Science Volume: 41 Issue: 6
6	Tedeschi A, Benedetto F	A Cloud-based Big Data Sentiment Analysis Application for Enterprises' Brand Monitoring in Social Media Streams	2015	1st International Forum On Research And Technologies For Society And Industry (RTSI 2015)
7	Ramanujam R Suresh, Nancyamala R, Nivedha J, Kokila J	SENTIMENT ANALYSIS USING BIG DATA	2015	International Conference on Computation Of Power, Energy, Information and Communication (ICCPEIC)
8	Guellil Imene, Boukhalfa Kamel	Social Big Data Mining : A Survey Focused on Opinion Mining and Sentiments Analysis	2015	12th IEEE International Conference on Programming and Systems (ISPS)
9	Skuza Michal, Romanowski Andrzej	Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction	2015	Proceedings Of The 2015 Federated Conference On Computer Science And Information Systems
10	Rahnama Amir Hossein Akhavan	Distributed Real-Time Sentiment Analysis for Big Data Social Streams	2014	International Conference On Control, Decision And Information Technologies (CODIT)
11	Mihanovic Ana, Gabelica Hrvoje, Krstic Zivko	Big Data and Sentiment Analysis using KNIME: Online Reviews vs. Social Media	2014	37th Internatiuonal Convention On Information And Communication Technology Electroninc s And Microelectroninc s (MIPRO)
12	Banic Lada, Mihanovic Ana, Brakus Marko	Using Big Data and Sentiment Analysis in Product Evaluation	2013	36th Internatiuonal Convention On Information And Communication Technology Electroninc s And Microelectroninc s (MIPRO)
13	Liu Bingwei, Blasch Erik, Chen Yu, Shen Dan, Chen Genshe	Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier	2013	IEEE International Conference On Big Data

IEEE-IET Electronic Library

NO:	Authors	title	Year	Published in
1	Kaoutar Ben Ahmed, Mohammed Bouhorma, Mohamed Ben Ahmed, Atanas Radenski	Visual sentiment prediction with transfer learning and big data analytics for smart cities	2016	IEEE International Colloquium on Information Science and Technology (CIST)
2	Xiangfeng Dai, Bob Prout	Unlock big data emotions: Weighted word embeddings for sentiment classification	2016	IEEE International Conference on Big Data

8.1.7.2 Appendix 2: Literature short list

Short list

1	Andrius Mudinas, Dell Zhang, Mark Levene	Combining lexicon and learning based approaches for concept-level sentiment analysis	2012	WISDOM '12: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining
2	Anuj Sharma, Shubhamoy Dey	A boosted SVM based sentiment analysis approach for online opinionated text	2013	RACS '13: Proceedings of the 2013 Research in Adaptive and Convergent Systems
3	Anuj Sharma, Shubhamoy Dey	An artificial neural network based approach for sentiment analysis of opinionated text	2012	RACS '12: Proceedings of the 2012 ACM Research in Applied Computation Symposium
4	Banic Lada, Mihanovic Ana, Brakus Marko	Using Big Data and Sentiment Analysis in Product Evaluation	2013	36th Internatiuonal Convention On Information And Communication Technology Electroninc s And Microelectroninc s (MIPRO)
5	Boyan Bonev, Gema Ramírez-Sánchez, Sergio Ortiz Rojas	Opinum: statistical sentiment analysis for opinion classification	2012	WASSA '12: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis
6	Chin-Sheng Yang, Cheng-Hsiung Chen, Pei-Chann Chang	Harnessing consumer reviews for marketing intelligence: a domain-adapted sentiment classification approach	2015	Information Systems and e-Business Management
7	Farhan Hassan Khan, Usman Qamar, Saba Bashir	Multi-Objective Model Selection (MOMS)-based Semi-Supervised Framework for Sentiment Analysis	2016	Cognitive Computation
8	Farhan Hassan Khan, Usman Qamar, Saba Bashir	A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet	2016	Knowledge and Information Systems
9	Felipe Bravo-Marquez, Marcelo Mendoza, Barbara Poblete	Combining strengths, emotions and polarities for boosting Twitter sentiment analysis	2013	WISDOM '13: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining
10	Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore	Fast learning for sentiment analysis on bullying	2012	WISDOM '12: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining
11	Liu Bingwei, Blasch Erik, Chen Yu, Shen Dan, Chen Genshe	Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier	2013	IEEE International Conference On Big Data
12	Mihanovic Ana, Gabelica Hrvoje, Krstic Zivko	Big Data and Sentiment Analysis using KNIME: Online Reviews vs. Social Media	2014	37th Internatiuonal Convention On Information And Communication Technology Electroninc s And Microelectroninc s (MIPRO)
13	Nodarakis Nikolaos, Sioutas, Spyros, Tsakalidis Athanasios, Tzimas Giannis	MR-SAT: A MapReduce Algorithm for Big Data Sentiment Analysis on Twitter	2016	Proceedings Of The 12th International Conference On Web Information Systems And Technologies
14	Vo Ngoc Phu, Nguyen Duy Dat, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, Tuan A. Nguyen	Fuzzy C-means for english sentiment classification in a distributed system	2016	Applied Intelligence
15	Xiangfeng Dai, Bob Prout	Unlock big data emotions: Weighted word embeddings for sentiment classification	2016	IEEE International Conference on Big Data
16	Xing Fang, Justin Zhan	Sentiment analysis using product review data	2015	Journal of Big Data

8.2 TV commercials aired

8.2.1 Voices of the gulf: business owner's impact

Voices From The Gulf: Louisiana Restaurant Owners (Series) – (24 October, 2010)
https://www.youtube.com/watch?v=oNN5xZgxq0k

Voices from the Gulf: Mississippi Fisherman – (26 October, 2010)
https://www.youtube.com/watch?v=_TN8_TN1sPU

Voices from the Gulf: Florida business owners – (10 January, 2011)
https://www.youtube.com/watch?v=JL6ooGCMISs

Voices from the Gulf: Alabama beaches business owners – (17 march, 2011)
https://www.youtube.com/watch?v=25_LICFMgMI

Voices From The Gulf: Louisiana Restaurant Owners – (17 march, 2011)
https://www.youtube.com/watch?v=vbqjOiCJEGM

8.2.2 Voices of the gulf: tourism

Voices From The Gulf: Best place – (23 August, 2011)
https://www.youtube.com/watch?v=onGNrCc7RQA

Voices From The Gulf: Great vacation – (23 August, 2011)
https://www.youtube.com/watch?v=jep5B9YBV-0

the gulf makes a perfect winter getaway – (21 November, 2011)
https://www.youtube.com/watch?v=tO2i_mMZfz4

Now is the perfect time to visit the gulf – (7 December, 2012)
https://www.youtube.com/watch?v=0FidIIQO6ZY

8.2.3 BP's commitment

BP gulf coast update – (20 December, 2011)
https://www.youtube.com/watch?v=hoOfIR4Vk1o

BP's ongoing commitment to the gulf - (27 April, 2012)
https://www.youtube.com/watch?v=BV13FugHAos

BP's ongoing commitment to America - (22 August, 2012)
https://www.youtube.com/watch?v=V6m-VJy8Toc

BP's commitment to America has never been stronger – (1 may, 2013)
https://www.youtube.com/watch?v=LLBTT7Mvk2Q

8.3 Protocol Manual labeling Tweets

Tweets that contain a negative sentiment towards the subject BP are labeled = Negative
Example: Amazing that these douche bags @BP_America @BPGlobalPR even have the nerve to deny. Profoundly immoral http://huff.to/1kXUvJQ =Negative

Tweets that contain a positive sentiment towards the subject BP are labeled = Positive
Example: Delighted to have joined the team @BP_America yesterday! Looking forward to this next chapter = Positive

Tweets that do neither contain a positive or negative sentiment towards the subject BP are labeled = Neutral
Example: @mike_petriello took me a second to realize @BP_America was British Petroleum and not @baseballpro = Neutral

News Tweets that contain a negative decision or statement towards the subject BP are labeled = Negative
Example: @JosephFKaminski @kimbo1pow @BP_America Dumped Dispersants worse than the oil even stronger Pathogen http://www.huffingtonpost.com/2013/04/25/ = Negative

News Tweets that contain a positive decision or statement towards the subject BP are labeled = Positive
Example: Michael Gibbs was promoted to Head of #IT @BP_America. Congrats, Michael! http://ow.ly/rHxUe = Positive

News Tweets that neither contain a negative or positive decision or statement towards the subject BP are labeled = Neutral
Example: releases/bp-announces-significant-discovery-in-the-deepwater-gulf-of-mexi.html \xe2\u20ac\xa6 = Neutral

Sarcastic Tweets toward the subject BP are labeled = Negative
Example: hi @BP_America, remember when you guys spilled billions of gallons of oil into the ocean? haha cool = Negative

Tweets that are off-topic or unrelated to the subject BP are removed
Example: What a beautiful view from the balcony..

8.4 Results classifier 1

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,470309	0,374291	0,496296	0,379962	0,481663	0,373106	0,468983	0,373106	0,47981	0,381132	0,474328	0,366038
Neutral	0,796063	0,748937	0,792683	0,750532	0,793503	0,748937	0,791135	0,753645	0,793814	0,748481	0,790988	0,752051
Negative	0,749306	0,837408	0,745398	0,831783	0,745663	0,833592	0,74799	0,830101	0,746435	0,832428	0,749126	0,83133
Average run	0,671893	0,653545	0,678126	0,654092	0,673609	0,651878	0,669369	0,652284	0,673353	0,654014	0,671481	0,649806
Average precision	0,672787509											
Average recall	0,652380969											
F1 score total	0,662427117											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,486747	0,383302	0,475962	0,374291	0,476998	0,371698	0,474453	0,369318
Neutral	0,79483	0,746812	0,795286	0,747269	0,791868	0,744991	0,789929	0,748632
Negative	0,745329	0,835207	0,74645	0,836893	0,74446	0,834627	0,747823	0,831526
Average run	0,675635	0,655107	0,672566	0,652818	0,671109	0,650439	0,670735	0,649825
Average precision	0,672787509							
Average recall	0,652380969							
F1 score total	0,662427117							

Naïve Bayes results classification BP data (un pre-processed data)

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,467492	0,284906	0,472669	0,277358	0,491803	0,283554	0,477707	0,283554	0,480892	0,284906	0,472492	0,275472
Neutral	0,772572	0,756609	0,783826	0,765492	0,782178	0,768856	0,781638	0,765725	0,776304	0,764509	0,779938	0,768786
Negative	0,744219	0,823575	0,755045	0,841738	0,760391	0,842818	0,757862	0,840372	0,756766	0,83482	0,761136	0,840108
Average run	0,661428	0,621697	0,670513	0,628196	0,678124	0,631743	0,672402	0,629884	0,671321	0,628078	0,671189	0,628122
Average precision	0,670211973											
Average recall	0,627176164											
F1 score total	0,647980296											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,485342	0,281132	0,471947	0,270833	0,459807	0,270321	0,477273	0,277358
Neutral	0,779261	0,76315	0,781591	0,761167	0,776741	0,759271	0,781182	0,767407
Negative	0,754875	0,839985	0,753804	0,844634	0,751479	0,836497	0,758127	0,840372
Average run	0,673159	0,628089	0,669114	0,625545	0,662676	0,62203	0,672194	0,628379
Average precision	0,670211973							
Average recall	0,627176164							
F1 score total	0,647980296							

Naïve Bayes results classification BP data (pre-processed data)

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,732283	0,179537	0,746032	0,181467	0,74359	0,167954	0,734513	0,734513	0,160232	0,72807	0,160232	0,738095
Neutral	0,833845	0,933419	0,838784	0,936426	0,836907	0,939003	0,835965	0,835965	0,933419	0,832822	0,93299	0,835838
Negative	0,650759	0,862069	0,653846	0,881844	0,651982	0,878338	0,626327	0,626327	0,880597	0,637131	0,862857	0,632135
Average run	0,738963	0,658342	0,746221	0,666579	0,74416	0,661765	0,732268	0,732268	0,658083	0,732674	0,652026	0,735356
Average precision	0,699493937											
Average recall	0,71045333											
F1 score total	0,70493104											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,179537	0,737705	0,173745	0,723577	0,171815	0,783333	0,764228	0,181467
Neutral	0,931701	0,833972	0,934278	0,838536	0,934708	0,935997	0,836022	0,935137
Negative	0,859195	0,640693	0,855491	0,64346	0,881503	0,630573	0,647436	0,868195
Average run	0,656811	0,737457	0,654505	0,735191	0,662675	0,783301	0,749228	0,6616
Average precision	0,699493937							
Average recall	0,71045333							
F1 score total	0,70493104							

Naïve Bayes results classification Niek Sanders data (un pre-processed data)

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,44	0,309524	0,415902	0,294372	0,422222	0,28976	0,433022	0,28976	0,434251	0,308026	0,420245	0,297826
Neutral	0,760921	0,824847	0,759529	0,82263	0,76233	0,826796	0,762911	0,826972	0,765684	0,820683	0,764983	0,825777
Negative	0,522634	0,494163	0,518595	0,490234	0,520325	0,498054	0,525773	0,498047	0,520792	0,513672	0,519348	0,497076
Average run	0,574518	0,542845	0,564676	0,535746	0,568292	0,538204	0,573902	0,53826	0,573576	0,54746	0,568192	0,540226
Average precision	0,571078472											
Average recall	0,54164417											
F1 score total	0,555972016											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,410494	0,28913	0,428571	0,299349	0,425868	0,292842	0,435435	0,315904
Neutral	0,766825	0,824669	0,763158	0,828149	0,759529	0,823049	0,765766	0,82272
Negative	0,535857	0,523346	0,532922	0,503891	0,520243	0,5	0,518219	0,498054
Average run	0,571058	0,545715	0,574884	0,543796	0,568547	0,53863	0,57314	0,54556
Average precision	0,571078472							
Average recall	0,54164417							
F1 score total	0,555972016							

Naïve Bayes results classification Niek Sanders data (pre-processed data)

8.5 Results classifier 2

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,763158	0,164151	0,745455	0,154717	0,783784	0,164151	0,732759	0,164151	0,765766	0,160377	0,773109	0,173585
Neutral	0,799272	0,733313	0,797611	0,729812	0,798156	0,735883	0,798811	0,734062	0,799934	0,73558	0,803589	0,734062
Negative	0,690287	0,873937	0,686853	0,872776	0,689761	0,87007	0,690236	0,872003	0,689297	0,871616	0,690433	0,876257
Average run	0,750906	0,590467	0,743306	0,585768	0,757234	0,590035	0,740602	0,590072	0,751665	0,589191	0,75571	0,594635
Average precision	0,750384615											
Average recall	0,5908456											
F1 score total	0,661126544											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,775701	0,156604	0,754237	0,167925	0,767241	0,167925	0,739496	0,166038
Neutral	0,80171	0,740134	0,802121	0,734669	0,80278	0,736491	0,802445	0,737401
Negative	0,692213	0,873163	0,690992	0,875097	0,692543	0,876257	0,691789	0,873163
Average run	0,756541	0,589967	0,749117	0,592563	0,754188	0,593557	0,744577	0,592201
Average precision	0,750384615							
Average recall	0,5908456							
F1 score total	0,661126544							

Multinomial Naïve Bayes results classification BP data (un pre-processed data)

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,699605	0,333962	0,685824	0,337736	0,7	0,343396	0,69434	0,343396	0,691176	0,354717	0,700389	0,339623
Neutral	0,797876	0,75258	0,797996	0,749545	0,796779	0,751063	0,79845	0,750455	0,799417	0,748937	0,794202	0,756831
Negative	0,728197	0,858855	0,72635	0,858082	0,72381	0,852282	0,72614	0,856148	0,72772	0,858855	0,730259	0,851121
Average run	0,741892	0,648466	0,736723	0,648454	0,740196	0,648913	0,739643	0,65	0,739438	0,65417	0,741617	0,649192
Average precision	0,743208432											
Average recall	0,651777689											
F1 score total	0,694496765											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,70412	0,354717	0,714844	0,345283	0,71374	0,35283	0,734615	0,360377
Neutral	0,802269	0,751366	0,79839	0,75258	0,79799	0,747116	0,801727	0,761081
Negative	0,726946	0,859629	0,72778	0,858082	0,722258	0,855762	0,733047	0,856922
Average run	0,744445	0,655237	0,747004	0,651982	0,744663	0,651903	0,756463	0,65946
Average precision	0,743208432							
Average recall	0,651777689							
F1 score total	0,694496765							

Multinomial Naïve Bayes results classification BP data (pre-processed data)

8.6 Results classifier 3

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,125919	0,807547	0,142437	0,822642	0,163904	0,820755	0,17498	0,820755	0,179984	0,824528	0,18096	0,832075	0,188045	0,854717	0,226954	0,826642
Neutral	0,573541	0,22374	0,590447	0,405282	0,59878	0,476624	0,62176	0,480571	0,640972	0,464481	0,656677	0,452338	0,655067	0,457195	0,666791	0,541287
Negative	0,568456	0,386448	0,688419	0,289637	0,709877	0,311292	0,725689	0,386698	0,699687	0,431555	0,696009	0,458623	0,70329	0,462877	0,733788	0,49884
Average run	0,422639	0,472579	0,473768	0,505853	0,490853	0,536223	0,507476	0,562674	0,506881	0,573521	0,511216	0,581012	0,515467	0,591596	0,542511	0,622256
F1 score total	0,446215619		0,489285089		0,512536252		0,533651779		0,538145708		0,543883718		0,550914349		0,579653982	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,239782	0,830189	0,235772	0,820755	0,253421	0,803774	0,246971	0,807547	0,215809	0,783019	0,258665	0,760377	0,273475	0,75283	0,295559	0,728302
Neutral	0,687905	0,580146	0,698513	0,570431	0,714753	0,597146	0,729689	0,583485	0,73814	0,590468	0,743455	0,603522	0,752048	0,613236	0,757261	0,641166
Negative	0,784085	0,544857	0,787733	0,571152	0,787557	0,602088	0,785609	0,62065	0,77974	0,581428	0,780533	0,657386	0,776159	0,679814	0,781857	0,699923
Average run	0,570591	0,65173	0,574006	0,654113	0,585243	0,667669	0,587423	0,670561	0,577896	0,651638	0,594218	0,673762	0,600561	0,68196	0,611559	0,689797
F1 score total	0,608467373		0,611446925		0,623745119		0,626244558		0,612555791		0,63149464		0,638677229		0,648326063	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,283453	0,743396	0,190517	0,833962	0,144945	0,898113	0,133817	0,90566	0,135767	0,898113
Neutral	0,762844	0,62204	0,821262	0,426837	0,824066	0,301457	0,824413	0,266545	0,84507	0,236794
Negative	0,774636	0,699149	0,763667	0,702243	0,764706	0,568059	0,755973	0,513921	0,733973	0,562258
Average run	0,606978	0,688195	0,591815	0,654347	0,577906	0,58921	0,571401	0,562042	0,571603	0,565722
F1 score total	0,645040021		0,62151236		0,583503033		0,566682885		0,568647477	

Fuzzy Fingerprint $\mu 1(i)$ results classification BP pre-processed data

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,126068	0,807547	0,1365	0,775472	0,157895	0,769811	0,171085	0,779245	0,178437	0,80566	0,174483	0,828302	0,164811	0,866038	0,206832	0,833962
Neutral	0,62069	0,229508	0,566216	0,419247	0,569001	0,498179	0,582578	0,50759	0,597923	0,506982	0,605831	0,485732	0,609421	0,443837	0,607818	0,575896
Negative	0,571508	0,397138	0,679167	0,252127	0,704883	0,256767	0,707815	0,308198	0,691176	0,327146	0,694334	0,336427	0,698206	0,331013	0,743056	0,331013
Average run	0,439422	0,478065	0,460627	0,482282	0,47726	0,508252	0,48716	0,531678	0,489179	0,546596	0,491549	0,550153	0,490813	0,546963	0,519235	0,58029
F1 score total	0,457929432		0,471205923		0,492268735		0,508446037		0,51629621		0,519202935		0,517368593		0,548067604	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,2	0,866038	0,21467	0,828302	0,250911	0,779245	0,246869	0,781132	0,27106	0,75283	0,250825	0,716981	0,273558	0,733962	0,287926	0,701887
Neutral	0,661765	0,519126	0,701626	0,523983	0,714659	0,581664	0,732866	0,571342	0,745113	0,578628	0,74061	0,574681	0,760887	0,583485	0,761206	0,598057
Negative	0,786414	0,465584	0,784252	0,577726	0,766683	0,617556	0,768591	0,643465	0,76437	0,701006	0,752458	0,680588	0,74736	0,711524	0,746245	0,730085
Average run	0,549393	0,616916	0,566849	0,643337	0,577418	0,659488	0,582776	0,665313	0,593514	0,677488	0,581298	0,657417	0,593935	0,676324	0,598459	0,676676
F1 score total	0,581199758		0,602676032		0,615730178		0,621315094		0,632727242		0,617018569		0,632457479		0,635168588	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,27753	0,703774	0,233475	0,726415	0,212219	0,74717	0,208735	0,775472	0,183724	0,843396
Neutral	0,769844	0,585914	0,795249	0,508197	0,810219	0,471767	0,813181	0,453248	0,841259	0,381299
Negative	0,741696	0,733952	0,736069	0,755994	0,738005	0,74942	0,734741	0,740139	0,745974	0,716551
Average run	0,596357	0,674546	0,588265	0,663535	0,586814	0,656119	0,585552	0,656286	0,590319	0,647082
F1 score total	0,633046365		0,623636844		0,619534282		0,618904917		0,617398639	

Fuzzy Fingerprint $\mu 2(i)$ results classification BP pre-processed data

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,106733	0,649057	0,114943	0,603774	0,136646	0,539623	0,148114	0,6	0,189493	0,571698	0,164083	0,579245	0,111768	0,856604	0,121409	0,916981
Neutral	0,590103	0,191748	0,602489	0,279126	0,651796	0,341323	0,638465	0,418993	0,642223	0,555407	0,644527	0,527002	0,69331	0,355279	0,769231	0,194175
Negative	0,510628	0,418182	0,52381	0,425532	0,514275	0,515667	0,549262	0,446422	0,588235	0,437137	0,596206	0,425532	0,684848	0,174855	0,635152	0,387234
Average run	0,402488	0,419662	0,413747	0,436144	0,434239	0,465538	0,44528	0,488471	0,473317	0,521414	0,468272	0,510593	0,496642	0,462246	0,508597	0,499463
F1 score total	0,410895592		0,424650271		0,449343885		0,465876984		0,496202959		0,48851777		0,478827025		0,50398894	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,164504	0,950943	0,154525	0,939623	0,160189	0,841509	0,164963	0,841509	0,169736	0,824528	0,172287	0,807547	0,16323	0,796226	0,182	0,760377
Neutral	0,712412	0,256675	0,709754	0,263653	0,706843	0,444568	0,720483	0,444568	0,722693	0,523058	0,724242	0,539745	0,742612	0,525485	0,742316	0,574029
Negative	0,782456	0,177563	0,815832	0,22824	0,821398	0,496325	0,819133	0,496325	0,826305	0,529594	0,824472	0,543907	0,808981	0,553578	0,807292	0,572147
Average run	0,553124	0,461727	0,560037	0,477172	0,56281	0,594134	0,568193	0,594134	0,572911	0,625727	0,573667	0,6304	0,571608	0,625097	0,577202	0,635518
F1 score total	0,503309918		0,515294167		0,578048182		0,580874078		0,598155404		0,600696994		0,5971568		0,60495807	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,177933	0,74717	0,190733	0,518868	0,183382	0,437736	0,148518	0,425	0,141018	0,511321
Neutral	0,736529	0,574333	0,738286	0,633495	0,747377	0,689624	0,761459	0,698119	0,767964	0,681735
Negative	0,807432	0,582979	0,806782	0,685493	0,82543	0,683559	0,811811	0,67118	0,820961	0,641779
Average run	0,573965	0,634827	0,5786	0,612619	0,585396	0,60364	0,573929	0,5981	0,576648	0,611612
F1 score total	0,602863843		0,595123769		0,594377892		0,585765326		0,593615295	

Fuzzy Fingerprint $\mu 1(i)$ results classification BP un pre-processed data

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,111789	0,656604	0,131096	0,633962	0,184975	0,55283	0,181625	0,645283	0,225188	0,620755	0,226797	0,613208	0,162975	0,777358	0,171798	0,85283
Neutral	0,60688	0,224818	0,638889	0,327973	0,654206	0,382282	0,659355	0,465109	0,647448	0,623483	0,645628	0,67108	0,678384	0,550364	0,734583	0,368629
Negative	0,518536	0,416634	0,528757	0,441006	0,506377	0,568279	0,564231	0,480851	0,622748	0,427853	0,647969	0,388781	0,704715	0,329594	0,667451	0,548936
Average run	0,412402	0,413747	0,432914	0,467647	0,448519	0,50113	0,468403	0,530414	0,498461	0,557364	0,506798	0,55769	0,515358	0,552439	0,52461	0,590132
F1 score total	0,413073208		0,449610801		0,47336729		0,497484014		0,526269436		0,531027397		0,533254513		0,555445466	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,164504	0,884906	0,154525	0,892453	0,160189	0,89434	0,164963	0,883019	0,169736	0,886792	0,172287	0,886792	0,16323	0,896226	0,182	0,858491
Neutral	0,712412	0,461468	0,709754	0,454794	0,706843	0,48574	0,720483	0,488774	0,722693	0,491808	0,724242	0,507585	0,742612	0,472694	0,742316	0,534891
Negative	0,782456	0,431335	0,815832	0,390716	0,821398	0,377176	0,819133	0,423985	0,826305	0,447195	0,824472	0,437911	0,808981	0,439072	0,807292	0,479691
Average run	0,553124	0,59257	0,560037	0,579321	0,56281	0,585752	0,568193	0,598593	0,572911	0,608599	0,573667	0,610763	0,571608	0,602664	0,577202	0,624357
F1 score total	0,572167744		0,569515579		0,574052035		0,582996766		0,59021599		0,59163413		0,586725181		0,5998546	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,177933	0,864151	0,190733	0,8	0,183382	0,824528	0,148518	0,841509	0,141018	0,867925
Neutral	0,736529	0,526699	0,738286	0,587985	0,747377	0,583434	0,761459	0,493932	0,767964	0,46693
Negative	0,807432	0,462282	0,806782	0,487814	0,82543	0,464603	0,811811	0,398839	0,820961	0,46693
Average run	0,573965	0,617711	0,5786	0,625267	0,585396	0,624189	0,573929	0,578094	0,576648	0,600595
F1 score total	0,59503497		0,601028944		0,604170391		0,576004043		0,588377489	

Fuzzy Fingerprint $\mu 2(i)$ results classification BP un pre-processed data

8.7 Results classifier 4

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,503145	0,301887	0,517241	0,28125	0,476341	0,285444	0,477707	0,285444	0,496855	0,298113	0,508475	0,288462
Neutral	0,778673	0,763758	0,777963	0,784776	0,774588	0,771055	0,781638	0,765725	0,778295	0,76315	0,790188	0,77913
Negative	0,757183	0,837922	0,767572	0,831675	0,763617	0,830751	0,757862	0,840372	0,756217	0,837146	0,768476	0,849071
Average run	0,679667	0,634522	0,687592	0,632567	0,671515	0,629084	0,672402	0,630514	0,677122	0,632803	0,689046	0,638888
Average precision	0,680348172											
Average recall	0,632980613											
F1 score total	0,655810195											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,4791	0,281132	0,517007	0,287879	0,505017	0,284906	0,51	0,291429
Neutral	0,780223	0,765278	0,784314	0,766423	0,788965	0,769535	0,786736	0,774483
Negative	0,753927	0,836237	0,754671	0,845021	0,759529	0,851593	0,758922	0,840372
Average run	0,671083	0,627549	0,685331	0,633108	0,684504	0,635344	0,685219	0,635428
Average precision	0,680348172							
Average recall	0,632980613							
F1 score total	0,655810195							

Naïve Bayes results classification model 4

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,715953	0,347826	0,715385	0,350943	0,714286	0,339623	0,72973	0,339623	0,721805	0,362264	0,695817	0,345283
Neutral	0,79176	0,753725	0,795498	0,751063	0,791401	0,754402	0,792038	0,755009	0,793778	0,751366	0,79515	0,756527
Negative	0,727424	0,848567	0,725658	0,853055	0,72664	0,848028	0,728329	0,848028	0,727693	0,851508	0,731497	0,852282
Average run	0,745046	0,650039	0,745514	0,651687	0,744109	0,647351	0,750032	0,647553	0,747759	0,655046	0,740821	0,651364
Average precision	0,745337444											
Average recall	0,650906049											
F1 score total	0,694928433											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,691729	0,34717	0,721805	0,362264	0,72973	0,356604	0,714286	0,339623
Neutral	0,793103	0,754098	0,793778	0,751366	0,792038	0,755009	0,791401	0,754402
Negative	0,729748	0,849961	0,727693	0,851508	0,728329	0,848028	0,72664	0,848028
Average run	0,738193	0,65041	0,747759	0,655046	0,750032	0,653214	0,744109	0,647351
Average precision	0,745337444							
Average recall	0,650906049							
F1 score total	0,694928433							

Multinomial Naïve Bayes results classification model 4

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,125733	0,809434	0,1696	0,84127	0,164657	0,824528	0,1756	0,828302	0,173207	0,824528	0,177387	0,837736	0,185874	0,849057	0,214391	0,820755
Neutral	0,622561	0,222829	0,596437	0,406497	0,601908	0,478749	0,630399	0,469642	0,643794	0,447177	0,657754	0,448087	0,657277	0,467517	0,65939	0,531876
Negative	0,568994	0,400232	0,694737	0,306265	0,71831	0,315545	0,713599	0,401779	0,701689	0,433875	0,707156	0,454756	0,712029	0,45321	0,730278	0,486852
Average run	0,439096	0,477498	0,486924	0,51801	0,494958	0,539608	0,506533	0,566574	0,50623	0,568527	0,514099	0,580193	0,518393	0,589928	0,534687	0,613161
F1 score total	0,457492784		0,501986602		0,516319558		0,534873765		0,535572591		0,545150075		0,55185192		0,571241352	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,237316	0,820755	0,225272	0,820755	0,362582	0,866265	0,361181	0,807547	0,249851	0,790566	0,275792	0,771698	0,278437	0,75283	0,2726	0,733962
Neutral	0,680932	0,594111	0,702273	0,562842	0,705566	0,596539	0,722409	0,586217	0,733786	0,590771	0,746125	0,613843	0,745663	0,613236	0,755498	0,615361
Negative	0,796829	0,524749	0,794454	0,564965	0,782698	0,58778	0,623041	0,614849	0,784719	0,631477	0,777176	0,66628	0,772928	0,677881	0,781739	0,695282
Average run	0,571692	0,646538	0,573999	0,64952	0,616949	0,683528	0,568877	0,669538	0,589452	0,670938	0,599698	0,68394	0,599009	0,681316	0,603279	0,681535
F1 score total	0,606816056		0,609429176		0,648534068		0,615116582		0,627560885		0,639054937		0,637516826		0,640023814	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,280907	0,724528	0,192913	0,832075	0,14564	0,888679	0,135823	0,90566	0,134015	0,915094
Neutral	0,759005	0,620522	0,817469	0,44323	0,825563	0,311779	0,82491	0,277474	0,832827	0,249545
Negative	0,777021	0,70611	0,764328	0,691029	0,774327	0,5785	0,769231	0,525909	0,756098	0,527456
Average run	0,605644	0,68372	0,59157	0,655445	0,581843	0,592986	0,576655	0,569681	0,574313	0,564031
F1 score total	0,642318364		0,621871666		0,587361784		0,573146665		0,569125856	

Fuzzy Fingerprint $\mu 1(i)$ results model 4

	k =5		k = 10		k = 15		k = 20		k = 25		k = 30		k = 50		k = 100	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,12588	0,809434	0,136531	0,767925	0,157364	0,766038	0,170672	0,790566	0,168726	0,824528	0,172009	0,830189	0,166426	0,869811	0,200987	0,845283
Neutral	0,621287	0,228597	0,567457	0,421372	0,570537	0,499696	0,594398	0,496053	0,604714	0,475106	0,604325	0,483607	0,613035	0,456891	0,610836	0,561324
Negative	0,572067	0,395978	0,682604	0,259474	0,707937	0,258701	0,70398	0,328306	0,699675	0,333333	0,702303	0,33024	0,708017	0,324439	0,7513	0,335267
Average run	0,439745	0,478003	0,462198	0,482924	0,478613	0,508145	0,489683	0,538309	0,491038	0,544323	0,492879	0,548012	0,495826	0,550381	0,521041	0,580624
F1 score total	0,458076589		0,472333331		0,492936898		0,512845969		0,516309405		0,518985184		0,521680847		0,54922139	

	k =200		k = 300		k = 400		k = 500		k = 600		k = 700		k = 800		k = 900	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,206976	0,850943	0,206019	0,839623	0,25335	0,784906	0,248927	0,766038	0,248148	0,758491	0,272095	0,737736	0,273826	0,726415	0,270231	0,70566
Neutral	0,650127	0,54493	0,696995	0,506982	0,709037	0,585914	0,72974	0,579539	0,739387	0,571038	0,750675	0,590468	0,750965	0,590468	0,760458	0,585003
Negative	0,8	0,454756	0,783711	0,561872	0,77112	0,609472	0,769764	0,643852	0,763134	0,6628	0,753568	0,694122	0,752693	0,70263	0,753612	0,726218
Average run	0,552367	0,616877	0,562241	0,636159	0,577836	0,660097	0,58281	0,663143	0,583556	0,66411	0,592113	0,674109	0,592495	0,673171	0,594767	0,672294
F1 score total	0,582842526		0,59692056		0,616233228		0,620386832		0,621232613		0,63045581		0,630261529		0,631158541	

	k =1000		k = 2000		k = 3000		k = 4000		k = 5000	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,276786	0,701887	0,228916	0,716981	0,213228	0,760377	0,203809	0,74717	0,185	0,837736
Neutral	0,76068	0,583789	0,796634	0,517304	0,811943	0,470553	0,809326	0,474196	0,828037	0,403461
Negative	0,747045	0,733179	0,738414	0,745553	0,742244	0,74942	0,750099	0,735886	0,759252	0,70611
Average run	0,594837	0,672951	0,587988	0,659946	0,589138	0,660117	0,587745	0,652417	0,590763	0,649102
F1 score total	0,631487698		0,621892435		0,622611158		0,618394445		0,618560063	

Fuzzy Fingerprint $\mu 2(i)$ results model

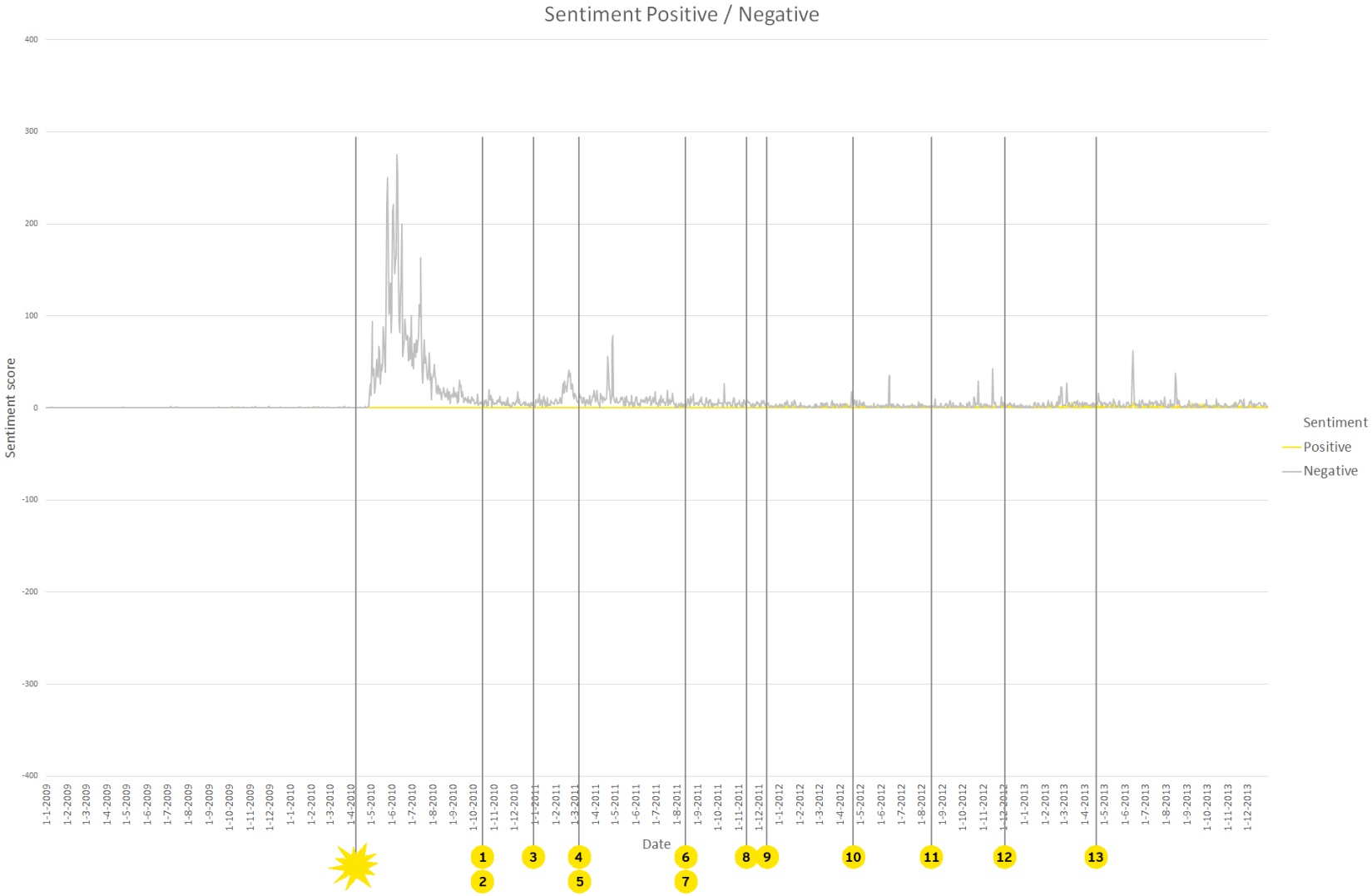
8.8 Results classifier 5

	Run 1		Run 2		Run 3		Run 4		Run 5		Run 6	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	0,995192	0,390566	0,987342	0,588679	0,990654	0,4	0,995238	0,4	0,995763	0,443396	1	0,486792
Neutral	0,931215	0,969945	0,9293	0,829994	0,932594	0,945052	0,918455	0,974499	0,91713	0,95082	0,921086	0,885853
Negative	0,933237	1	0,820114	0,999613	0,903429	0,998453	0,954898	0,99884	0,936934	0,999613	0,866622	1
Average run	0,953215	0,786837	0,912252	0,806095	0,942226	0,781168	0,956197	0,791113	0,949942	0,797943	0,929236	0,790882
Average precision	0,9434648											
Average recall	0,800114201											
F1 score total	0,865896623											

	Run 7		Run 8		Run 9		Run 10	
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
Positive	1	0,486792	0,991701	0,450943	0,993266	0,556604	0,992337	0,488679
Neutral	0,921086	0,885853	0,923601	0,946873	0,932477	0,94748	0,935185	0,981178
Negative	0,866622	1	0,925501	0,999227	0,934924	1	0,958039	0,99768
Average run	0,929236	0,790882	0,946935	0,799014	0,953556	0,834695	0,961854	0,822512
Average precision	0,9434648							
Average recall	0,800114201							
F1 score total	0,865896623							

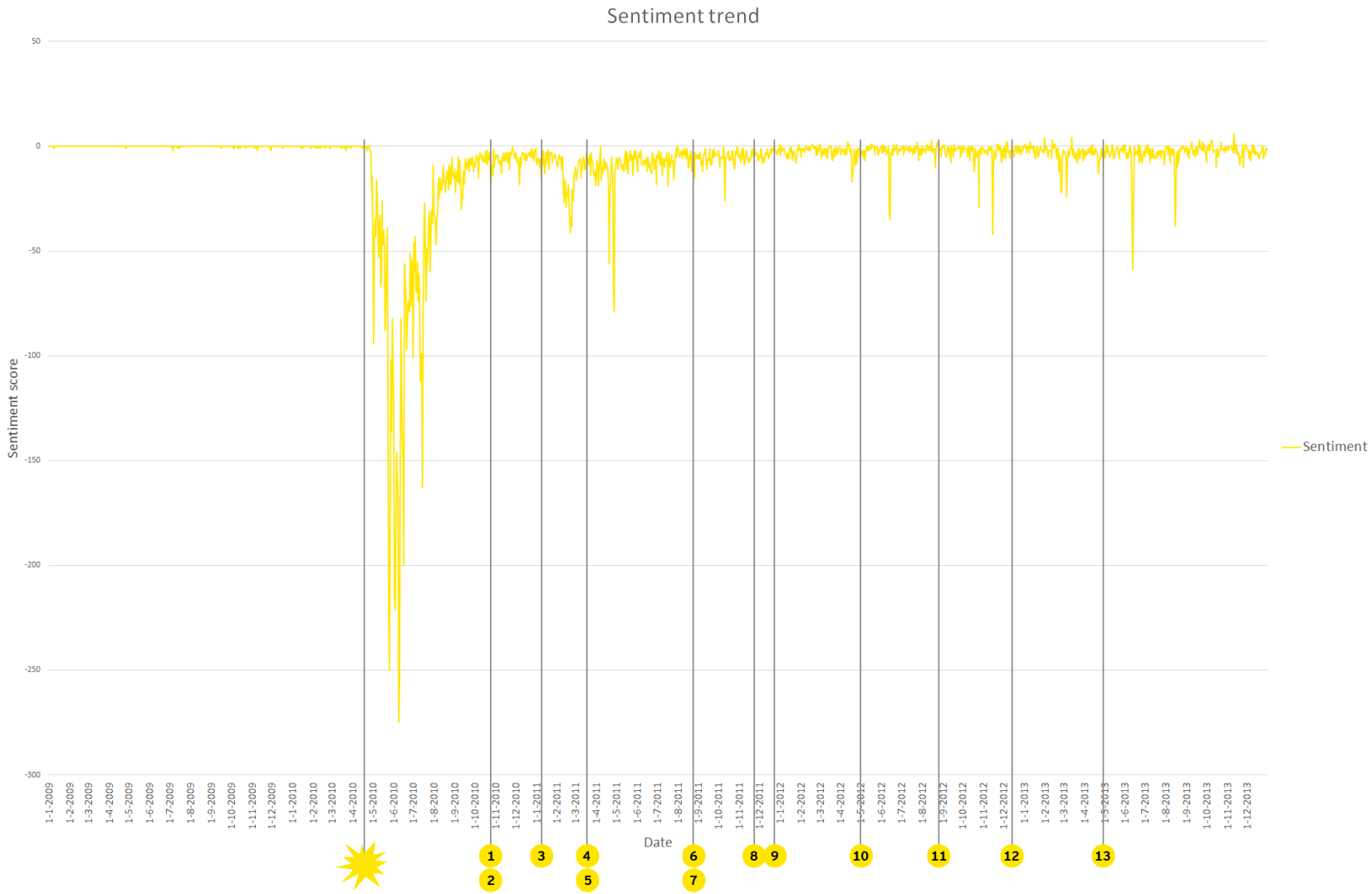
Penalized Multinomial Naïve Bayes model results

8.9 Sentiment trend

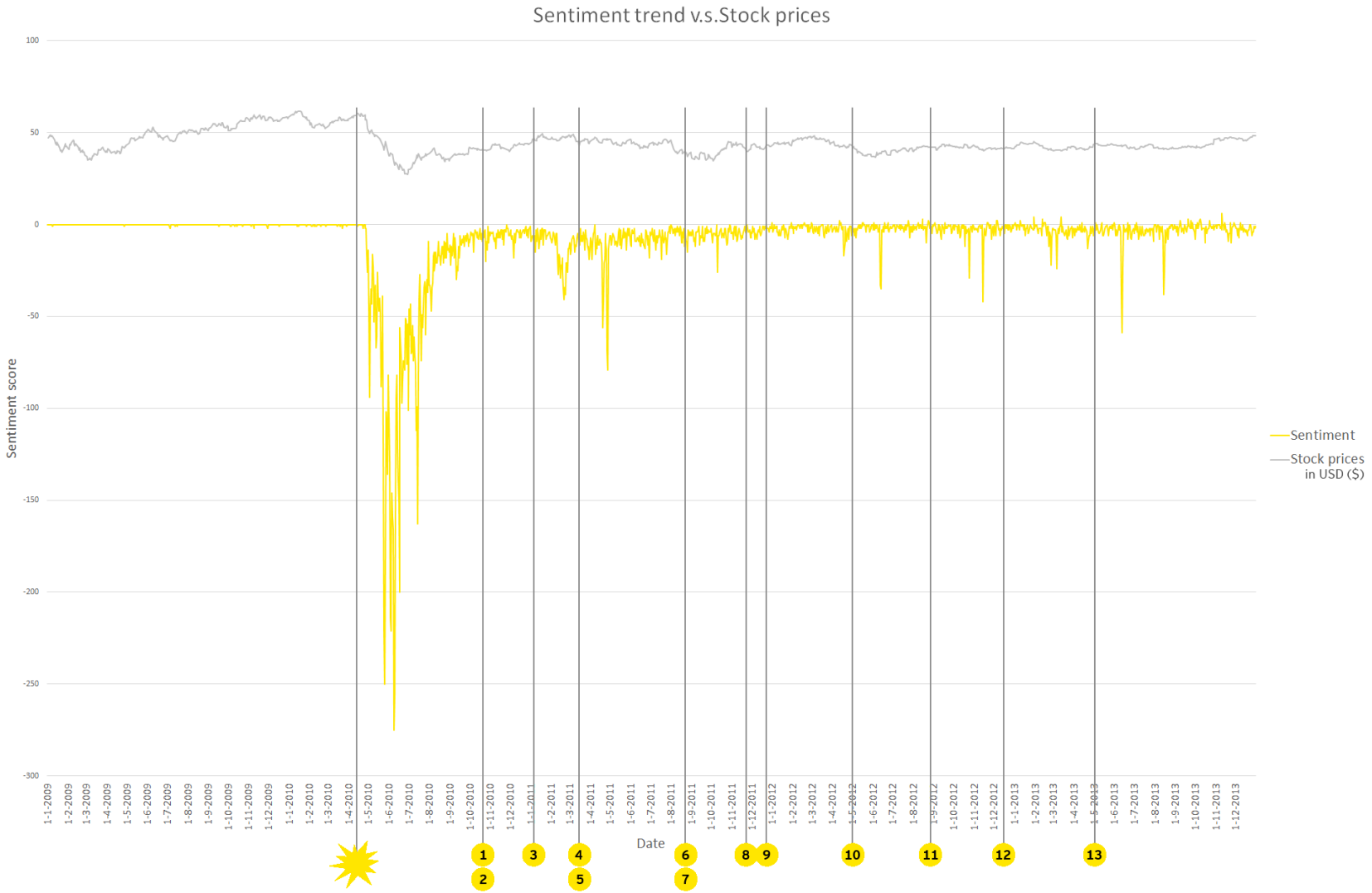


sentiment positive / negative

Classified



Classified sentiment trend



Classified sentiment trend v.s. stock prices

8.10 Web crawler code

```
import csv
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time


### webdriver and opening webpage
chrome_path = r"C:\chromedriver_win32\chromedriver.exe"
driver = webdriver.Chrome(chrome_path)
driver.get("https://twitter.com/search?l=en&q=to%3ABP_america%20since%3A200
9-01-01%20until%3A2014-01-01&src=typd")

# scrolling down to bottom page
scroll_pause = 2
last_height = driver.execute_script("return document.body.scrollHeight")

while True:
    # scroll down to bottom
    driver.execute_script("window.scrollTo(0,
document.body.scrollHeight);")

    # waiting on page to load
    time.sleep(scroll_pause)

    # calculating new height and compare to last scroll height
    new_height = driver.execute_script("return document.body.scrollHeight")
    if new_height == last_height:
        break
    last_height = new_height


### extracting data

# names
names = driver.find_elements_by_class_name("FullNameGroup")

# Tweets
Tweets = driver.find_elements_by_class_name("js-Tweet-text-container")

# dates
dates = driver.find_elements_by_class_name("Tweet-timestamp")


### making list per category
Tweet_pages = len(names)

namelist = []
datelist = []
Tweetlist = []

for i in range(Tweet_pages):
    namelist.append(names[i].text)
    datelist.append(dates[i].text)
    Tweetlist.append(Tweets[i].text)


### writing to csv
row1 = zip(namelist)
```

```
with open('testname.csv', 'w', encoding="utf-8") as f:
    writer = csv.writer(f)
    for row in row1:
        writer.writerow(row)

row2 = zip(datelist)
with open('testdate.csv', 'w', encoding="utf-8") as f:
    writer = csv.writer(f)
    for row in row2:
        writer.writerow(row)

row3 = zip(Tweetlist)
with open('testTweet.csv', 'w', encoding="utf-8") as f:
    writer = csv.writer(f)
    for row in row3:
        writer.writerow(row)


### clean up and close browser
driver.close()
```

8.11 Data cleaning code 1

```
import pandas as pd
from slangdict import slangdict
from appos import appos
import string
import itertools
from autocorrect import spell
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer


### importing excel
excelfile = "full-corpus.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('full-corpus')

# check column names
print(dataframe1.columns)



### removing duplicate Tweets
dataframe1.duplicated()
dataframe1.drop_duplicates(keep='first', inplace=True)


### removing irrelevant Tweets
dataframe1 = dataframe1[dataframe1.Sentiment.str.contains("irrelevant") ==
False]


### removing ReTweets
dataframe1 = dataframe1[dataframe1.TweetText.str.contains("RT") == False]

### conversion of ’ to '
# define pattern
patterncomma = r'’'

# replacing regex functions based on column Tweet
dataframe1.TweetText.replace({patterncomma: "'"},inplace=True, regex=True)


### hasthag removal
# define pattern
patternhash = r'#'

# replacing regex functions based on column Tweet
dataframe1.TweetText.replace({patternhash: ""},inplace=True, regex=True)


### removal of usernames
# define pattern
patternuser = r'(@\w+)'

# replacing regex functions based on column Tweet
```

```
dataframe1.TweetText.replace({patternuser: ""},inplace=True, regex=True)


#### removing capital letters
dataframe1['TweetText'] = dataframe1['TweetText'].str.lower()


### HTML REMOVING
# define pattern \S anything but a space, + match one or more
patternhtml = r'http\S+'

# replacing regex functions based on column Tweet
dataframe1.TweetText.replace({patternhtml: ""},inplace=True, regex=True)


##### Apostrof removing (contraction words)
# calling dictionary of apostroph contracted words
dictionary = appos

#checking index and resetting (working)
dataframe1.reset_index(inplace=True)

# replacing contraction words
counter = list(range(0,2950))  ## place last index value here of datasets
last position
counter2 = range(0,10000) ## value that makes it possible to iterate over
all dictionary values
counter3 = [counter2] ## make list out of values
for j in counter:
    words = dataframe1['TweetText'][j].split()
    print(words)
    for x in counter3:
        reformed = [dictionary[x] if x in dictionary else x for x in words]
        reformed = " ".join(reformed)
        dataframe1.set_value(j, 'TweetText', reformed)
        counter2 = 0


### Removing punctuation
counter = list(range(0,2950))  ## place last index value here of datasets
last position
for k in counter:
    for c in string.punctuation:
        punct = dataframe1['TweetText'][k].replace(c," ")
        dataframe1.set_value(k, 'TweetText', punct)


### cleaning van slang
dictionary = slangdict


# replacing slang words (working)
counter = list(range(0,2950))  ## place last index value here of datasets
last position
counter2 = range(0,10000) ## value that makes it possible to iterate over
all dictionary values
counter3 = [counter2] ## make list out of values
for i in counter:
    words = dataframe1['TweetText'][i].split()
    for z in counter3:
        reformed = [dictionary[z] if z in dictionary else z for z in words]
```

```
        reformed = " ".join(reformed)
        dataframe1.set_value(i, 'TweetText', reformed)


### standardizing words
counter = list(range(0,2950))  ## place last index value here

for g in counter:
    words = dataframe1['TweetText'][g]
    words = ''.join(''.join(s)[:2] for _, s in itertools.groupby(words))
    words = ''.join(words)
    dataframe1.set_value(g, 'TweetText', words)



### spelling correction
counter = list(range(0,2950))  ## place last index value here

for k in counter:
    words = dataframe1['TweetText'][k].split()
    for l in range(len(words)):
        correctword = spell(words[l])
        if correctword != words[l]:
            words[l] = correctword
    words = " ".join(words)
    dataframe1.set_value(k, 'TweetText', words)


### filter out stopwords
counter = list(range(0,2950))  ## place last index value here of datasets
last position
counter2 = range(0,10000) ## value that makes it possible to iterate over
all dictionary values
counter3 = [counter2] ## make list out of values

stopwords_list = set(stopwords.words('english'))
print(stopwords_list)

for m in counter:
    words = dataframe1['TweetText'][m].split()
    for c in counter3:
        words = [c for c in words if c not in stopwords_list]
        words = " ".join(words)
        dataframe1.set_value(m, 'TweetText', words)


#### removing capital letters
dataframe1['TweetText'] = dataframe1['TweetText'].str.lower()


### Lemmitization
lemmatizer = WordNetLemmatizer()
counter = list(range(0,2950))  ## place last index value here

for g in counter:
    words = dataframe1['TweetText'][g].split()
    for f in range(len(words)):
        lemmaword = lemmatizer.lemmatize(words[f])
        if lemmaword != words[f]:
            words[f] = lemmaword
    words = " ".join(words)
```

```
    dataframe1.set_value(g, 'TweetText', words)


### writing dataframe to excel
dataframe1 = dataframe1.applymap(lambda x:
x.encode('unicode_escape').decode('utf-8') if isinstance(x, str) else x)
dataframe1.to_excel("cleaned_sanders.xlsx", encoding = "utf-8")
```

8.12 Classification code 1 Naïve Bayes -BP

```
import pandas as pd
import nltk
import random
import csv
import math
import numpy as np

### importing excel
excelfile = "BP_cycle_1_train_test.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')
print(dataframe1.columns)

# print(type(dataframe1["Tweet"]))

## reset index
dataframe1 = dataframe1.reset_index(drop=True)

### Importing the necessary information (tweet and classification)

# making list of seperate words with all lower cases
counter = list(range(0,6408))  ## place last index value
for i in counter:
    words = dataframe1['Tweet'][i].split()
    words = [string.lower() for string in words]
    dataframe1.set_value(i, 'Tweet', words)

### forming tuple of the combined two columns
dataframe1['Combined'] = list(zip(dataframe1.Tweet, dataframe1.Sentiment))


### making word features
def extract_words_tweet(tweets):
    all_words = []
    for (words, sentiment) in tweets:
        all_words.extend(words)
    return all_words

extracted_words = extract_words_tweet(dataframe1["Combined"])

def get_word_features(wordlist):
    wordlist = nltk.FreqDist(wordlist)
    wordlist_sorted = sorted(wordlist, key=lambda x: wordlist[x.lower()],
reverse=True)#[:2000]
    return wordlist_sorted


word_features =
get_word_features(extract_words_tweet(dataframe1["Combined"]))

def find_features(document):
    words = set(document)
    features = {}
```

```
    for w in word_features:
        features[w] = (w in words)
    return features

## making featuresets for all tweets with sentiment
counter = list(range(0,6408))  ## place last index value
featuresets = []

for j in counter:
    find_feat = find_features(dataframe1["Combined"][j][0])
    sentiment = dataframe1["Sentiment"][j]
    tuple = (find_feat, sentiment)
    featuresets.append(tuple)


### Shuffling data
featuresets_shuffled = random.sample(featuresets, len(featuresets))



numberfolds = 10

# we need to round down for all subsetsizes because of rounding errors
subsetsize_featuresets = math.floor(6408/numberfolds)

# list with only sentiment after shuffeling
sentiment_featuresets = [x[1] for x in featuresets_shuffled]
print(sentiment_featuresets)


## K-fold cross validation and classifier
testresults = []
classresults = []

for k in range(0,numberfolds):
    # train and test sets
    test_round =
featuresets_shuffled[k*subsetsize_featuresets:][:subsetsize_featuresets]
    train_round = featuresets_shuffled[:k*subsetsize_featuresets] +
featuresets_shuffled[(k+1)*subsetsize_featuresets:]

    # classifier
    classifier = nltk.NaiveBayesClassifier.train(train_round)
    classresults.append((nltk.classify.accuracy(classifier,
test_round))*100)
    print("naive bayes accuracy percent:",
(nltk.classify.accuracy(classifier, test_round))*100)
    print(classifier.show_most_informative_features(50))

    # making list of predicted sentiment classes (compare with actual
classes)
    for a in range(0, subsetsize_featuresets):
        testresults.append(classifier.classify(test_round[a][0]))


# list of all separate accuracies per test and train round
print(classresults)

#average accuracy classifier
print(sum(classresults) / float(len(classresults)))
```

```
# writing list with predicted classes
# making rows of the lists
rows1 = zip(testresults)
with open("predicted_classes.csv", "w") as f:
    writer = csv.writer(f)
    for row in rows1:
        writer.writerow(row)


## appending list so that we have the same structure as within the
classified list
sentiment_total_list = []

for O in range(0,numberfolds):
    # test sets
    test_sentiment =
sentiment_featuresets[O*subsetsize_featuresets:][:subsetsize_featuresets]
    sentiment_total_list.append(test_sentiment)
    print(sentiment_total_list)



# writing list with features and classified classes to csv
with open("classifier_classes.csv", "w") as g:
    writer = csv.writer(g)
    for row in sentiment_total_list:
        writer.writerow(sentiment_total_list)
```

8.13 Classification code 1 Naïve Bayes - Niek Sanders

```
import pandas as pd
import nltk
import random
import csv
import math

### importing excel
excelfile = "cleaned_sanders.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')
print(dataframe1.columns)

print(type(dataframe1["TweetText"]))

## reset index
dataframe1 = dataframe1.reset_index(drop=True)

### Importing the necessary information (Tweet and classification)

# making list of seperate words with all lower cases
counter = list(range(0,2949))  ## place last index value
for i in counter:
    words = dataframe1['TweetText'][i].split()
    words = [string.lower() for string in words]
    dataframe1.set_value(i, 'TweetText', words)

### forming tuple of the combined two columns
dataframe1['Combined'] = list(zip(dataframe1.TweetText,
dataframe1.Sentiment))


### making word features
def extract_words_Tweet(Tweets):
    all_words = []
    for (words, sentiment) in Tweets:
        all_words.extend(words)
    return all_words

extracted_words = extract_words_Tweet(dataframe1["Combined"])

def get_word_features(wordlist):
    wordlist = nltk.FreqDist(wordlist)
    wordlist_sorted = sorted(wordlist, key=lambda x: wordlist[x.lower()],
reverse=True)#[:2000]
    return wordlist_sorted


word_features =
get_word_features(extract_words_Tweet(dataframe1["Combined"]))

def find_features(document):
    words = set(document)
    features = {}
```

```
    for w in word_features:
        features[w] = (w in words)
    return features

## making featuresets for all Tweets with sentiment
counter = list(range(0,2949))  ## place last index value
featuresets = []

for j in counter:
    find_feat = find_features(dataframe1["Combined"][j][0])
    sentiment = dataframe1["Sentiment"][j]
    tuple = (find_feat, sentiment)
    featuresets.append(tuple)


### Shuffling data so that each category is represented equally in the
# to divide the proportions equally over train and test per topic
disttopic = dataframe1["Topic"].value_counts()
print(disttopic)

# shuffeling the data
numberfolds = 10


# we need to round down for all subsetsizes because of rounding errors
featuresets_apple = featuresets[:902]
featuresets_apple = random.sample(featuresets_apple,
len(featuresets_apple))
subsetsize_apple = math.floor(902/numberfolds)

featuresets_microsoft = featuresets[902:1638]
featuresets_microsoft = random.sample(featuresets_microsoft,
len(featuresets_microsoft))
subsetsize_microsoft = math.floor(736/numberfolds)

featuresets_google = featuresets[1638:2368]
featuresets_google = random.sample(featuresets_google,
len(featuresets_google))
subsetsize_google = math.floor(730/numberfolds)

featuresets_twitter = featuresets[2368:2949]
featuresets_twitter = random.sample(featuresets_twitter,
len(featuresets_twitter))
subsetsize_twitter = math.floor(581/numberfolds)

## making lists with only sentiment after shuffeling
sentiment_apple = [x[1] for x in featuresets_apple]
sentiment_microsoft = [x[1] for x in featuresets_microsoft]
sentiment_google = [x[1] for x in featuresets_google]
sentiment_twitter = [x[1] for x in featuresets_twitter]


## K-fold cross validation and classifier
testresults = []
classresults = []

for k in range(0,numberfolds):
    # train and test sets
    test_round_apple =
featuresets_apple[k*subsetsize_apple:][:subsetsize_apple]
    train_round_apple = featuresets_apple[:k*subsetsize_apple] +
```

```
featuresets_apple[(k+1)*subsetsize_apple:]

    test_round_microsoft =
featuresets_microsoft[k*subsetsize_microsoft:][:subsetsize_microsoft]
    train_round_microsoft = featuresets_microsoft[:k*subsetsize_microsoft]
+ featuresets_microsoft[(k+1)*subsetsize_microsoft:]

    test_round_google =
featuresets_google[k*subsetsize_google:][:subsetsize_google]
    train_round_google = featuresets_google[:k*subsetsize_google] +
featuresets_google[(k+1)*subsetsize_google:]

    test_round_twitter =
featuresets_twitter[k*subsetsize_twitter:][:subsetsize_twitter]
    train_round_twitter = featuresets_twitter[:k*subsetsize_twitter] +
featuresets_twitter[(k+1)*subsetsize_twitter:]

    testround_total = test_round_apple + test_round_microsoft +
test_round_google  + test_round_twitter
    traininground_total = train_round_apple + train_round_microsoft +
train_round_google  + train_round_twitter

    # classifier
    classifier = nltk.NaiveBayesClassifier.train(traininground_total)
    classresults.append((nltk.classify.accuracy(classifier,
testround_total))*100)
    print("naive bayes accuracy percent:",
(nltk.classify.accuracy(classifier, testround_total))*100)
    print(classifier.show_most_informative_features(50))

    # making list of predicted sentiment classes (compare with actual
classes)
    for a in range(0, subsetsize_apple):
        testresults.append(classifier.classify(test_round_apple[a][0]))

    for b in range(0,subsetsize_microsoft):
        testresults.append(classifier.classify(test_round_microsoft[b][0]))

    for c in range(0,subsetsize_google):
        testresults.append(classifier.classify(test_round_google[c][0]))

    for d in range(0,subsetsize_twitter):
        testresults.append(classifier.classify(test_round_twitter[d][0]))


# list of all seperate accuracies per test and train round
print(classresults)

#average accuracy classifier
print(sum(classresults) / float(len(classresults)))


# writing list with predicted classes
# making rows of the lists
rows1 = zip(testresults)
with open("predicted_classes.csv", "w") as f:
    writer = csv.writer(f)
    for row in rows1:
        writer.writerow(row)
```

```
## appending list so that we have the same structure as within the
classified list
# 90 of apple, 73 of microsoft, 73 of google and 58 of twitter again and
again
sentiment_total_list = []

for o in range(0,numberfolds):
    # test sets
    test_sentiment_apple =
sentiment_apple[o*subsetsize_apple:][:subsetsize_apple]
    sentiment_total_list.append(test_sentiment_apple)
    print(sentiment_total_list)
    test_sentiment_microsoft =
sentiment_microsoft[o*subsetsize_microsoft:][:subsetsize_microsoft]
    sentiment_total_list.append(test_sentiment_microsoft)
    print(sentiment_total_list)
    test_sentiment_google =
sentiment_google[o*subsetsize_google:][:subsetsize_google]
    sentiment_total_list.append(test_sentiment_google)
    print(sentiment_total_list)
    test_sentiment_twitter =
sentiment_twitter[o*subsetsize_twitter:][:subsetsize_twitter]
    sentiment_total_list.append(test_sentiment_twitter)
    print(sentiment_total_list)


# writing list with features and classified classes to csv
with open("classifier_classes.csv", "w") as g:
    writer = csv.writer(g)
    for row in sentiment_total_list:
        writer.writerow(sentiment_total_list)
```

8.14 Classification code 2 Multinomial Naïve Bayes - BP

```
import pandas as pd
import csv
import numpy as np
from sklearn.utils import shuffle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
from sklearn.model_selection import KFold

### For multin. Naive Bayes with scikit classes need to be integers
# positive = 1, negative = 2, neutral = 0


### importing excel
excelfile = "Manuallabel23.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')
print(dataframe1.columns)

# shuffeling data
dataframe1 = shuffle(dataframe1)

## reset index
dataframe1 = dataframe1.reset_index(drop=True)

# defining data
data = dataframe1["Tweet"]
classes = dataframe1["Sentiment"]

# Vectorization to turn covert text into a matrix of token counts and all
the words that appear
vector = CountVectorizer()

# split in K-fold parts
kf = KFold(n_splits=10)


classresults = []
predictresults = []
testclasses = []

for train, test in kf.split(data):
    train_data = np.array(data)[train]
    test_data = np.array(data)[test]
    train_classes = np.array(classes)[train]
    test_classes = np.array(classes)[test]

    # learn the vector
    vector.fit(train_data)

    # transform training data into ducement-term matrix
    train_data_matrix = vector.transform(train_data)
```

```
    # transform testing data into document-term matrix
    test_data_matrix = vector.transform(test_data)

    # classifier
    nb = MultinomialNB(alpha=1, fit_prior=True, class_prior=None)

    # train the model
    nb.fit(train_data_matrix, train_classes)

    # prediction classes for train_data_matrix
    pred_classes_1 = nb.predict(test_data_matrix)

    #  calculate accuracy
    classresults.append((metrics.accuracy_score(test_classes,
pred_classes_1)*100))

    # appending results for confusion matrix
    predictresults.append(pred_classes_1)

    # appending test classes for confusion matrix
    testclasses.append(test_classes)

print(classresults)

# printing average accuracy
print(sum(classresults)/float(len(classresults)))


# writing classified results to csv
with open("predicted_classes.csv", "w") as g:
    writer = csv.writer(g)
    writer.writerow(predictresults)


# writing classes to csv
with open("classifier_classes.csv", "w") as f:
    writer = csv.writer(f)
    writer.writerow(testclasses)


print(testclasses)


print(predictresults)
```

8.15 Classification code 3 Fuzzy Fingerprint - BP

```
import pandas as pd
from sklearn.utils import shuffle
import nltk
from sklearn.cross_validation import KFold
from sklearn.model_selection import KFold as KFold2
import math
import numpy as np
from operator import itemgetter
from pandas_ml import ConfusionMatrix

### importing excel
excelfile = "BP training_test set.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')
dataframe1['tweet_id'] = dataframe1.index

print(dataframe1.columns)
#column sentiment
# 0= neutral, 1=positive, 2=negative

# dividing data into classes
classneu = dataframe1.loc[dataframe1['Sentiment'] == 0]
classpos = dataframe1.loc[dataframe1['Sentiment'] == 1]
classneg = dataframe1.loc[dataframe1['Sentiment'] == 2]


# shuffeling data
classneu = shuffle(classneu)
classpos = shuffle(classpos)
classneg = shuffle(classneg)


## reset index for classes
classneu = classneu.reset_index(drop=True)
classpos = classpos.reset_index(drop=True)
classneg = classneg.reset_index(drop=True)


# split in K-fold parts
kf2 = KFold2(n_splits=10, shuffle=True)

#  initialize top-k
topk = 10

# initialize starting fold
fold = 0

# check for the correct shape and size of the datasets
print(classneu.shape)
print(classpos.shape)
print(classneg.shape)
```

```
experiment_input = [classpos, classneg, classneu]

# Create fold with training and test sets that is folded 10 times for cross
validation
folds = [[(x,y) for x,y in KFold(len(class_label), n_folds=10,
shuffle=True)] for class_label in experiment_input]


# initiate loop that iterates over the different folds per class
for f in range(0,len(folds[0])): # loops from 0 to 10 folds
    print(f)
    class_number = 0
    fingerprints = {}
    # mapping that categorizes the integers as classes
    mapping = {0: 'pos', 1:'neg', 2:'neu'}

    # loop that inputs a class from the input categories
    for class_label in experiment_input:

        train_index_pos = folds[0][f][0]
        train_index_neg = folds[1][f][0]
        train_index_neu = folds[2][f][0]

        train_pos = classpos.ix[train_index_pos]
        train_neg = classneg.ix[train_index_neg]
        train_neu = classneu.ix[train_index_neu]

        unique_words_pos = list(train_pos['Tweet'].str.split(' ',
expand=True).stack().unique())
        unique_words_neg = list(train_neg['Tweet'].str.split(' ',
expand=True).stack().unique())
        unique_words_neu = list(train_neu['Tweet'].str.split(' ',
expand=True).stack().unique())
        unique_words_all_classes = unique_words_pos + unique_words_neg +
unique_words_neu

        # Make training data
        train_index = folds[class_number][f][0]
        train = class_label.ix[train_index]


        # Put extracted words in dataframe tweet
        all_words = list(train['Tweet'].str.split(' ',
expand=True).stack())
        # convert all to lower cases
        all_words = [x.lower() for x in all_words]
        print(all_words)

        # check for the words
        print(len(all_words))

        # Make top-k frequency list
        def get_word_features(wordlist):
            wordlist = nltk.FreqDist(wordlist)
            wordlist_sorted = sorted(wordlist, key=lambda x:
wordlist[x.lower()], reverse=True)[:topk]
            return wordlist_sorted

        top_k_class = get_word_features(all_words)
        print(top_k_class)
```

```
        # make all and unique words for counting occurences and making ICF
scores
        all_words_class = list(train['Tweet'].str.split(' ',
expand=True).stack())
        top_k_words_class = top_k_class
        df_unique_words_class = pd.DataFrame(top_k_words_class,
columns=['unique_words'])

        # create counter that counts the appearances of words in each class
        df_unique_words = pd.DataFrame({'unique_words':
unique_words_all_classes})
        classcounts = pd.Series(np.concatenate([x.split() for x in
df_unique_words.unique_words])).value_counts()
        classcounts = pd.DataFrame({'unique_words': classcounts.index,
'class_counts': classcounts.values})

        class_counts = pd.merge(df_unique_words_class, classcounts,
on='unique_words')

        # convert al 3 scores to 2.99 otherwise the ICF score will be 0..
        class_counts['class_counts'].replace(3, 2.99, inplace=True)

        # make word count in training data per class and merge to dataframe
        word_count = pd.DataFrame({'words': all_words_class})
        wordcounts = pd.Series(np.concatenate([x.split() for x in
word_count.words])).value_counts()
        wordcounts = pd.DataFrame({'unique_words': wordcounts.index,
'word_counts': wordcounts.values})
        print(wordcounts)

        ICF = pd.merge(class_counts, wordcounts, on='unique_words')


        # making total doc value (3 as fingerprint library is 3)
        ICF["total_doc"] = 3

        # column with log(3/class counts)
        ICF["log_scores"] = np.log10((ICF["total_doc"] /
ICF["class_counts"]))

        # create ICF scores
        ICF["ICF"] = ICF["log_scores"] * ICF["word_counts"]
        print(ICF)

        # sort frame by ICf scores (high-low)
        ICF = ICF.sort_values('ICF', ascending=False)

        # take top-k words based on ICF and delete all unneccesary
information
        ICF_topk = ICF
        ICF_topk_words = ICF_topk['unique_words']
        ICF_topk_words = ICF_topk_words.to_frame().reset_index()
        ICF_topk_words = ICF_topk_words.drop('index', 1)
        print(ICF_topk_words)

        # make fingerprint per class
        fingerprint = ICF_topk_words.rename(columns={'unique_words':
'word'})
```

```
        # assign ranks for the top-k words per fingerprint
        fingerprint['Rank'] = range(1, fingerprint.shape[0] + 1)

        # assign mu1 and mu 2 values per word for the fingerprint
        fingerprint['mu1'] = fingerprint['Rank'].apply(lambda x: (topk - x)
/ topk)
        fingerprint['mu2'] = fingerprint['Rank'].apply(lambda x: (1 - ((1 -
0.2) * (x / (0.2 * topk)))) if x <= 0.2 * topk else (0.2 * (1 - ((x - 0.2 *
topk) / (topk - (0.2 * topk))))))

        # making totals for the T2S2 score later on
        mu1sum = fingerprint['mu1'].sum()
        mu2sum = fingerprint['mu2'].sum()

        # apply previous set mapping on classes in dictionary
        fingerprints[mapping[class_number]] = fingerprint

        # class +1 for iterations over classes
        class_number += 1

        # reset classes for next loop cycle
    class_number = 0

    # Classification
    print('step 2')

    # storing results per class in dictionary
    results = {}

    # Assign tweet id's per input and per class
    for class_label in experiment_input:
        test_index = folds[class_number][f][1]
        test = class_label.ix[test_index]

        # make separate words of tweets in test data
        test['Tweet'] = test['Tweet'].apply(lambda x: [a.lower() for a in
x.split()])

        tweet_ids = test['tweet_id'].tolist()
        test_results = pd.DataFrame({'tweet_id': tweet_ids})

        # Making loop that makes a copy of the test data and splits the
words
        class_number_2 = 0
        for fp in fingerprints:
            print(fp)
            test_class = test.copy(True)
            word_column = test_class.apply(lambda x: pd.Series(x['Tweet']),
axis=1).stack().reset_index(level=1, drop=True)
            word_column.name = 'word'

            # joining all separate words
            test_temp = test_class.join(word_column)

            # Merge words matching the fingerprint id's with the words
            test_temp = test_temp.merge(fingerprints[fp], on='word',
how='left')

            # Assigning mu values for the words that correspond to the
fingerpint words
            test_temp = test_temp.groupby('tweet_id').agg({'mu1':'sum',
```

```
'mu2':'sum'}).reset_index()
            test_temp.rename(columns={'mu1':'mu1total', 'mu2':'mu2total'},
inplace=True)

            # Summing the total of the mu values
            test_class = test_class.merge(test_temp, on='tweet_id',
how='left')
            test_class['mu1total'].fillna(0, inplace=True)
            test_class['mu2total'].fillna(0, inplace=True)

            # Calculating T2S2 scores based on the total mu values from the
fingerprint and the mu values of the parsed tweet
            test_class['T2S2mu1'+str(mapping[class_number_2])] =
test_class['mu1total'].apply(lambda x: x / mu1sum)
            test_class['T2S2mu2'+str(mapping[class_number_2])] =
test_class['mu2total'].apply(lambda x: x / mu2sum)
            print(test_class)

            # merging all separate variables together in one table
            test_r = test_class[['tweet_id',
'T2S2mu1'+str(mapping[class_number_2]),
'T2S2mu2'+str(mapping[class_number_2])]].copy(True)
            test_results = test_results.merge(test_r, on='tweet_id',
how='left')

            # number +1 to iterate over class
            class_number_2 += 1
        results[mapping[class_number]] = test_results

        # Check length to see if all goes well
        print(len(results))

        # number +1 to iterate over fold
        class_number += 1

    # appending the results of the folds to one total output
    for r in results:
        if f == 0:
            #first fold
            if r == 'pos':
                overall_pos_results = results[r]
            elif r == 'neg':
                overall_neg_results = results[r]
            elif r == 'neu':
                overall_neu_results = results[r]
        else:
            if r == 'pos':
                overall_pos_results =
overall_pos_results.append(results[r], ignore_index=True)
            elif r == 'neg':
                overall_neg_results =
overall_neg_results.append(results[r], ignore_index=True)
            elif r == 'neu':
                overall_neu_results =
overall_neu_results.append(results[r], ignore_index=True)


overall_pos_results.sort_values('tweet_id',inplace=True)
overall_neg_results.sort_values('tweet_id',inplace=True)
overall_neu_results.sort_values('tweet_id',inplace=True)
```

```
# check for the desired size of the output
print(overall_pos_results)
print(type(overall_pos_results))
# print(overall_neg_results.shape)
# print(overall_neu_results.shape)

# Store only mu1 and mu2 values in separate dataframes
# Pos Mu 1
overall_pos_results_mu1 = pd.DataFrame()
overall_pos_results_mu1["pos"]= overall_pos_results["T2S2mu1pos"]
overall_pos_results_mu1["neg"]= overall_pos_results["T2S2mu1neg"]
overall_pos_results_mu1["neu"]= overall_pos_results["T2S2mu1neu"]


# Pos Mu 2
overall_pos_results_mu2 = pd.DataFrame()
overall_pos_results_mu2["pos"]= overall_pos_results["T2S2mu2pos"]
overall_pos_results_mu2["neg"]= overall_pos_results["T2S2mu2neg"]
overall_pos_results_mu2["neu"]= overall_pos_results["T2S2mu2neu"]


# Neg Mu 1
overall_neg_results_mu1 = pd.DataFrame()
overall_neg_results_mu1["pos"]= overall_neg_results["T2S2mu1pos"]
overall_neg_results_mu1["neg"]= overall_neg_results["T2S2mu1neg"]
overall_neg_results_mu1["neu"]= overall_neg_results["T2S2mu1neu"]


# Neg Mu 2
overall_neg_results_mu2 = pd.DataFrame()
overall_neg_results_mu2["pos"]= overall_neg_results["T2S2mu2pos"]
overall_neg_results_mu2["neg"]= overall_neg_results["T2S2mu2neg"]
overall_neg_results_mu2["neu"]= overall_neg_results["T2S2mu2neu"]


# Neu Mu 1
overall_neu_results_mu1 = pd.DataFrame()
overall_neu_results_mu1["pos"]= overall_neu_results["T2S2mu1pos"]
overall_neu_results_mu1["neg"]= overall_neu_results["T2S2mu1neg"]
overall_neu_results_mu1["neu"]= overall_neu_results["T2S2mu1neu"]


# Neu Mu 2
overall_neu_results_mu2 = pd.DataFrame()
overall_neu_results_mu2["pos"]= overall_neu_results["T2S2mu2pos"]
overall_neu_results_mu2["neg"]= overall_neu_results["T2S2mu2neg"]
overall_neu_results_mu2["neu"]= overall_neu_results["T2S2mu2neu"]


# Pick highest T2S2 score that is calculated as prediction per mu function
overall_pos_results_mu1['prediction'] =
overall_pos_results_mu1.idxmax(axis=1)
overall_pos_results_mu2['prediction'] =
overall_pos_results_mu2.idxmax(axis=1)
overall_neg_results_mu1['prediction'] =
overall_neg_results_mu1.idxmax(axis=1)
overall_neg_results_mu2['prediction'] =
overall_neg_results_mu2.idxmax(axis=1)
overall_neu_results_mu1['prediction'] =
overall_neu_results_mu1.idxmax(axis=1)
overall_neu_results_mu2['prediction'] =
```

```
overall_neu_results_mu2.idxmax(axis=1)


print(overall_pos_results_mu1)
print(overall_pos_results_mu2)
print(overall_neg_results_mu1)
print(overall_neg_results_mu2)
print(overall_neu_results_mu1)
print(overall_neu_results_mu2)

# adding original classes for confusion matrix
overall_pos_results_mu1["original class"]= 'pos'
overall_pos_results_mu2["original class"]= 'pos'
overall_neg_results_mu1["original class"]= 'neg'
overall_neg_results_mu2["original class"]= 'neg'
overall_neu_results_mu1["original class"]= 'neu'
overall_neu_results_mu2["original class"]= 'neu'

# make confusion matrix per category
# mu 1
conf_mat_pos_mu1 = pd.crosstab(overall_pos_results_mu1["original class"],
overall_pos_results_mu1["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_pos_mu1)
conf_mat_neg_mu1 = pd.crosstab(overall_neg_results_mu1["original class"],
overall_neg_results_mu1["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_neg_mu1)
conf_mat_neu_mu1 = pd.crosstab(overall_neu_results_mu1["original class"],
overall_neu_results_mu1["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_neu_mu1)

print("")
print("")

# mu 2
conf_mat_pos_mu2 = pd.crosstab(overall_pos_results_mu2["original class"],
overall_pos_results_mu2["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_pos_mu2)
conf_mat_neg_mu2 = pd.crosstab(overall_neg_results_mu2["original class"],
overall_neg_results_mu2["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_neg_mu2)
conf_mat_neu_mu2 = pd.crosstab(overall_neu_results_mu2["original class"],
overall_neu_results_mu2["prediction"], rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(conf_mat_neu_mu2)
```

8.16 Improved pre-processing

```
import pandas as pd
from slangdict import slangdict
from emoticons import emoticons
from appos import appos
import string
import itertools
from autocorrect import spell
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import nltk

### importing excel
excelfile = "Manuallabel2.xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')

# check column names
print(dataframe1.columns)


### removing duplicate tweets
dataframe1.duplicated()
dataframe1.drop_duplicates(keep='first', inplace=True)


### removing Retweets
dataframe1 = dataframe1[dataframe1.Tweet.str.contains("RT") == False]


### removing \(UTF encoding errors)
# define pattern \S anything but a space, + match one or more
patternhtml = r'\\S+'


### removing \\ (UTF encoding errors)
# define pattern \S anything but a space, + match one or more
patternhtml = r'\\\S+'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patternhtml: ""},inplace=True, regex=True)


### conversion of ’ to '
# define pattern
patterncomma = r'’'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patterncomma: "'"},inplace=True, regex=True)


### hasthag removal
# define pattern
```

```
patternhash = r'#'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patternhash: ""},inplace=True, regex=True)


### removal of usernames
# define pattern
patternuser = r'(@\w+)'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patternuser: ""},inplace=True, regex=True)


#### removing capital letters
dataframe1['Tweet'] = dataframe1['Tweet'].str.lower()


### HTML REMOVING
# define pattern \S anything but a space, + match one or more
patternhtml = r'http\S+'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patternhtml: ""},inplace=True, regex=True)


# reset index
dataframe1 = dataframe1.reset_index(drop=True)


### replacing emoticons

# First separate any string that might be attached to the emoticons at
front or back
patternemoticon0 = r':'
patternemoticon1 = r'\('
patternemoticon2 = r'\)'
patternemoticon3 = r'\['
patternemoticon4 = r'\]'
patternemoticon5 = r'\{'
patternemoticon6 = r'\}'

# replacing regex functions based on column tweet
dataframe1.Tweet.replace({patternemoticon0: " :"},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon1: "( "},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon2: ") "},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon3: "[ "},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon4: "] "},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon5: "{ "},inplace=True, regex=True)
dataframe1.Tweet.replace({patternemoticon6: "} "},inplace=True, regex=True)

### replacing emoticons
dictionary0 = emoticons

# replacing slang words (working)
counter0 = list(range(0,18743))  ## place last index value here of datasets
last position
counter1 = range(0,100) ## value that makes it possible to iterate over all
dictionary values
counter2 = [counter1] ## make list out of values
for a in counter0:
```

```
    words0 = dataframe1['Tweet'][a].split()
    for b in counter2:
        emotifree = [dictionary0[b] if b in dictionary0 else b for b in
words0]
        emotifree = " ".join(emotifree)
        dataframe1.set_value(a, 'Tweet', emotifree)



# calling dictionary of apostroph contracted words
dictionary1 = appos

# replacing contraction words
counter3 = list(range(0,18743))  ## place last index value here of datasets
last position
counter4 = range(0,100) ## value that makes it possible to iterate over all
dictionary values
counter5 = [counter4] ## make list out of values
for c in counter3:
    words1 = dataframe1['Tweet'][c].split()
    for d in counter5:
        reformed = [dictionary1[d] if d in dictionary1 else d for d in
words1]
        reformed = " ".join(reformed)
        dataframe1.set_value(c, 'Tweet', reformed)
        counter2 = 0



### Removing punctuation
counter6 = list(range(0,18743))  ## place last index value here of datasets
last position
for e in counter6:
    for f in string.punctuation:
        punct = dataframe1['Tweet'][e].replace(f," ")
        dataframe1.set_value(e, 'Tweet', punct)



### cleaning van slang
dictionary1 = slangdict

# replacing slang words (working)
counter7 = list(range(0,18743))  ## place last index value here of datasets
last position
counter8 = range(0,5100) ## value that makes it possible to iterate over
all dictionary values
counter9 = [counter8] ## make list out of values
for g in counter7:
    words2 = dataframe1['Tweet'][g].split()
    for h in counter9:
        reformed2 = [dictionary1[h] if h in dictionary1 else h for h in
words2]
        reformed2 = " ".join(reformed2)
        dataframe1.set_value(g, 'Tweet', reformed2)



### standardizing words
counter10 = list(range(0,18743))  ## place last index value here
```

```
for i in counter10:
    words3 = dataframe1['Tweet'][i]
    words3 = ''.join(''.join(s)[:2] for _, s in itertools.groupby(words3))
    words3 = ''.join(words3)
    dataframe1.set_value(i, 'Tweet', words3)



### spelling correction
counter11 = list(range(0,18743))  ## place last index value here

for j in counter11:
    words4 = dataframe1['Tweet'][j].split()
    print(j)
    for k in range(len(words4)):
        correctword = spell(words4[k])
        if correctword != words4[k]:
            words4[k] = correctword
    words4 = " ".join(words4)
    dataframe1.set_value(j, 'Tweet', words4)



### negation handling

#list of tags that are adj, verb or noun etc.
""" adjective = ["JJ", "JJR", "JJS"]
verb = ["VB", "VBD", "VBG", "VBN", "VBP", "VBZ"]
noun = ["NN", "NNP", "NNS"]
adverb = ["RB", "RBR", "RBS"]
determiner = ["DT", "PDT", "WDT"] """

counter12 = list(range(0,18743))  ## place last index value here

for l in counter12:
    flag = False
    counter_words = 0
    words5 = dataframe1['Tweet'][l].split()
    print(words5)
    tagged_words5 = tagged = nltk.pos_tag(words5)
    limit = len(words5)-1

    for w in tagged_words5[:limit]:
        if w[0] == "no" or w[0] == "not" or w[0] == "none" or w[0] ==
"neither" or w[0] == "never" or w[0] == "hardly" or w[0] == "scarcely" or
w[0] == "barely" or w[0] == "cannot":
            flag = not flag
            if flag == True:
                if w[1] == "JJ" or w[1] == "JJR" or w[1] == "JJS" or w[1]
== "VB" or w[1] == "VBD" or w[1] == "VBG" or w[1] == "VBN" or w[1] == "VBP"
or w[1] == "VBZ" or w[1] == "NN" or w[1] == "NNP" or w[1] == "NNS" or w[1]
== "RB" or w[1] == "RBR" or w[1] == "RBS" or w[1] == "DT" or w[1] == "PDT"
or w[1] == "WDT":
                    words5[counter_words + 1] = "not_" +
words5[counter_words + 1]
        counter_words += 1
        flag = False

    words5 = " ".join(words5)
    dataframe1.set_value(l, 'Tweet', words5)
```

```
### filter out stopwords
counter13 = list(range(0,18743))  ## place last index value here of
datasets last position
counter14 = range(0,155) ## value that makes it possible to iterate over
all dictionary values
counter15 = [counter14] ## make list out of values

stopwords_list = set(stopwords.words('english'))
# print(stopwords_list)

for m in counter13:
    words6 = dataframe1['Tweet'][m].split()
    for n in counter15:
        words6 = [n for n in words6 if n not in stopwords_list]
        words6 = " ".join(words6)
        dataframe1.set_value(m, 'Tweet', words6)


#### removing capital letters
dataframe1['Tweet'] = dataframe1['Tweet'].str.lower()


### Lemmitization
lemmatizer = WordNetLemmatizer()
counter16 = list(range(0,18743))  ## place last index value here

for o in counter16:
    words7 = dataframe1['Tweet'][o].split()
    for p in range(len(words7)):
        lemmaword = lemmatizer.lemmatize(words7[p])
        if lemmaword != words7[p]:
            words7[p] = lemmaword
    words7 = " ".join(words7)
    dataframe1.set_value(o, 'Tweet', words7)


# writing dataframe to excel
dataframe1 = dataframe1.applymap(lambda x:
x.encode('unicode_escape').decode('utf-8') if isinstance(x, str) else x)
dataframe1.to_excel("test.xlsx", encoding = "utf-8")
```

8.17 Classification code IDF Multinomial Naïve Bayes

Note that this code works fine but is far from optimal due to the time constraints near the end of the thesis.

```
import pandas as pd
from sklearn.utils import shuffle
import nltk
from sklearn.cross_validation import KFold
from sklearn.model_selection import KFold as KFold2
import math
import numpy as np
from operator import itemgetter


### importing excel
excelfile = "pre-processing _cycle2_(labels).xlsx"

# loading spreadsheat
sheet1 = pd.ExcelFile(excelfile, sep='\s*,\s*', header=0, encoding = "utf-
8")
print(sheet1.sheet_names)

# Load sheet into DataFrame
dataframe1 = sheet1.parse('Sheet1')
dataframe1['tweet_id'] = dataframe1.index

print(dataframe1.columns)
#column sentiment
# 0= neutral, 1=positive, 2=negative

# dividing data into classes
classneu = dataframe1.loc[dataframe1['Sentiment'] == 0]
classpos = dataframe1.loc[dataframe1['Sentiment'] == 1]
classneg = dataframe1.loc[dataframe1['Sentiment'] == 2]


# shuffeling data
classneu = shuffle(classneu)
classpos = shuffle(classpos)
classneg = shuffle(classneg)


## reset index for classes
classneu = classneu.reset_index(drop=True)
classpos = classpos.reset_index(drop=True)
classneg = classneg.reset_index(drop=True)

print(classpos.shape[0])

# split in K-fold parts
kf2 = KFold2(n_splits=10, shuffle=True)

# initialize starting fold
fold = 0

# Create fold with training and test sets that is folded 10 times for cross
validation
folds = [[(x,y) for x,y in KFold(len(class_label), n_folds=10,
shuffle=True)] for class_label in experiment_input]
```

```
results_folds_pos = []
results_folds_neg = []
results_folds_neu = []

# initiate loop that iterates over the different folds per class
for f in range(0,len(folds[0])): # loops from 0 to 10 folds
    print(f)
    class_number = 0

    train_index_pos = folds[0][f][0]
    train_index_neg = folds[1][f][0]
    train_index_neu = folds[2][f][0]

    train_pos = classpos.ix[train_index_pos]
    train_neg = classneg.ix[train_index_neg]
    train_neu = classneu.ix[train_index_neu]

    # counting total words in training data
    train_all_words1 = list(train_pos['Tweet'].str.split(' ',
expand=True).stack())
    train_all_words2 = list(train_neg['Tweet'].str.split(' ',
expand=True).stack())
    train_all_words3 = list(train_neu['Tweet'].str.split(' ',
expand=True).stack())
    train_all_words_total = train_all_words1 + train_all_words2 +
train_all_words3
    total_words_train = len(train_all_words1) + len(train_all_words2) +
len(train_all_words3)

    unique_words_pos = list(train_pos['Tweet'].str.split(' ',
expand=True).stack().unique())
    unique_words_neg = list(train_neg['Tweet'].str.split(' ',
expand=True).stack().unique())
    unique_words_neu = list(train_neu['Tweet'].str.split(' ',
expand=True).stack().unique())
    unique_words_all_classes = unique_words_pos + unique_words_neg +
unique_words_neu

    # counting total unique words in training data
    # convert all to lower cases
    lower_words1 = [x.lower() for x in train_all_words1]
    lower_words2 = [x.lower() for x in train_all_words2]
    lower_words3 = [x.lower() for x in train_all_words3]
    all_lower_words = lower_words1 + lower_words2 +lower_words3


    # Make frequency list
    def get_word_features(wordlist):
        wordlist = nltk.FreqDist(wordlist)
        wordlist_sorted = sorted(wordlist, key=lambda x:
wordlist[x.lower()], reverse=True)
        return wordlist_sorted

    all_unique_words = get_word_features(all_lower_words)
    total_all_unique_words = len(all_unique_words)


    # compute prior probabilities for training data
```

```
    priorpos = train_pos.shape[0] / (train_pos.shape[0] +
train_neg.shape[0] + train_neu.shape[0])
    priorneg = train_neg.shape[0] / (train_pos.shape[0] +
train_neg.shape[0] + train_neu.shape[0])
    priorneu = train_neu.shape[0] / (train_pos.shape[0] +
train_neg.shape[0] + train_neu.shape[0])


    # make all and unique words for counting occurences and making ICF
scores
    df_unique_words_class = pd.DataFrame(all_unique_words,
columns=['unique_words'])


    # create counter that counts the appearances of words in each class
    df_unique_words = pd.DataFrame({'unique_words':
unique_words_all_classes})

    classcounts = pd.Series(np.concatenate([x.split() for x in
df_unique_words.unique_words])).value_counts()
    classcounts = pd.DataFrame({'unique_words': classcounts.index,
'class_counts': classcounts.values})
    class_counts = pd.merge(df_unique_words_class, classcounts,
on='unique_words')


    # make word count in training data per class and merge to dataframe
    word_count = pd.DataFrame({'words': all_lower_words})
    wordcounts = pd.Series(np.concatenate([x.split() for x in
word_count.words])).value_counts()
    wordcounts = pd.DataFrame({'unique_words': wordcounts.index,
'word_counts_all': wordcounts.values})

    class_word_count = pd.merge(class_counts, wordcounts,
on='unique_words')


    # column with penalties
    class_word_count['penalty'] = [1 if x == 1 else 2/3 if x == 2 else 1/3
for x in class_word_count['class_counts']]

    # make word counts per sentiment class
    # positive
    word_count_pos = pd.DataFrame({'words': lower_words1})
    wordcounts_pos = pd.Series(np.concatenate([x.split() for x in
word_count_pos.words])).value_counts()
    wordcounts_pos = pd.DataFrame({'unique_words': wordcounts_pos.index,
'word_counts_pos': wordcounts_pos.values})
    class_word_count = pd.merge(class_word_count, wordcounts_pos,
how='left', on='unique_words')
    class_word_count.word_counts_pos.fillna(0, inplace=True)

    # negative
    word_count_neg = pd.DataFrame({'words': lower_words2})
    wordcounts_neg = pd.Series(np.concatenate([x.split() for x in
word_count_neg.words])).value_counts()
    wordcounts_neg = pd.DataFrame({'unique_words': wordcounts_neg.index,
'word_counts_neg': wordcounts_neg.values})
    class_word_count = pd.merge(class_word_count, wordcounts_neg,
how='left', on='unique_words')
    class_word_count.word_counts_neg.fillna(0, inplace=True)
```

```
    # Neutral
    word_count_neu = pd.DataFrame({'words': lower_words3})
    wordcounts_neu = pd.Series(np.concatenate([x.split() for x in
word_count_neu.words])).value_counts()
    wordcounts_neu = pd.DataFrame({'unique_words': wordcounts_neu.index,
'word_counts_neu': wordcounts_neu.values})
    class_word_count = pd.merge(class_word_count, wordcounts_neu,
how='left', on='unique_words')
    class_word_count.word_counts_neu.fillna(0, inplace=True)

    ## compute conditional probabilites for each word per class
    cond_prob = pd.DataFrame({'condprob_words': all_unique_words})

    # pos
    condprob_pos = []
    for i in range(0, class_word_count.shape[0]):
        condprob_pos.append((class_word_count["word_counts_pos"][i] + 1) /
(len(train_all_words1) + total_all_unique_words))

    pos = pd.Series(condprob_pos)
    cond_prob['condprob_pos'] = pos.values

    # neg
    condprob_neg = []
    for i in range(0, class_word_count.shape[0]):
        condprob_neg.append((class_word_count["word_counts_neg"][i] + 1) /
(len(train_all_words2) + total_all_unique_words))

    neg = pd.Series(condprob_neg)
    cond_prob['condprob_neg'] = neg.values

    # neu
    condprob_neu = []
    for i in range(0, class_word_count.shape[0]):
        condprob_neu.append((class_word_count["word_counts_neu"][i] + 1) /
(len(train_all_words3) + total_all_unique_words))

    neu = pd.Series(condprob_neu)
    cond_prob['condprob_neu'] = neu.values


    ## penalties toevoegen op cond probabilities voor IDF classes
    cond_prob["penalty"] = class_word_count["penalty"]

    cond_prob["condprob_pos_penalized"] = cond_prob.penalty *
cond_prob.condprob_pos
    cond_prob["condprob_neg_penalized"] = cond_prob.penalty *
cond_prob.condprob_neg
    cond_prob["condprob_neu_penalized"] = cond_prob.penalty *
cond_prob.condprob_neu



    #### Parsing tweets and calculating probabilities per class

    # first make test data available
    test_index_pos = folds[0][f][1]
    test_index_neg = folds[1][f][1]
    test_index_neu = folds[2][f][1]
```

```
    test_pos = classpos.ix[test_index_pos]
    test_pos = test_pos.reset_index(drop=True)
    test_neg = classneg.ix[test_index_neg]
    test_neg = test_neg.reset_index(drop=True)
    test_neu = classneu.ix[test_index_neu]
    test_neu = test_neu.reset_index(drop=True)


    # positive test tweets
    counter1 = range(0,test_pos.shape[0])

    results_pos_tweets_pos = []
    results_pos_pos = []


    for i in counter1:
        words = test_pos['Tweet'][i].split()
        for j in range(0, len(words)):
            try:
                pos_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_pos_penalized'].iloc[0]
                results_pos_pos.append(pos_prob)
            except IndexError:
                results_pos_pos.append((1 / (len(train_all_words1) +
total_all_unique_words)))

        pos_sum_pos = sum(results_pos_pos)
        results_pos_tweets_pos.append(pos_sum_pos)


    # negative test tweets
    counter2 = range(0,test_pos.shape[0])

    results_neg_tweets_pos = []
    results_neg_pos = []


    for h in counter2:
        words = test_pos['Tweet'][h].split()
        for j in range(0, len(words)):
            try:
                neg_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neg_penalized'].iloc[0]
                results_neg_pos.append(neg_prob)
            except IndexError:
                results_neg_pos.append((1 / (len(train_all_words2) +
total_all_unique_words)))

        neg_sum_pos = sum(results_neg_pos)
        results_neg_tweets_pos.append(neg_sum_pos)


    # neutral test tweets
    counter3 = range(0,test_pos.shape[0])

    results_neu_tweets_pos = []
    results_neu_pos = []


    for h in counter3:
        words = test_pos['Tweet'][h].split()
```

```
        for j in range(0, len(words)):
            try:
                neu_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neu_penalized'].iloc[0]
                results_neu_pos.append(neu_prob)
            except IndexError:
                results_neu_pos.append((1 / (len(train_all_words3) +
total_all_unique_words)))

        neu_sum_pos = sum(results_neu_pos)
        results_neu_tweets_pos.append(neu_sum_pos)


    result_fold_pos = pd.DataFrame(data={'pos': results_pos_tweets_pos,
'neg': results_neg_tweets_pos, 'neu': results_neu_tweets_pos})
    results_folds_pos.append(result_fold_pos)
    result_total_pos = pd.concat(results_folds_pos)
    result_total_pos.reset_index(inplace=True)
    del result_total_pos['index']
    result_total_pos['prediction'] = result_total_pos.idxmax(axis=1)
    print(result_total_pos)

    results_pos_pos = []
    results_neg_pos = []
    results_neu_pos = []


    # negative test tweets
    counter1 = range(0,test_neg.shape[0])

    results_pos_tweets_neg = []
    results_pos_neg = []


    for i in counter1:
        words = test_neg['Tweet'][i].split()
        for j in range(0, len(words)):
            try:
                neg_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_pos_penalized'].iloc[0]
                results_pos_neg.append(neg_prob)
            except IndexError:
                results_pos_neg.append((1 / (len(train_all_words1) +
total_all_unique_words)))

        pos_sum_neg = sum(results_pos_neg)
        results_pos_tweets_neg.append(pos_sum_neg)

    print(len(results_pos_tweets_neg))

    # negative test tweets
    counter2 = range(0,test_neg.shape[0])

    results_neg_tweets_neg = []
    results_neg_neg = []


    for h in counter2:
        words = test_neg['Tweet'][h].split()
        for j in range(0, len(words)):
            try:
```

```
                neg_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neg_penalized'].iloc[0]
                results_neg_neg.append(neg_prob)
            except IndexError:
                results_neg_neg.append((1 / (len(train_all_words2) +
total_all_unique_words)))

        neg_sum_neg = sum(results_neg_neg)
        results_neg_tweets_neg.append(neg_sum_neg)

    print(len(results_neg_tweets_neg))

    # neutral test tweets
    counter3 = range(0,test_neg.shape[0])

    results_neu_tweets_neg = []
    results_neu_neg = []


    for h in counter3:
        words = test_neg['Tweet'][h].split()
        for j in range(0, len(words)):
            try:
                neu_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neu_penalized'].iloc[0]
                results_neu_neg.append(neu_prob)
            except IndexError:
                results_neu_neg.append((1 / (len(train_all_words3) +
total_all_unique_words)))

        neu_sum_neg = sum(results_neu_neg)
        results_neu_tweets_neg.append(neu_sum_neg)

    print(len(results_neu_tweets_neg))

    result_fold_neg = pd.DataFrame(data={'pos': results_pos_tweets_neg,
'neg': results_neg_tweets_neg, 'neu': results_neu_tweets_neg})
    results_folds_neg.append(result_fold_neg)
    result_total_neg = pd.concat(results_folds_neg)
    result_total_neg.reset_index(inplace=True)
    del result_total_neg['index']
    result_total_neg['prediction'] = result_total_neg.idxmax(axis=1)
    print(result_total_neg)

    results_pos_neg = []
    results_neg_neg = []
    results_neu_neg = []



    # neutral test tweets
    counter1 = range(0,test_neu.shape[0])

    results_pos_tweets_neu = []
    results_pos_neu = []


    for i in counter1:
        words = test_neu['Tweet'][i].split()
        for j in range(0, len(words)):
            try:
```

```
                neu_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_pos_penalized'].iloc[0]
                results_pos_neu.append(neu_prob)
            except IndexError:
                results_pos_neu.append((1 / (len(train_all_words1) +
total_all_unique_words)))

        pos_sum_neu = sum(results_pos_neu)
        results_pos_tweets_neu.append(pos_sum_neu)

    print(len(results_pos_tweets_neu))


    # negative test tweets
    counter2 = range(0,test_neu.shape[0])

    results_neg_tweets_neu = []
    results_neg_neu = []


    for h in counter2:
        words = test_neu['Tweet'][h].split()
        for j in range(0, len(words)):
            try:
                neu_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neg_penalized'].iloc[0]
                results_neg_neu.append(neu_prob)
            except IndexError:
                results_neg_neu.append((1 / (len(train_all_words2) +
total_all_unique_words)))

        neg_sum_neu = sum(results_neg_neu)
        results_neg_tweets_neu.append(neg_sum_neu)

    print(len(results_neg_tweets_neu))



    # neutral test tweets
    counter3 = range(0,test_neu.shape[0])

    results_neu_tweets_neu = []
    results_neu_neu = []


    for h in counter3:
        words = test_neu['Tweet'][h].split()
        for j in range(0, len(words)):
            try:
                neu_prob = cond_prob.loc[cond_prob['condprob_words'] ==
words[j], 'condprob_neu_penalized'].iloc[0]
                results_neu_neu.append(neu_prob)
            except IndexError:
                results_neu_neu.append((1 / (len(train_all_words3) +
total_all_unique_words)))

        neu_sum_neu = sum(results_neu_neu)
        results_neu_tweets_neu.append(neu_sum_neu)

    print(len(results_neu_tweets_neu))
```

```
    result_fold_neu = pd.DataFrame(data={'pos': results_pos_tweets_neu,
'neg': results_neg_tweets_neu, 'neu': results_neu_tweets_neu})
    results_folds_neu.append(result_fold_neu)
    result_total_neu = pd.concat(results_folds_neu)
    result_total_neu.reset_index(inplace=True)
    del result_total_neu['index']
    result_total_neu['prediction'] = result_total_neu.idxmax(axis=1)
    print(result_total_neu)

    results_pos_neg = []
    results_neg_neg = []
    results_neu_neg = []

writer = pd.ExcelWriter('output.xlsx')
result_total_pos.to_excel(writer,'positivbe')
result_total_neg.to_excel(writer,'negative')
result_total_neu.to_excel(writer,'neutral')
writer.save()
```