THE UNIVERSITY OF THE WITWATERSRAND,
JOHANNESBURG

Using Big Data for corporate brand analysis on the internet

Author:
Mike NKONGOLO - 1171635

A dissertation submitted in fulfillment of the requirements for the degree of Masters of Science

in the

School of Computer Science and Applied Mathematics

January 27, 2020

Declaration of Authorship

I, Wa Nkongolo Mike NKONGOLO - 1171635, hereby acknowledge that this dissertation "Using Big Data for corporate brand analysis on the internet" is my own, except where I indicated otherwise.

Signed: **W.N.M.N**

Date: January 27, 2020

"La patience est la longueur du temps."

Jean De La Fontein

THE UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Abstract

School of Computer Science and Applied Mathematics

Masters of Science

Using Big Data for corporate brand analysis on the internet

by Mike NKONGOLO - 1171635

From a marketing perspective, a company's reputation is a valuable, intangible asset. This reputation greatly influences the company's image. As a result, consumers choose companies with a positive reputation and are willing to pay more for their products or services. A positive corporate reputation can create a competitive advantage and barriers to entry for competitors (Zhang et al., 2019). Improving a company's image creates a new interest in Reputation Analysis. However, Reputation Analysis systems that focus on revealing the company's image on the internet are not effective in solving various problems, such as automatic and real-time Data Collection, Feature Extraction, classification, and Visualization. Hence, this work investigates the use of Machine Learning (ML) and Natural Language Processing (NLP) to solve these types of problems. Artificial Neural Networks (ANN), Sentistrength, and Bag-Of-Words (BOW) are introduced as classifiers-the Accuracy, Precision, and Empirical error metrics have been used to implement and measure the framework performance as in Jadav and Vaghela (2016). In general, the main difficulties in using the Feature Extraction and classification approach for Reputation Analysis are to minimize False positives (FP) and negatives (FN) and to maximize Accuracy (Rehman et al., 2019).

This work describes a Brand/ Reputation Analysis framework that uses sentiment contexts (retrieved from the web) to perform automated Reputation Analysis. The framework is in four stages. The first performed web crawling based on a query which is specified by the user. The second locates relevant information within textual data using a Named Entity Recognition (NER). The third records relevant information in a database for Feature Extraction and classification. Lastly, the framework was used for Reputation Analysis. The datasets for training were from: WITS marketing team, Sentistrength lexicon, and the Clueweb09. In testing the computational framework, ANN and Sentistrength achieved competitive results comparing to the works conducted by Shukri et al. (2015), Jadav and Vaghela (2016), and Rasool et al. (2019). The results revealed that ANN achieved more than 90% Accuracy in demarcating positive from negative comments of textual data. Particularly, the project does sentiment analysis (SA) on Wits online content. As such, this research investigates using sentiment analysis on online content from a University.

Keywords: Sentiment Analysis, Web Crawling, Feature Extraction, Reputation Analysis, Big Data, Data Analytics, Applied Computing.

Résumé

D'un point de vue du *marketing*, la réputation d'une entreprise est un ensemble immatériel précieux. Cette réputation influence grandement l'image de l'entreprise. En conséquence, les consommateurs choisissent des entreprises ayant une réputation positive et sont disposés à payer plus pour leurs produits ou services. Une réputation d'entreprise positive peut créer un avantage concurrentiel et des barrières à l'entrée pour les concurrents. L'amélioration de l'image d'une entreprise crée un nouvel intérêt pour l'analyse de réputation. Cependant, les systèmes d'analyse de réputation qui se concentrent sur la révélation de l'image de l'entreprise sur internet ne sont pas efficaces pour résoudre divers problèmes, tels que la Collection des données automatique et en temps réel, l'extraction de fonctionnalités, la classification et la visualisation. Par conséquent, ce travail étudie l'utilisation du Machine Learning (ML) et du Natural Language Processing (NLP) pour résoudre ces types de problèmes.

Des algorithmes puissant comme par exemple, les réseaux de neurones artificiels (ANN), Sentistrength et Bag-Of-Words (BOW) sont introduits comme classificateurs et les métriques d'erreur empirique ou d'exactitude voir aussi de précision ont été utilisées pour implémenter et mesurer les performances. En général, les principales difficultés rencontrées dans l'utilisation de l'approche d'extraction et de classification des fonctionnalités pour l'analyse de réputation sont de minimiser les faux positifs (FP) et les négatifs (FN) et de maximiser la précision.

Ce travail décrit ainsi un system d'analyse qui utilise des dectection des sentiments (récupérés textuellement sur le Web) pour effectuer une analyse de réputation automatisée. Le system est en quatre étapes. La première effectue une analyse Web en fonction d'une requête spécifiée par l'utilisateur. La seconde localise les informations pertinentes dans les données textuelles en utilisant une reconnaissance d'entité nommée (NER). La troisième enregistre les informations pertinentes dans une base de données pour l'extraction et la classification des entités. Enfin, le system a été utilisé pour l'analyse de la réputation.

Les ensembles de données pour l'entraînement algorithmique provenaient de: l'équipe de marketing WITS, lexique Sentistrength et le Clueweb09. En testant le cadre de calcul, ANN et Sentistrength ont obtenu des résultats compétitifs. Les résultats ont révélé que l'ANN a atteint une précision de plus de 90% dans la démarcation positive des commentaires négatifs des données textuelles. En particulier, et en tant que telle, cette recherche examine l'utilisation de l'analyse des sentiments sur le contenu en ligne d'une université.

Mots-clés: Analyse du sentiment, Analyse Web, Extraction de fonctionnalités, Analyse de la réputation, Big Data.

Acknowledgements

I would like to acknowledge the following people for their contribution:

1. My father, Professor Nkongolo Mukendi Jean-Paul for the financial support without which I could not complete this study;
2. My mother Melanie Masengu Ngandu and my family members;
3. The anonymous examiners for the comments which improved the quality and clarity of this dissertation.

Contents

List of Figures

List of Tables

List of Abbreviations

API	**A**pplication **p**rogramming **i**nterface
BOW	**B**ag **o**f **w**ords
MT	**M**achine **t**ranslation
FE	**F**eature **e**xtraction
TTS	**T**ext **t**o **s**peech
SD	**S**entiment **d**etection
TM	**T**opic **m**odeling
DE	**D**ata **e**xtraction
CGM	**C**onsumer **g**enerated **m**edia
GBS	**G**raph **b**ased **s**entiment
NER	**N**amed **e**ntity **r**ecognition
NLP	**N**atural **l**anguage **p**rocessing
POS	**P**art **o**f **s**peech **t**agger
RSS	**R**ich **s**ite **s**ummary
ANN	**A**rtificial **n**eural **n**etworks
AI	**A**rtificial **i**ntelligence
IR	**I**nformation **r**etrieval
SASA	**S**entiment **a**nalyser
GUI	**G**raphical **u**ser **i**nterface
HTML	**H**yper **t**ext **m**arkup **l**anguage
ETL	**E**xtract **t**ransform **l**oad
URL	**U**niform **r**esource **l**ocator
WWW	**W**orld **w**ide **w**eb
SVM	**S**upport **v**ector **m**achines
D	**D**ictionary
IMDB	**I**nternet **m**ovie **d**atabase
ODP	**O**pen **d**irectory **p**roject
SQL	**S**tructured **q**uery **l**anguage
TP	**T**rue **p**ositive
TN	**T**rue **n**egative
FP	**F**alse **p**ositive
FN	**F**alse **n**egative
ML	**M**achine **l**earning
ISP	**I**nternet **s**ervice **p**rovider
ORES	**O**rganisation **r**eputation **e**valuation **s**cheme
FIFO	**F**irst **i**n **f**irst **o**ut
CRA	**C**orporate **r**eputation **a**nalysis
IEEE	**I**nstitue **o**f **e**lectrical **a**nd **e**lectronics **e**ngineers
IP	**I**nternet **p**rotocol
TCP	**T**ransmission **c**ontrol **p**rotocol
OSI	**O**pen **s**ystems **i**nterconnection
DBMS	**D**ata **b**ase **m**anagement **s**ystem
DB	**D**ata **b**ase

DML	**D**ata **m**anipulation **l**anguage
DDL	**D**ata **d**efinition **l**anguage
DQL	**D**ata **q**uery **l**anguage
DDL	**D**ata **d**efinition **l**anguage
SA	**S**entiment **a**nalysis
RA	**R**eputation **a**nalysis
IS	**I**nformation **s**ystem
LR	**L**inear **r**egression
NB	**N**aïve **b**ayes
RBF	**R**adial **b**asis **f**unction
ORS	**O**verall **r**eputation **s**core
PCA	**P**rincipal **c**omponent **a**nalysis
LDA-RF	**L**inear **r**andom **d**iscriminant **a**nalysis **f**orest
IOT	**I**nternet **o**f **t**hings
SC	**S**pectral **c**lustering
SEO	**S**earch **e**ngine **o**ptimisation

List of Symbols

α	learning rate
Φ	the parameter of the encoder (recognition model)
Ψ	the parameter of the decoder (generative model)
s_i	set of neutral sentences
s_i^+	set of positive sentences
s_i^-	set of negative sentences
S	set of sentences
y	output, target values
$(x^{(i)}, y^{(i)})$	output of our hypothesis on input x (same dimension as y)
a_i^l	activation/ output of unit i in layer l of the network ($a_i^{(l)} = x_i$)
$h_{w,b}^{(x)}$	the i^{th} training example
SS_p	the average of sentimentality score of sentences
p	a web page
SS_p	$SS_p = \frac{1}{i} * \sum_{S_i \in p}(S_i^+ - S_i^- - S_i - 3)$
$W_{i,j}^{(l)}$	the parameter associated with the connection between unit j and layer l
X	input features for a training example
$Z_i^{(l)}$	total weighted sum of inputs to unit i in layer $l(a_i^{(l)} = f(Z_i^{(l)}))$
$I_G(f)$	gini impurity
$I_E(f)$	entropy
$E(A)$	expected input for the tree
d	distance
v	set of branches in DT
p_i	set of positive sample in DT
n_i	set of negative sample in DT
n	number of negative sample
t_i	common feature
C_j	a class
N_j	number of features
σ	sigmoid function
W	matrix weight
g(A)	information gain
F	F-measure

Publications and further readings

- Nkongolo, M. (2018). Classifying search results using neural networks and anomaly detection. Educor Multidisciplinary Journal, 2(1): 102-127 .[1]

The interested reader may use the following link for additional information of the subject of this research.

- Mike Nkongolo (Author), 2018, Textual Classification for Sentiment Detection. Brand Reputation Analysis on the Web using Natural Language Processing and Machine Learning, Munich, GRIN Verlag.[2]

[1] https://journals.co.za/ Content/ journal/educor V2 n1
[2] https://www.grin.com/document/419732

To my parents, Professor Mukendi Jean-Paul and Melanie Masengu Ngandu…

Chapter 1

Introduction

Since the dawn of time, reputation has played a predominant role in almost all aspects of life (Confente et al., 2019). For instance, two companies operating in the same sector are competing and, according to Hanson, Jiang, and Dahl (2019) and Hirve and Reddy (2019), each company tries to protect and improve its image or reputation. An additional example is the global university ranking system presented by Peters (2019) - where universities are classified according to different metric performances and thus, each university tries to improve its reputation to stay on top of the hill. A Reputation Analysis (RA) will therefore reflect on how content extracted from various sources of data can, in one way or another, positively or negatively affect the image/ brand of a corporation (Sugiyama, 2019; Giachanou, Gonzalo, and Crestani, 2019; Khan and Digout, 2018). In this research, we define RA as the mining of customer opinions with regard to specific products or services and the judgment that customers have towards the organization's products or services (Lewandowski, 2020).

Therefore, Chen et al. (2019) argued that determining a company's reputation is vital for marketing and customer relationship management because, in many instances, marketers use different techniques to determine the reputation of a company (Kim, Youn, and Lee, 2019). Usually, surveys are often conducted to solve this problem and, occasionally, open-ended questions are generally used to gather relevant information about the reputation of the company/ brand (Yılmazel and Arslan, 2019). However and unfortunately, Lin et al. (2019) and Lukyanenko et al. (2019) has shown that it is very costly to collect and analyze the large volumes of high-quality survey data that is crucial for an in depth analysis for corporate reputation.

1.1 Problem statement

To perform RA, marketers use questionnaires to conduct opinion polls with a specific target group. Currently, with the use of the internet, a significant repository of textual opinions has been created (Gasc et al., 2019). Marketers can identify the reputation of different brands or companies using this repository of textual opinions (Gasc et al., 2019). For instance, we can cite various sources, the most popular of which are C-Net, IMDB, Amazon, RottenTomatoes, Twitter, and Facebook (Boit and El-Gayar, 2019; Al Marouf, Hasan, and Mahmud, 2019).

This research focuses on Brand Reputation Analysis and, as a case study, the reputation of the University of the Witwatersrand is assessed and analysed. It is possible to detect and analyze the opinions as well as the experiences of hundreds of people on almost all existing products (Ramanathan and Meyyappan, 2019; Rasool et al., 2019; Jadav and Vaghela, 2016). The detection and analysis of these textual

opinions over the internet has enabled marketers to determine whether a product or service has potential within the market (Ramanathan and Meyyappan, 2019). However, this task is tedious as it is done manually. Therefore, an Information System (IS) should be implemented to automate Corporate Reputation Analysis processes. Unfortunately, it might be very difficult for the framework to draw an inference (positive, negative, or neutral) as there are various contradictory opinions and this phenomenon can reduce the Accuracy rate (Hanson, Jiang, and Dahl, 2019).

The RA process is therefore a powerful conduit to automatically extract opinions and feelings from online sources and classify them as positive, negative, or neutral. This automatic and tripartite classification of textual data will eventually optimize the process of Corporate-Brand Reputation.

A RA framework is very broad and complex due to the following aspects:

1. A user query might not always provide optimal results - this affects Feature Extraction (FE) and classification;
2. FE might ignore relevant information that is useful in the RA process;
3. Variations into information sources might affect the detection, extraction, and processing mechanisms.

This study collected, extracted, classified, and visualised textual data for RA. Thus, irrelevant features were identified and significantly reduced; Machine Learning (ML) techniques that facilitate the process of classifying relevant features from irrelevant features were investigated (Hirve and Reddy, 2019). Most ML RA models use different techniques, such as Naïve Bayes (NB), Linear Regression (LR), and Support Vector Machine (SVM) (Kumar and Kumar Gupta, 2019; Rasool et al., 2019; Jadav and Vaghela, 2016). However, these techniques are not very effective for classifying textual data as relevant or irrelevant with *great* Precision (Nikam, 2015; Rasool et al., 2019; Jadav and Vaghela, 2016).

It would therefore be imperative to use robust classifiers according to various requirements of RA (Duggan et al., 2015). The exactitude of FE is of paramount importance and classifiers must be precise and very effective.

1.1.1 Objectives and research question

We considered several factors for the successful implementation of the RA framework (Nkongolo, 2018). The main objective of this research was to present an approach to use computational methods to analyse the web presence of an entity to establish their reputation. In this case, Sentiment Analysis (SA) was used as a proxy. This dissertation introduces such a framework and tests its performance on automated SA for Brand Reputation.

The sub-objectives are as follows:

1. Develop different approaches for predicting the polarity scores of web pages. An attempt was made to find out if the web page content was positive, negative, or neutral;

2. Implement automatic textual data collection in real-time;
3. Improve FE and classification Accuracy in RA;
4. Implementing solution techniques of studying the reputation of the University of the Witwatersrand on the internet;
5. Evaluating the proposed framework systematically;
6. Compare results to existing work conducted by Jadav and Vaghela (2016) and Rasool et al. (2019).

1.1.2 Research question

Main research question. Is publicly available data suitable to address the problem of Reputation Analysis ?

At best, the platform can only feasibly provide a successful performance based on the quality of the data set provided to the system (Arif-Uz-Zaman et al., 2017). The crawler is responsible for providing the quality of the textual data set [we needed a labeled data set (labels of positive, negative and neutral sentiments)] (Tsytsarau and Palpanas, 2012). If there is an insufficient volume of data, then the system may fail to perform the reputation analysis (Nkongolo, 2018). The question of whether the system can be implemented is relative and is dependent on the number of times the system is able to make a good classification (Moraes, Valiati, and Neto, 2013). We are referring to the training data utilized for training a classifier for sentiment detection. There are other factors involved in this research question. One of the main components of the system is the commitment to ensuring data extraction success (Guggilla et al., 2018). Data extraction success refers to web crawling success (Yılmazel and Arslan, 2019).

1.1.3 Motivation and importance of the research

This Section covers the importance of this dissertation in general. The objectives of this research allows us to comprehend the advantages and disadvantages of Corporate Reputation Analysis (CRA). The main challenges faced by such computational frameworks have been presented by Jo, Park, and Ha (2019) and Chiranjeevi, Santosh, and Vishnuvardhan (2019). Compared to other fields in which Natural Language Processing (NLP) has been applied, CRA on the internet appears to be challenging (Fu et al., 2012). It requires a high level of Accuracy (Khan and Digout, 2018; Sundaresan et al., 2019). Similarly, the FE component (entity recognition) would have difficulty in identifying relevant features (Donneau-Golencer et al., 2019). The lack of labelled textual datasets is another problem for RA frameworks (Nkongolo, 2018).

Essentially, a RA framework must collect, extract, and classify relevant data about a corporation (Goswami, Nandi, and Chatterjee, 2019). It is therefore necessary to provide a query to the web crawler, this query must extract textual data in full (Cho and Garcia-Molina, 2002). Thus, the query given to the RA framework must retrieve enough textual data for classification and visualization (Nkongolo, 2018).

In recent years, we have experienced a great rise in the number of RA frameworks and they are gaining popularity (Chiranjeevi, Santosh, and Vishnuvardhan, 2019; Goswami, Nandi, and Chatterjee, 2019). It becomes important for companies to have such RA frameworks for customer management and satisfaction (Arif-Uz-Zaman et al., 2017). In addition, the use of such a RA framework has the following advantages for the company that wishes to use the software:

1. Analyze the company performance;
2. Enhance the company marketing policies;
3. Estimate the popularity of a specific product or service on the market.

According to Shukri et al. (2015), the information provided by the web crawler must be correlated with the end-user query and corporate *visibility*. Fake news is another factor likely to mislead classifiers built with some ML techniques (Shukri et al., 2015). The combination of NLP and ML can overcome these difficulties, but only if it is applied with care and if researchers in the field of Software Engineering and Data Science work in synergy with ML and NLP specialists (Almatarneh and Gamallo, 2017). Particularly, Deep Learning in RA will yield various solutions (Guggilla et al., 2018). An effective technique to mitigate different issues in RA is to introduce an application of how the proposed framework works for a clearly defined system, such as CRA.

1.1.4 Research contribution

The first contribution of this research is the implementation of a computational framework that is capable of analysing the reputation of a corporation in real time, thus enabling automatic RA on the internet. This is not the first application in RA (Khan and Digout, 2018). The Reputation model implemented in this research is trained and then evaluated on some interesting textual data (Nkongolo, 2018). The RA framework presented typically reduces misclassification - contrary to existing frameworks (Jadav and Vaghela, 2016; Rasool et al., 2019).

One more implication of this contribution is to retrieve textual data with the use of a web crawler. This crawling process is strongly connected to data collection and the RA model. This allows automatic FE as an important phase in RA. Several challenges were met in the implementation of this framework. The main challenge remains incorporating different components with regard to textual data collection, FE, classification, and visualization (Hussein, 2018). The last contribution of this research is therefore to present "techniques" that advance knowledge in CRA field. Techniques presented are simple and fast enough to be easily implemented and tested but this RA framework was not evaluated by end-users. By automatically extracting and classifying opinions on specific products from the internet, we automate the RA process (Sundaresan et al., 2019).

1.1.5 Research motivation

Even though SVM and Naïve Bayes (NB) classifiers have been widely applied in various fields, there is a minuscule amount of research on CRA where Sentistrength, Bag-Of-Words (BOW), and ANN (Artificial neural network) have been utilized (Hirve

and Reddy, 2019; Hanson, Jiang, and Dahl, 2019; Jo, Park, and Ha, 2019). The reputation of two brands namely, Nike and Adidas, have been assessed by Rasool et al. (2019) (see Figure 1.3).

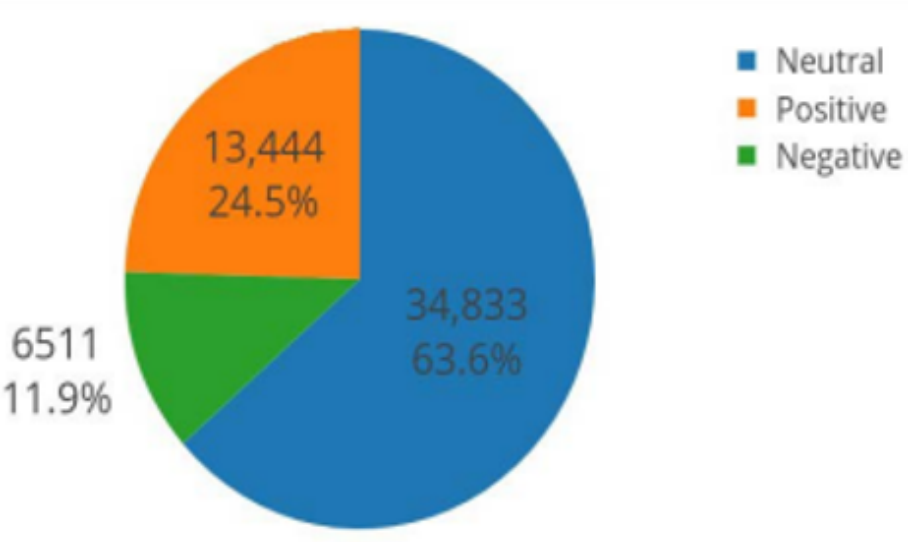


FIGURE 1.1: Reputation Analysis for Nike using Naïve Bayes (Rasool et al., 2019).

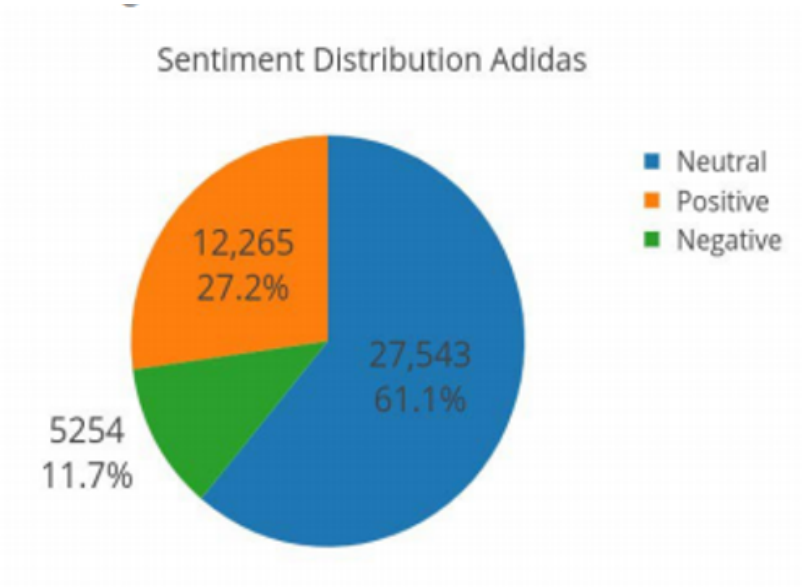


FIGURE 1.2: Reputation Analysis for Adidas using Naïve Bayes (Rasool et al., 2019).

FIGURE 1.3: Reputation Analysis for Adidas and Nike (Rasool et al., 2019).

Rasool et al. (2019) compares users opinions on brands such as, Nike and Adidas using NB. Rasool et al. (2019) then argued that Adidas critics are more positive than Nike critics and there is a very small difference from the negative comments (see Figure 1.3). Lastly, the NB classifier achieved 63% of Accuracy. This demonstrates the limitation of the NB classifier in terms of producing high Accuracy for RA frameworks. The number of papers using a combination of ANN, Sentistrength,

	NB	SVM	RBF Kernel SVM
Gold	69%	72%	73%
Movie	74%	74%	74%
Twitter	76%	76%	78%

TABLE 1.1: Accuracy with Twitter dataset and classifiers comparison (Jadav and Vaghela, 2016)

and BOW are also very limited (Li et al., 2019; Chen et al., 2019; Sugiyama, 2019). Figure 1.4 illustrates a comparative study conducted by Jadav and Vaghela (2016) in which NB, SVM, and RBF (Radial basis function) kernel SVM have been used. The best algorithm is RBF kernel SVM with 78% Accuracy on Twitter dataset.

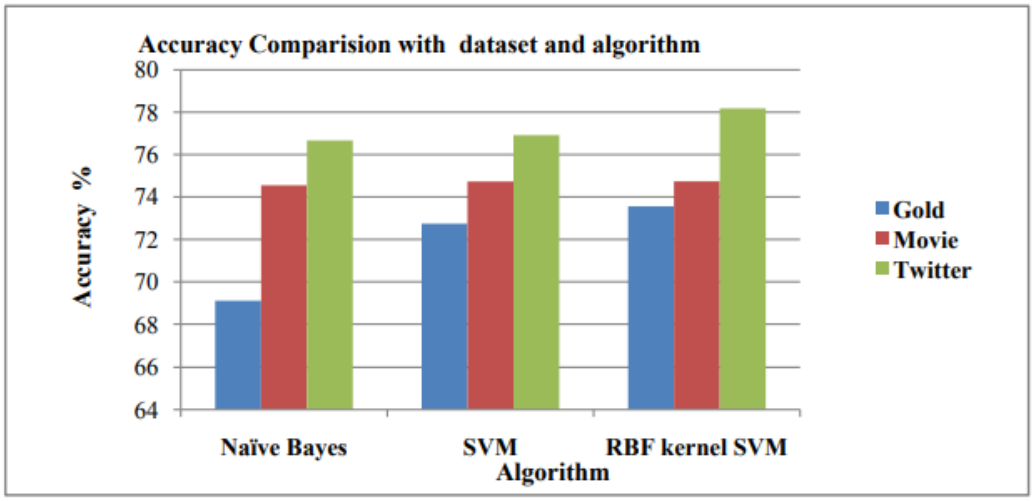


FIGURE 1.4: Algorithms comparison study (Jadav and Vaghela, 2016)

Figure 1.3, Figure 1.4, and Table 1.1 show low Accuracy produced by current ML techniques used in RA. Hence, this research used three classifiers: ANN, Sentistrength, and BOW algorithms. These classifiers are combined with a Named Entity Recognition (NER) in order to enhance Accuracy predictions and minimise classification errors (Seth et al., 2019).

1.2 Research tasks

The RA framework presented in this research focuses mainly on the implementation of various components to create the final robust and accurate RA system. These are the main tasks completed in this research:

1. Dataset collection: this involves collecting the required textual data sets using a web crawler as in Hong, Lee, and Yu (2019) and Tabassum and Khan (2019).

2. **Feature Extraction (FE):** this task includes designing and extracting a novel, *discriminative* set of features from the input texts from the data sets (Tabassum and Khan, 2019). This is likely the most integral step in the research, as this step has a direct effect on the eventual sentiment detection rates of the system (Taboada, 2016). These features are gradient-based spatio-temporal features as well as Deep Learning-based features (Hong, Lee, and Yu, 2019).

3. Classifier selection, design, and implementation: this task consists of selecting, designing, and implementing appropriate classifiers for the system (Nkongolo, 2018). The classifier was trained on the extracted features. The chosen classifiers are ANN, Sentistrength, and BOW algorithms.

4. Reporting and visualisation: after classifying textual data, the RA model concentrates primarily on the visualization/ reporting (Chou, Chao, and Yu, 2019). This implies that for each web-page crawled, the framework assigned a polarity or prediction. This prediction can only be positive, negative, or neutral (Vural et al., 2013). Positivity was marked by a yellow color, negativity by a red color, and neutrality by a green color.

5. Data labelling: at this level, the framework provides the opportunity to read the content of the web page in order to test the Accuracy of classifiers (Hirve and Reddy, 2019). The end-user can read the textual content of a web-page and find out its polarity. He/ she can now compare the discovered polarity with that given by classifiers.

1.3 Structure of the document

The dissertation is structured as follows :

1. Chapter 2 presents a detailed, concise literature review of what relevant work has been done in this problem domain in the fields of ML and NLP. Chapter 2 also discusses related works which focuses on RA, SD (sentiment detection), Web Crawling, and visualization. The current state-of-the-art approach to the problem was also identified and discussed in Chapter 2.

2. Chapter 3 discusses the proposed methodologies that were employed in this research as well as techniques used.

3. In Chapter 4, we present an implementation of the framework as well as the data set used.

4. Chapter 5 consists of experiments and results as well as an objective discussion regarding the results.

5. The final Chapter 6 consists of the summary and conclusion of the dissertation.

1.4 Conclusion

In this Chapter (Chapter 1), we proposed a RA framework that is effective for conducting Brand Reputation Mining on the internet. The framework is concerned with Information Retrieval (IR), the extraction and mining of mentions expressed across the internet in respect to a particular corporation, institution, or organization, and the tone and sentiment of such mentions.

Chapter 2

Literature review

NLP and ML techniques are used in CRA - these techniques rely on automatic and real-time data collection, FE, textual classification using SA and, finally, data visualization (Lam et al., 2019). Hence, this Chapter 2 debates literature with regard to techniques used in CRA, such as Web Mining, SD, classification algorithms and visualization (Lam et al., 2019).

2.1 Corporate Reputation Analysis (CRA)

According to Shukri et al. (2015), NLP techniques, such as SA (sentiment analysis), can be applied in different sectors to study the reputation of companies. Shukri et al. (2015) used SA to compare the reputation of three leading automotive industry companies (Mercedes, BMW, and Audi) using NB classifier. Shukri et al. (2015) extracted customers opinions for each company - results showed that Audi's has the highest, positive polarity of 83% compared to BMW and Mercedes. A sample of textual feature extracted in Shukri et al. (2015) is presented in Table 2.1 and Table 2.2.

Tweets	Sentiment Detection
@MercedesBenz Intelligent	Positive
@Amazing @MercedesBenz	Positive
@MercedesBenz-issue	Negative

TABLE 2.1: Tweets' sample for Mercedes (Shukri et al., 2015)

Tweets	Sentiment Detection
@Worst decisions - buying	Negative
@Proud to own an Audi	Positive
@Sorry - this is rubbish	Negative

TABLE 2.2: Tweets' sample for Audi (Shukri et al., 2015)

The study of CRA began in the 1980s (Mescheder, Nowozin, and Geiger, 2017). This study captured the attention of many researchers from different fields, such as economists, sociologists, psychologists, computer scientists, and marketers (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). For instance, economists use strategies from game theory to describe or predict reputation (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). Furthermore, sociologists define CRA as a description of rivalries between different firms for the benefit of a given society (Chiranjeevi, Santosh, and

Vishnuvardhan, 2019). This definition is similar to management specialists - defining CRA as the behavioral judgment of a person towards a company reputation (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). This research unites these definitions in order to create a system capable of automating CRA. Thus, we define CRA as an automatic/ manual collection of Big Data for user opinions classification extracted on the internet and thus reflecting the performance, visibility and relative image of a company (Guggilla et al., 2018) - this reputation influences the *attractiveness* of companies as well.

2.1.1 CRA techniques

Schwaiger (2004) stated that, CRA can be implemented using the following techniques:

1. Overall Reputation Score (ORS): this score represents the average arithmetic score of a web page. A number of papers criticize this technique as outdated (Schwaiger, 2004).

2. Principal Component Analysis (PCA): this algorithm minimizes the set of variables p to another set of m underlying dimensions (Webster, 2001). This yields correlations between p variables (Webster, 2001). PCA was used to study the reputation of American universities as discussed by Webster (2001). PCA can also be used as a reputation evaluation model and achieved better Accuracy prediction (see Figure 2.1).

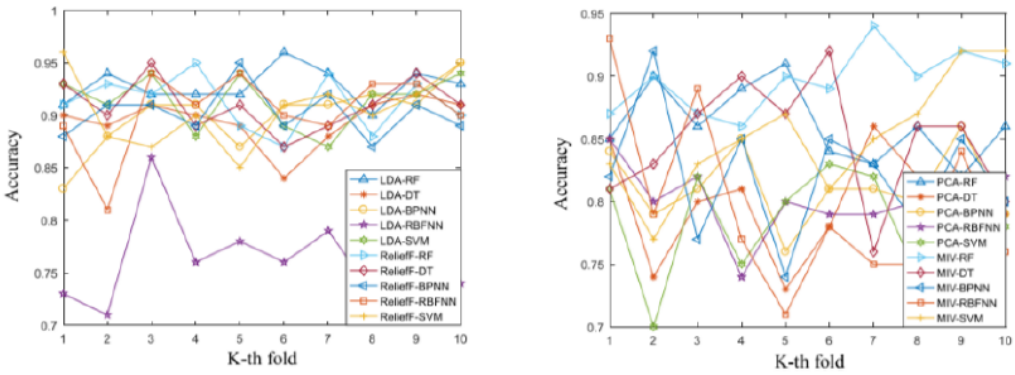


FIGURE 2.1: Reputation evaluation models comparing ReliefF, PCA and LDA feature dimensionality minimization techniques (Huang and Chen, 2019)

3. Page rank algorithm: this algorithm had several modifications since its invention by Larry Page (Massucci and Docampo, 2019). The algorithm categorizes web pages using their importance as a classification metric. Given that the internet represents a set of web pages interconnected as nodes; the algorithm starts by associating a score with each node (Webster, 2001). The score can be based on the number of citations received if the algorithm is to be used to study the reputation of a research paper for instance (Massucci and Docampo, 2019).

Recently, Huang and Chen (2019) focused on optimizing and evaluating the cumulative reputation of *crowdsourcing*. This optimization was implemented in a Cloud Computing environment such as Big Data (Huang and Chen, 2019). Huang and Chen (2019) built the optimization model of reputation evaluation by implementing different techniques such as Random Forest (RF) and Decision Tree (DT).[1] The multidimensional index and ML techniques were used to provide a model for evaluating the reputation of *crowdsourcing*. This model is based on Linear and random Discriminant Analysis Forest (LDA-RF) algorithm (Huang and Chen, 2019).

Classifiers	RF	DT	SVM	Page rank
LDA	0.92	0.89	0.91	3.00
PCA	0.86	0.79	0.78	1.40
Relief	0.91	0.91	0.90	3.60

TABLE 2.3: Algorithmic comparative study (Huang and Chen, 2019)

In a paper by Huang and Chen (2019), RF was used to validate ML methods, RF was then compared to other models, such as DT, ReliefF, Page rank and SVM. Table 2.3 shows the results obtained by Huang and Chen (2019) - the research concludes that RF is suitable in optimizing and evaluating the cumulative reputation of *crowdsourcing*. ReliefF algorithm, in return, provides classification Accuracy of 0.928, 0.862 and 0.912. In Huang and Chen (2019), PCA reduced feature dimensionality and the Precision of this algorithm is also impressive (see Table 2.3).

2.1.2 Reputation Analysis and Feature Extraction

Additionally, Chen et al. (2019) presents a CRA model evaluating an organization's reputation. The results demonstrate the effectiveness of the Internet of Things (IoT) architecture in RA (Chen et al., 2019). Figure 2.2 depicts a decrease of corporate reputation with time using NB and Organisation Reputation Evaluation Scheme (ORES) method (Chen et al., 2019).

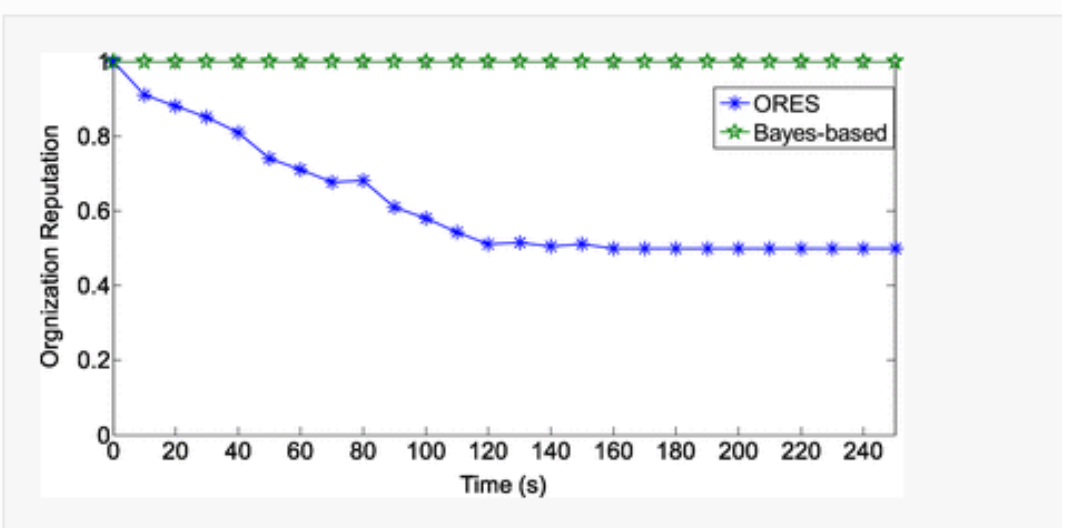


FIGURE 2.2: In this plot ORES (organisation reputation evaluation scheme) outperforms NB method (Chen et al., 2019)

[1] See Figure 2.1 and Table 2.3

The work proposed by Unnisa, Ameen, and Raziuddin (2016) introduces a reputational framework that can collect data on Twitter and makes use of a SA technique to process extracted features (Unnisa, Ameen, and Raziuddin, 2016). Relevant tweets are categorized into two groups (positive and negative). Unnisa, Ameen, and Raziuddin (2016) uses unsupervised ML techniques such as Spectral Clustering (SC) algorithm to achieve this goal. Since manual analysis of such Big Data is impossible, Unnisa, Ameen, and Raziuddin (2016) therefore proposes an automated and unsupervised approach.

2.2 ML and CRA

This Section 2.2 discusses various ML methods used in CRA. These methods have been used in the past beside SA (Moraes, Valiati, and Neto, 2013). This Section also addresses the difference of these ML methods (PCA and SC) to the approach presented in this research (ANN and BOW). Lastly, the Section also discusses the *strength* of web presence and search result ranking.

2.2.1 ML

ML is a sub field of AI (Artificial Intelligence) interested in automated detection of significant models in data (Bottou, Curtis, and Nocedal, 2018). The particularity of ML is that it equips computer programs with intelligence allowing them to learn and to adapt (Bottou, Curtis, and Nocedal, 2018). This learning can be defined as the process of converting the computer program experience into knowledge (Bottou, Curtis, and Nocedal, 2018). The goal of ML is to take into account the capabilities of computer programs to solve complex tasks such as classification (Bottou, Curtis, and Nocedal, 2018). We can divide ML into the following categories:

2.2.2 Supervised learning

This category of ML requires a set of training data in which each data point is labelled (Jamal et al., 2018). The algorithm uses this training data to indicate the class in which new entries belong (positive, negative or neutral). This algorithm builds a model that distinguishes features (Jamal et al., 2018). ANN, DT and SVM are supervised learning algorithms.

DT uses the concept of *nodes*, *edges* and *leaves*. Each node is assigned to sample attributes and these attributes correspond to the value of each branch (Jamal et al., 2018). $p+1$ is a vector element that represents some attributes $(p1, p2, p3, ..., pn)$. n is the number of attributes and each branch descending from n is a prediction of a test (Jamal et al., 2018). DT is easy to implement but the prediction Accuracy in RA seems to achieve a lower Accuracy rate (Jadav and Vaghela, 2016). In terms of speed, DT outperforms ANN as demonstrated by Jamal et al. (2018) - this is due to the fact that DT ignores irrelevant features. However, ANN does consider both relevant/ irrelevant feature, unless if we pre-process data (Rawat and Malhan, 2019). Therefore, in CRA, it is important to classify a significant amount of data and ANN is suitable in this scenario (Rawat and Malhan, 2019).

Information theory provided a technique to describe the order of attributes in DT (Jamal et al., 2018). For each attribute, the technique calculates the information gained and classifies the attributes based on certain parameters (Jamal et al., 2018). The

highest information gained is represented by the root node and the information gain decreases from the second node and so on (Jamal et al., 2018). In RA, the information gained can be used during FE to determine the degree of extraction (Jamal et al., 2018). This information gained is calculated using the following Equation 2.1:

$$gain(A) = I(p,n) - E(A) \tag{2.1}$$

$E(A)$ is the expected information for A_1. $E(A)$ can be computed using the following Equation 2.2:

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n}(p_i, n_i) \tag{2.2}$$

Where v is the set of branches from A (A is a root node for the tree), p_i is the set of positive samples for the i^{th} branch. n_i is the set of negative samples for the i^{th} branch. n is the number of negative samples while p is the set of positive samples. To split features, Entropy $I_E(f)$ and Gini impurity $I_G(f)$ are used as *thresholds* (Jamal et al., 2018):

$$I_G(f) = -\sum_{i}^{m} f_i(1-f_i) = 1 - \sum_{i}^{m} f_i^2 \tag{2.3}$$

$I_E(f)$ defines similarity measures on a target feature (Jamal et al., 2018). Having a group of variables $i \in (1,2,...,m)$ considering f_i as a fraction of item. Thus:

$$I_E(f) = -\sum_{i=1}^{m} f_i log_2 f_i \tag{2.4}$$

and

$$\sum_{i}^{m} f_i = 1. \tag{2.5}$$

Baradwaj and Pal (2012) used DT algorithm to assess student achievement. The gain ratio and value results are presented in Table 2.4. This ratio was used for feature selection (Baradwaj and Pal, 2012). Information gain has been used by Baradwaj and Pal (2012) to find the best attribute.

Gain value	Gain ratio
0.577	0.416
0.515	0.355
0.365	0.229
0.218	0.0288

TABLE 2.4: Gain ratio and gain value comparison (Baradwaj and Pal, 2012).

2.2.3 Unsupervised learning

In this category of ML, there is no need for a training sample. The prediction is detected by the algorithm (Baradwaj and Pal, 2012). Clustering is an example of unsupervised learning algorithms (Baradwaj and Pal, 2012). This algorithm determines how to group similar features without a training process. The PCA is also an example of unsupervised learning in which feature dimension and estimation of the

given dataset is reduced (Baradwaj and Pal, 2012). Features are grouped by $affinity$ with SC algorithm - this algorithm uses nearby points to classify features (Baradwaj and Pal, 2012). To achieve this, the algorithm constructs an $affinity$ matrix for a set of n data points $(n * n)$. Pairwise affinities are calculated for the dataset (Baradwaj and Pal, 2012). Affinity can be seen as a distance metric. *Eigenvectors* with the largest eigenvalues are then extracted by the PCA from the affinity matrix (Baradwaj and Pal, 2012). After this, the data is projected to a new space where features are clearly distinguished. The distance d between two data points i and j is calculated by (see Figure 2.3):

$$d(i,j) = max(d_k(i), d_k(j), d(i,j)) \tag{2.6}$$

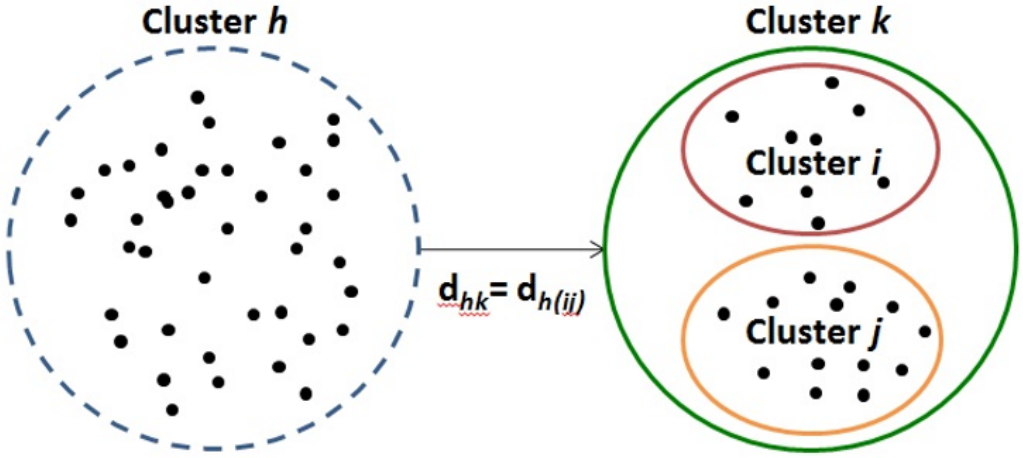


FIGURE 2.3: Computing *Reachability* distance between two clusters i and j using SC.

In CRA, textual data is represented in the form of BOW (Rawat and Malhan, 2019). Therefore, the words found in textual data are represented according to their appearance in the text (Rawat and Malhan, 2019). In this manner, it is also important to reduce the dimension of feature (Rawat and Malhan, 2019). In this research we also reduced the dimension of features - this process was performed by a NER but not PCA. We opted for the NER because it is easy to implement and the result produced by this method is easy to analyze (Lu and Zhai, 2008).

2.3 Strength of web presence

Web presence is defined by the fact that any company that would like to improve its on line reputation must consider its presence on the internet in order to strengthen its marketing and services popularity (Jones, Temperley, and Lima, 2009). The company will need to have a large presence on the web in order to monitor its reputation. Jones, Temperley, and Lima (2009) argued that there is a relationship between web presence and having a great *visibility* on the internet. The advantage of having a great visibility is that it can attract customers (Jones, Temperley, and Lima, 2009). We then present concepts to take into consideration with regards to strengthening the web presence (Jones, Temperley, and Lima, 2009):

1. Search engine optimization: it is obvious that a corporation needs a website that can be found by search engines. According to Jones, Temperley, and Lima (2009), 89% of customers utilise search engines to investigate a company's products or services before making a decision. Any corporation should take advantage of this, by considering Search Engine Optimization (SEO) for the website (Jones, Temperley, and Lima, 2009). SEO uses Page rank algorithms, such as Google, to generate a result of a query (Jones, Temperley, and Lima, 2009). This algorithm is based on incoming links of a specific web page. The algorithm assigns an important rank of all pages on the internet. It then considers the quality of outbound link and the rank of a page (Jones, Temperley, and Lima, 2009). From a linear algebra view point, let's denote by $x1, x2, x3$ the importance of three web pages. By observing the scenario carefully, we have the following for each node:

$$x1 = 1.x3 + \frac{1}{2}.x2 \tag{2.7}$$

$$x2 = \frac{1}{3}.x1 \tag{2.8}$$

$$x1 = \frac{1}{3}.x1 + \frac{1}{2}.x2 \tag{2.9}$$

If the importance of a web page is distributed among four nodes, each node gets $\frac{1}{4}$ importance score (Michalski, Carbonell, and Mitchell, 2013). v is the initial rank (vector) with entries equal to 14. The incoming link augments the importance of a page (Michalski, Carbonell, and Mitchell, 2013). In the first stage, the algorithm updates the rank score of each page by multiplying the matrix A with v. At this stage, the vector is $v_1 = A_v$. The algorithm repeats the process recursively and updates the importance vector $(v_2 = A(A_v) = A2v)$. By performing the numerical computation we have the results shown in Figure 2.4.

$$\mathbf{v} = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \quad \mathbf{Av} = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}, \quad \mathbf{A^2 v} = \mathbf{A\,(Av)} = \mathbf{A} \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

$$\mathbf{A^3 v} = \begin{pmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{pmatrix}, \quad \mathbf{A^4 v} = \begin{pmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^5 v} = \begin{pmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{pmatrix}$$

$$\mathbf{A^6 v} = \begin{pmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^7 v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^8 v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

FIGURE 2.4: Numerical computation of Page rank algorithm.

2. The corporate web site: This web site is important as it contains information about the company itself. To enhance its visibility, the information contained in

the corporate web site should portray the company vision and mission (Khan and Digout, 2018).

3. Social media: referred to as the communication channel with website and applications to share content or participate in social networking (Khan and Digout, 2018). Corporations can use this online platform to enhance their visibility on the internet (Khan and Digout, 2018).

2.3.1 Search result ranking

When a user provides a query to the search engine, the search engine starts by attempting to specify where the content of the web pages related to the query should appear (Duggan et al., 2015). This process is called ranking. More often, less-visible content of a web page may not appear after searching for it, while highly visible content appears (Duggan et al., 2015). It is therefore crucial for the corporation to improve its visibility to facilitate the customer, while researching about the service they provide (Duggan et al., 2015). The paper by Agichtein, Brill, and Dumais (2006) investigates the usefulness of responses obtained in real time web searching. Agichtein, Brill, and Dumais (2006) study aimed to enhance web search ranking by using 3,000 queries and a Big Data of user interactions with a search engine (Agichtein, Brill, and Dumais, 2006). The experiments demonstrate an amelioration of techniques used, such as MAP (mean average precision) - (see Table 2.5).

Method	MAP	Gain	Gain
RN	0.269	0.51	10%
RN + All	0.321	0.51	10%
BM25F	0.236	0.56%	31%
BM25F + All	0.292	0.56	31%

TABLE 2.5: Best method (Agichtein, Brill, and Dumais, 2006)

2.4 NLP in CRA

Beside ML, NLP is also a sub field of AI (Oshikawa, Qian, and Wang, 2018). NLP enables computer programs to mimic human intelligence by understanding and interpreting human languages (Oshikawa, Qian, and Wang, 2018). Humans can therefore communicate with computer programs using NLP - because NLP enables computer programs to read text and hear speech (Oshikawa, Qian, and Wang, 2018). NLP is also an interdisciplinary discipline connected to computer science and computational linguistics (Oshikawa, Qian, and Wang, 2018). In general, NLP segments textual data into pieces/ tokens (words, frequency, periods, etc.) and understands the correlation of tokens (Oshikawa, Qian, and Wang, 2018). To achieve this goal, NLP makes use of the following features:

1. Machine Translation (MT): this feature of NLP interprets speech or textual data into another language (Oshikawa, Qian, and Wang, 2018);

2. Content classification: indexing, searching, anomaly detection, and redundancy detection is used by this feature;

3. Topic Modeling (TM): detect meanings of data (speech, text) using Data Mining (Oshikawa, Qian, and Wang, 2018);

4. Data Extraction (DE): this feature automatically retrieves information from various sources;

5. SA: this feature is useful in Data Analytics (Oshikawa, Qian, and Wang, 2018) - it automatically identifies opinions of data stored;

6. Text-to-speech and speech-to-text recognition: translates text into speech, and vice versa (Oshikawa, Qian, and Wang, 2018).

2.4.1 Sentiment Detection (SD) in NLP

NLP has been widely used in CRA to process data using simplistic techniques, such as $n-gram$, Part of Speech tags (POS), and NER (Oshikawa, Qian, and Wang, 2018; Arpat, Shah, and Ayyar, 2018). For instance, Hussein (2018) evaluated the performance of NLP techniques. Following this, a discussion of different types of $n-gram$ Feature Selection techniques that capture sentiment cues including POS and characters were presented (Hussein, 2018). Hussein (2018) argued that these techniques can be used in SD, such as words segmentation, Maximum Match Algorithm (MMA), Conditional Random Filed (CRF) algorithm, and evolutionary algorithms. N-gram is used mostly in NLP and they are useful in SD (Hussein, 2018). According to Hussein (2018), BOW is also part of N-gram. Therefore, in our research, we used a similar technique in terms of Feature Selection as presented in (Hussein, 2018; Gonçalves et al., 2013). The framework used by Hussein (2018) is presented in Figure 2.5.

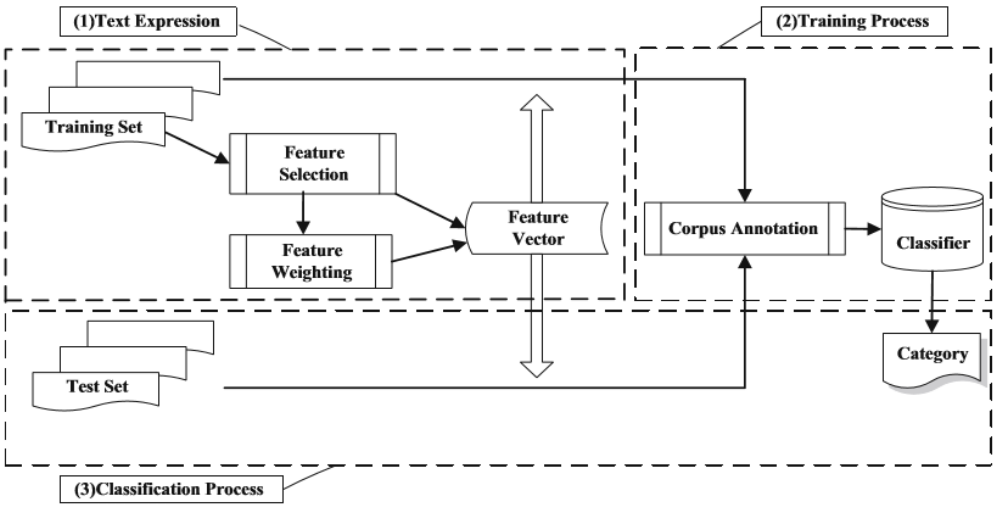


FIGURE 2.5: Framework architecture (Hussein, 2018).

In Hussein (2018), Feature Extraction (FE) for Sentiment Detection provides quality feature subsets and enhances Accuracy. The extraction algorithm used by Hussein (2018) sums up the text with common feature t_i. The computation is as follows:

$$FE(t_i) = \sum_{j=1}^{m} N(C_j, t_i) \tag{2.10}$$

Where $N = 2$, C_j is the class j and $N(C_j, t_i)$ is the number of feature $t_i \in C_j$. The optimisation of FE is therefore:

$$FE(C_j) = \sum_{i=1}^{N_j} N(C_j, t_i) \tag{2.11}$$

$$FE(C_j, t_i) = \frac{FE(C_j)}{NC_j} * N(C_j, t_i) \tag{2.12}$$

$$FE(t_i) = \sum_{j=1}^{M} FE(C_j, t_i) \tag{2.13}$$

N_j and NC_j is the number of features. We present results of Feature Selection (FS) obtained by Hussein (2018) in Table 2.6.

POS	FS 1	FS 2	FS 3	FS 4	FS 5
Nouns-grams	76%	70%	73%	78%	74%
Verbs-grams	74%	81%	80%	77%	78%
Adjective-grams	90%	93%	93%	92%	92%
Adverbs-grams	82%	92%	90%	90%	88%
4-POS	89%	94%	96%	92%	93%

TABLE 2.6: N-gram performance (Hussein, 2018)

2.4.2 Sentiment Analysis (SA) and Web Mining (WM)

Kosala and Blockeel (2000) focused on internet exploration. They researched on some common misunderstandings about the term WM. They then came up with three categories of Web exploration, which comprise of the contextual exploration, structural exploration, Web crawling.

Furthermore, Kosala and Blockeel (2000) have made some investigations into these categories (see Table 2.7). Finally, they analyzed the relationships between the above-mentioned exploration categories - in their survey, Kosala and Blockeel (2000) focused more on representational issues, processes, and learning algorithms.

Contextual Exploration	**Structural Exploration**	**Web Crawling**
Web site as DB	Links structure	Interactivity
Text Documents	URLs Structure	Server Logs
NER	Graphs	Relational Table
ML - NLP	Proprietary Algorithms	Statistical
Clustering	Categorization	Marketing

TABLE 2.7: Web exploration categories.

Moreover, O'Leary (2013) investigated some of the concerns of AI for Big Data. AI is applied in different ways to optimize the collection and classification of Big Data (O'Leary, 2013). Additionally, it can be used to model data of significant size to make sense of it (O'Leary, 2013). O'Leary (2013) conducted various studies analyzing all emerging issues and approaches integrating AI and Big Data.

Nasukawa and Yi (2003) illustrates a method of analyzing sentiments in order to collect feelings with a binary polarization (positive or negative). This involved an analysis of sentiment on textual data. Their approach was to detect sentimental expressions related to textual content. To detect sentimental expressions, Nasukawa and Yi (2003) implemented NLP algorithms. POS tagging allowed them to disambiguate polysemic expressions such as "like", which specifies sentiment when used as a verb. Syntax analysis has been implemented to detect relationships between sentiment expressions and the term subject.

To measure the quality of SA, Nasukawa and Yi (2003) developed a reference corpus comprising 175 cases of terms having contexts extracted from web pages. Each case was detected manually to finally represent a positive or negative feeling towards the subject. They detected 118 cases with positive polarity and 58 cases with negative polarity. Since we can represent sentiments using various expressions, including expressions that require our logic to be determined as a feeling, it was difficult to actually demonstrate the applicability of their system for SA.

Their experiment shows that we can actually collect relevant data on the sentiments of most texts (Nasukawa and Yi, 2003). The results found gave an Accuracy of about 95% and about 20% Recall. Nasukawa and Yi (2003) had 2,402 entries in the sentiment dictionary, as we can see in Table 2.8. They also used regular expressions in 14 cases.

POS	Total	Positive	Negative	Neutral
Adjective	2,465	969	1.495	1
Adverb	6	1	4	1
Noun	576	179	388	9
Sentiment verb	357	103	252	2
Transfer verb	109	-	-	-

TABLE 2.8: Distribution of Sentiments terms.

In return, Gonçalves et al. (2013) suggested a benchmark comparison study of 24 SA techniques (state of practice methods). The evaluation is based on a benchmark of 18 data sets, from social, movie and product postings, as well as opinions and comments in news articles (Gonçalves et al., 2013). The result shows that the evidence of measuring the predictive efficiency of these methods depends on a data set.

Gonçalves et al. (2013) deployed their code and all methods. This deployment created a reference system, which provides an API (Application Programming Interface) to access and compare SA techniques by referring to sentences (Gonçalves et al., 2013). They then tested techniques on all the data tagged (via AMT) from Twitter, MySpace, Runners World, BBC, Digg, and YouTube. Afterwards, calculations were performed to measure the average of $F-1$ - an average coverage of data from the internet (Gonçalves et al., 2013). They also calculated coverage based on the data set, averaging the results on the six notable events. Gonçalves et al. (2013) eventually implemented a tool based on ML called SailAil (Sentiment Analyzer (SASA)). SASA is an application based on ML techniques such as *Sentistrengh* and was tested with 17,000 tweets tagged on the 2012 US elections. Finally, a Friedman test was used for classification. To perform this statistical test, they categorized techniques in

descending order of $Macro - F1$ for each data set. More logically, the Friedman rank test in their experience is defined as follows:

$$F_R = \frac{12}{rc(c+1)} \sum_{j=1}^{c} R_j^2 - 3r(c+1) \tag{2.14}$$

In Equation 2.14, R_j^2 is the square of the sum of the rank positions of the technique $j(j = 1, 2, 3, ..., c)$. The quantity of information/ data is represented by r whereas c represents methods (Gonçalves et al., 2013).

Similarly, Vural et al. (2013) implemented a framework performing unsupervised sentimental analysis using textual data in Turkish. They have adapted the *Sentistrength* SA library by interpreting its lexicon in Turkish. Vural et al. (2013) created a system solving the problem of polarity classification. To test the performance of the implemented system, Vural et al. (2013) made use of a large body of Turkish film critics collected from different Turkish social networks. However, their platform is not supervised, but it provides a good classification, similar to the performance of supervised polarity classification methods (Vural et al., 2013). Table 2.9 is an example of English sentences and the polarity produced by *Sentistrength*.

Sentence	Positive	Negative	Binary Prediction
I can play Chess	+1	-1	+1
I like to read science	+2	-1	+1
This book is boring	+1	-2	-1
I hate you	+1	-4	-1
We loved each other but she disliked me	+4	-3	-1

TABLE 2.9: Polarity of English sentences with Sentistrength.

2.5 Focused Web Crawling

The literature related to the crawler are discussed by Fu et al. (2012). Fu et al. (2012) implemented a minimalistic crawler that can analyze opinions on the internet. This crawler consists of four modules: search engine, data management, classifiers, and data visualization (Fu et al., 2012). The data management component organized *URLs* by assigning weights to them. The search engine component categorized *URLs* in descending order FIFO (First in, First Out) based on their rank in the data management component (Fu et al., 2012). The design of the system is shown in Figure 2.6.

2.5.1 Visualization in Web Mining

One good way to assess the value of results is to examine them visually (Unwin, 2000). This should be a major application of visualization methods in Web Mining. However, there has not been much use of visualization in WM and there are few graphics available in Data Mining software (Unwin, 2000). This may be because many methods work with only discrete variables (so that continuous variables have to be discretized) and because graphical tools for multivariate categorical data are

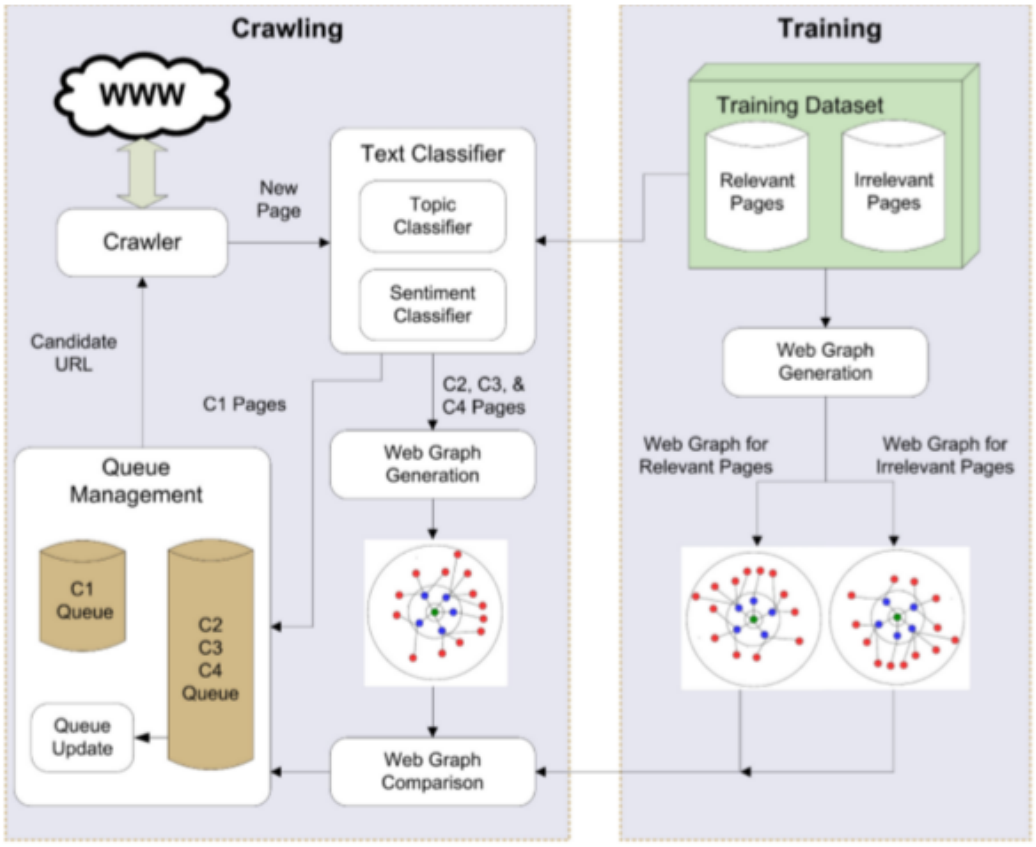


FIGURE 2.6: The web crawler (Fu et al., 2012)

not well-known or are considered ineffectual (Unwin, 2000).

There are a number of more sophisticated visualization techniques (Keim, 2002). For instance, Wang et al. (2012) proposed a Reputation Analysis System which processes Japanese text in real-time. Wang et al. (2012) have presented their improved, unsupervised algorithms which achieved 70% of Precision for opinion sentence classification. The framework of the system is comprised of their algorithms, Data Extraction, and Data Visualization. With an impressive Graphical User Interface (GUI), the system achieved 63% of Average.

2.5.2 Visualisation in CRA

Finally, Musto et al. (2015) created *CrowdPulse* - a purely agnostic system for textual analysis of social flows. The system performs social media analysis and makes use of algorithms for semantic processing (Musto et al., 2015). The system was implemented in order to detect the most dangerous zones of Italian territory, according to the content posted on social media (Musto et al., 2015). This system has been deployed to monitor the state of the city of L'Aquila after the terrible and shocking earthquake of April 2009 (Musto et al., 2015). With this in mind, the system has demonstrated its effectiveness in the context of the use of such technology in a remarkably innovative way (Musto et al., 2015).

Crowdpulse presents itself as a real-time framework of Semantic Analysis of social

flows. This platform focuses on the analytic approach. Each analysis is performed while executing the extraction of heuristic processing (Musto et al., 2015). In a typical case, a user interacts with the social network system on which he wants to analyze/ apply the heuristic. The user must then specify the type of process that he/ she would like to perform on the data content (Musto et al., 2015). The platform aims to extract, analyze, aggregate, and classify a huge amount of data/ information, which is of utmost importance for users (Musto et al., 2015). The architecture of the system presented by Musto et al. (2015) is explained in Figure 2.7, and a small description of each part of the system is provided.

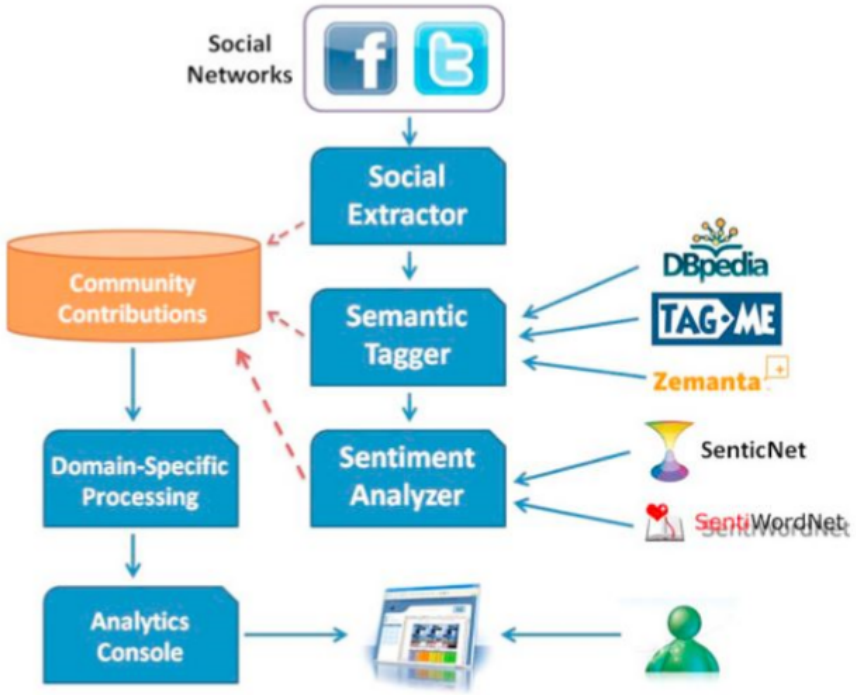


FIGURE 2.7: The architecture of the semantic content analysis framework (Musto et al., 2015).

1. Social extractor: populates a distributed database with relevant information pertaining to social network (tweets-hashtag, messages crawled),
2. Semantic tagger: categorizes each form of content (Tag.me and DBpedia was used) (Musto et al., 2015),
3. Sentiment analyzer: provides a polarity to each content,
4. Domain-specific processing: uses ML and NLP to produce results required for each specific scenario,
5. Analytics console: displays expectations via a GUI.

2.6 Conclusion

The purpose of this Chapter was to review literature of corporate reputation, evaluate its historical development, and bring a definition that serves as the theoretical basis of this project. This literature review has provided a detailed discussion of the key concepts used in the area of RA frameworks. Methods of segmenting textual data are discussed, as well as web crawling/ mining process. The Chapter ends by discussing the Sentiment Detection/ Visualization techniques used in CRA, NLP, and ML. In the next Chapter 3, methods applied in this research are explained.

Chapter 3

Research method

In Chapter 2, the multitude of approaches and techniques that were chosen to solve the problem of RA were reviewed. The best approaches typically pay careful attention to the features that are fed into the ML models, and particularly, how these features are extracted (Sänger and Pernul, 2018). This Chapter 3 serves to clearly detail the objectives of this study, and the research methodology followed during the course of the research.

3.1 Textual data retrieval

Firstly, we defined an entry page for textual data retrieval. One web page should contain URLs of other web pages. We then retrieved these URLs from the current page and added all of these affiliated URLs into the crawling queue (Xu, Gao, and Callan, 2018). Next, we crawled another page and repeated the same process as the first one recursively (Xu, Gao, and Callan, 2018). Essentially, we assumed the crawling scheme as *depth* - search or *breadth* - traversal (Xu, Gao, and Callan, 2018). As long as we accessed the internet and analyzed the web page, we crawled a website (Xu, Gao, and Callan, 2018).

Following this, we extracted the body content of the web page crawled and applied NER to eliminate irrelevant information (Xu, Gao, and Callan, 2018). Lastly, the relevant information was recorded in the database (DB) for later processing (see Figure 3.1).

FIGURE 3.1: Data collection and labelling.

The crawler performs on line textual data collection while the NER detects relevant features (words and sentences) within the data provided by the crawler (Xianghua et al., 2004). The NER then passes relevant features to the DB for later processing (Xianghua et al., 2004).

3.1.1 Sentiment Analysis, NLP and ML

The SA process classified the polarity of texts retrieved at different levels - an attempt is made to determine the polarity of textual data - whether the content is positive, negative, or neutral (Calefato et al., 2018). In our research, FE relied on *N*-gram or BOW techniques that are prominent in modern NLP (Sugiyama, 2019). *N*-gram is simply an aggregation of aggregate sequences as they appear in texts (see Figure 3.2).

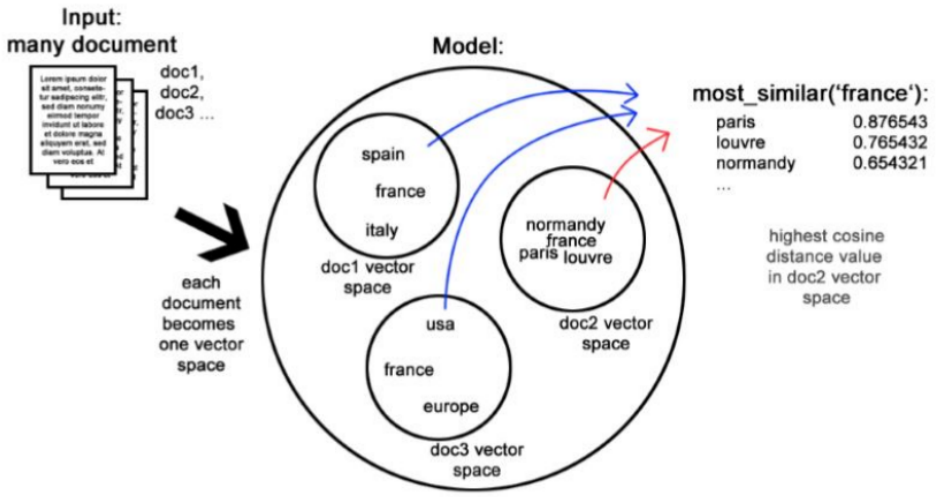


FIGURE 3.2: N-gram model.

The intuition is that n represents the frequency of aggregates sequence (Arif-Uz-Zaman et al., 2017). The parsing must be applied in order to obtain the syntactic paths (Ramanathan and Meyyappan, 2019). Nowadays, many analyzers are available for many languages - unfortunately, there are no parsers for all languages (Li et al., 2019). However, for English or Spanish, there is a plethora of parsers (Volcani and Fogel, 2006).

3.1.2 BOW

The BOW model is a representation used in NLP (Goswami, Nandi, and Chatterjee, 2019). This model represents text as a multi-set of words, without focusing on grammar or the order of words. However, the BOW model keeps multiplicity (Goswami, Nandi, and Chatterjee, 2019). BOWs can also be used in computer vision (O'Leary, 2013). This model is usually juxtaposed with classification algorithms in which the frequency of each word is utilized as a property forming a classifier (Deep Learning) (Sugiyama, 2019). In our research, the BOW model was used as a technique for generating words frequencies (Goswami, Nandi, and Chatterjee, 2019). After transforming textual data into BOW, we computed calculations to measure textual characteristics. One of the most used computation was word frequency or the number of times a term appears in the text (Goswami, Nandi, and Chatterjee, 2019) - see Figure 3.3.

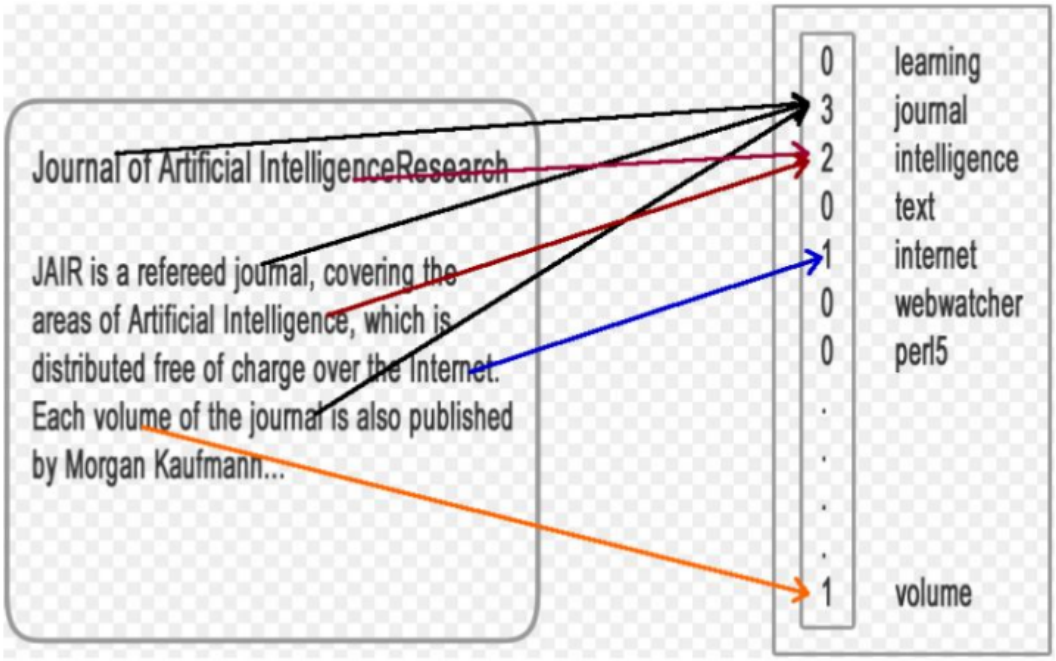


FIGURE 3.3: BOW model.

3.2 Autoencoders

In the field of ML, we can represent documents in the form of BOW, which minimizes textual observations to a vector of a fixed length (Zhang, 2018). Aside from its simplicity, BOW remains one of the most sufficient methods to utilize in document classification - more precisely in text classification (Koncz and Paralic, 2011).

To implement word frequency detection in a document, a small vector can be used to form a compact and meaningful representation of that document (Koncz and Paralic, 2011). This new representation will be utilized for Data Visualization and information analysis (Hanson, Jiang, and Dahl, 2019). As a core method in Deep Learning, Autoencoders have been the subject of previous research work since 1987 (Makhzani et al., 2015).

In this research, Autoencoders are used as a method that reconstruct models based on input features (Makhzani et al., 2015). We promptly implemented a Neural Network of Autoencoder. This Neural Network of Autoencoders makes use of hidden layers that learned textual characteristics (Luo et al., 2018). Apart from the fact that it is occasionally difficult to obtain good reconstructions with simple Autoencoders (Lin et al., 2019), we have indirectly attempted to enhance feature learning processes in terms of regularization to prevent them from over-fitting (Xu, Gao, and Callan, 2018). An Autoencoder is made of two parts (see Figure 3.4)- the *encoder* Φ and the *decoder* Ψ :

$$\Phi : X \Rightarrow Z \tag{3.1}$$

$$\Psi : Z \Rightarrow X \tag{3.2}$$

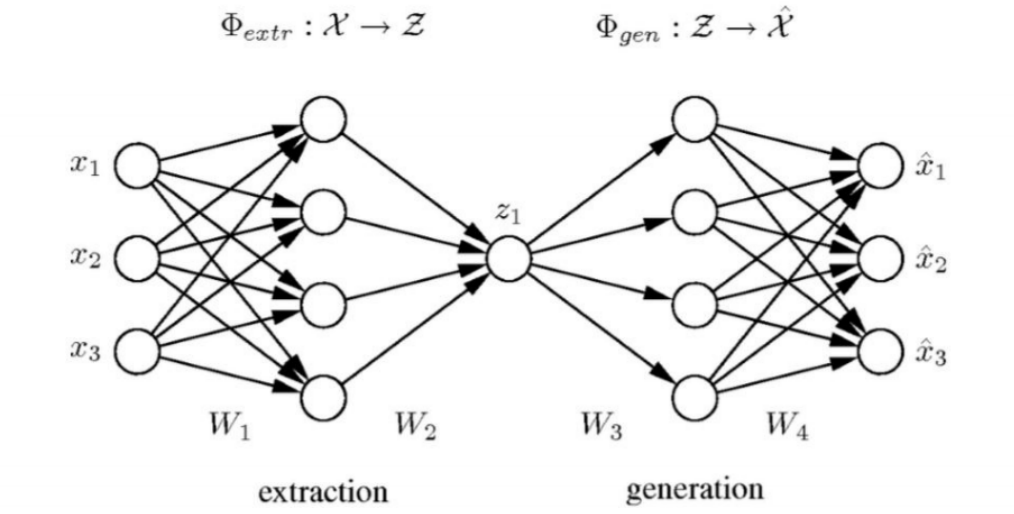


FIGURE 3.4: Autoencoders.

Generally, with less *hidden layers*, the encoding process computes input features $x \in \mathbb{R}^d = X$ and juxtaposes it to them $Z \in \mathbb{R}^p = F : Z = \sigma(W_x + b)\ Z$. Here, σ is the *sigmoid* function, W is a matrix *weight*, and b is a *bias* vector.

$$\hat{x} = \hat{\sigma}(\hat{W}Z + \hat{b}) \tag{3.3}$$

In Equation 3.3 $\hat{\sigma}$, $\hat{W}$ and $\hat{b}$ may differ in general from σ, W, b. Autoencoders are *unsupervised learning* models as discussed in the literature review (Chapter 2). Figure 3.5 depicts a scenario in which two phases, namely training and testing are required for the classification algorithm to perform prediction (Agichtein, Brill, and Dumais, 2006). Basically, the classification algorithm handles label data during the training process - the feature extractor component extract relevant information from the input dataset. Similarly, the ML algorithm build a classifier model used for prediction (Ingre and Yadav, 2015). This model performs a second Feature Extraction process useful for the algorithm in term of prediction Accuracy.

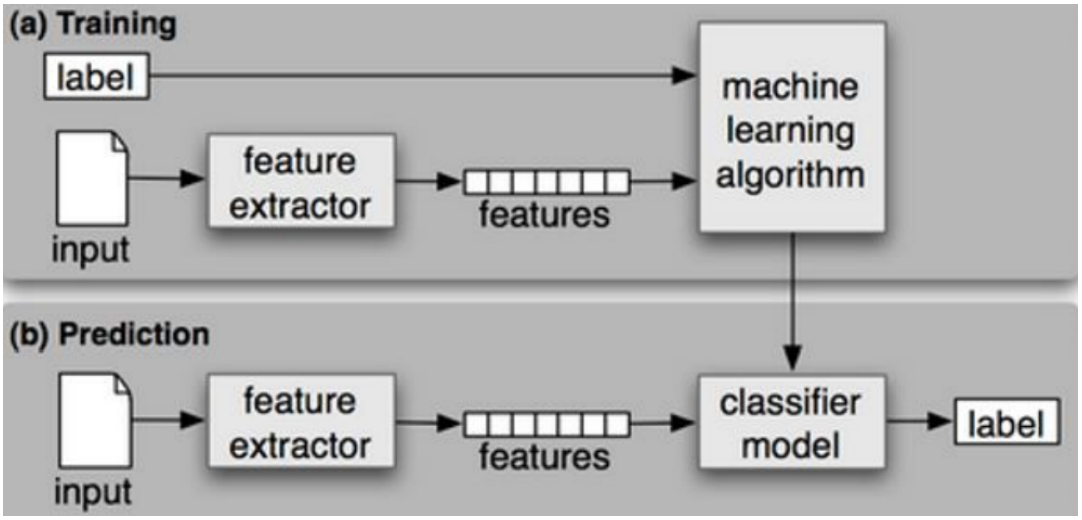


FIGURE 3.5: Supervised Autoencoding.

3.2.1 Neural Networks

Neural networks are generally regarded as being a major segment of the *supervised* learning discipline. As with all types of classification in supervised learning, the goal of a neural network is to utilize labeled data in order to *train* the network to be able to classify any new instances of data that may come in as either being *expected* or *unexpected* (Welbl, Stenetorp, and Riedel, 2018).

This labelled data consists of a vector X, corresponding to the various features of a particular object or environment and their values (Welbl, Stenetorp, and Riedel, 2018). A vector Y is also included, which, in a *binary* classification problem, classifies each *observation* as either expected or unexpected, based on the values of the vector X at that position (Welbl, Stenetorp, and Riedel, 2018).

An example of such a problem would be in $text - classification$, where positive/ negative and neutral observations would be classified differently in the Y vector, with different *combinations* of features in the X vector (Michalski, Carbonell, and Mitchell, 2013). After learning a *parameter* vector, W, which multiplies with X to minimize the *error* of classification with already-labeled data, the program would then be able to create labels for new data, and thus determines whether new instances of textual data are positive/ negative or neutral (Michalski, Carbonell, and Mitchell, 2013).

Features are input to the system as a layer of *nodes* (see Figure 3.6).

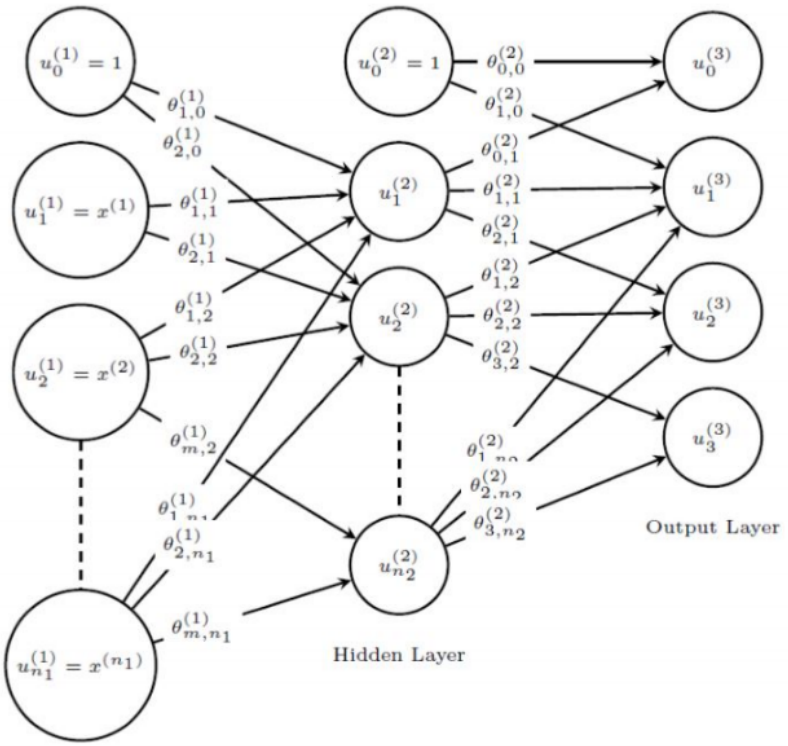


FIGURE 3.6: Neural networks structure.

In order to deal with non-linear relationships between data, values obtained in the input layer of nodes are then converted using a set of parameters w_1 into a new

set of features, known as the hidden layer (Michalski, Carbonell, and Mitchell, 2013). This process can be repeated for a number of hidden layers, however, networks with one or two hidden layers are most commonly used for their lower computation times (Haixiang et al., 2017; Michalski, Carbonell, and Mitchell, 2013). Finally, at the end of the hidden layers, the features are once again converted into a final output value, which classifies the observation as being either positive/ negative or neutral (Haixiang et al., 2017). On the first run-through of this algorithm, randomized parameters are used, which are unlikely to give accurate results (Michalski, Carbonell, and Mitchell, 2013). Thus, through a process known as $back-propagation$, labeled training data can be used to iteratively update each set of parameters, until the algorithm has a reliable set in order to make accurate classifications (Haixiang et al., 2017). To $update$ the parameters, a $cost-function$ is developed, which is based on the *mean* error of the observation in relation to the current classification function (Michalski, Carbonell, and Mitchell, 2013). Hence, the algorithm is progressively updated until it is deemed to be suitable to tackle any new data of the same nature given to it (see Figure 3.7).

Algorithm 1 Artificial Neural Network (Network Propagation)

Require: Randomly Initialize Network Parameters ▷ $\Theta_{d,s}^{(l)} := \Theta_{d,s}^{(l)} - \alpha * \frac{\partial J(\Theta)}{\partial \Theta_{d,s}^{(l)}}$

1: **while** != Converged **do**

2: ▷ Repeat Until Convergence $\Theta_j := \Theta_j - \alpha * \frac{1}{m} \sum_{i=1}^{m} (h\Theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

Training Sample $h_{w,b}^{(x)}$

3: Apply-Forward Propagation ▷ $W_1 * x_1 + W_n * x_n$

4: Update-Network-parameters-via-Back-Propagation

5: ▷ $E = \frac{1}{2N} * \sum_{i=1}^{N} ||(y(x) - \hat{y}(x))||^2$

6: **end while**

FIGURE 3.7: Neural networks Algorithm.

The value of the $cost-function$ eventually converges to a minimum, which is when the $learning-process$ will be complete (Haixiang et al., 2017). In this research, the neural network algorithm is implemented by *propagating* the *activation function* $g(S)$ values from the input layer to the output layer (Haixiang et al., 2017). In Equation 3.6, $u_i^{(t)}$ represents the activation value of i^{th} node of the t^{th} layer and in Equation 3.8, $g(S)$ represents the activation function (Michalski, Carbonell, and Mitchell, 2013). θ is a specific hypothesis function parameter with a numerical value:

$$u_0^{(t)} = 1, t \in [1, 2] \tag{3.4}$$

$$u_i^{(1)} = X^{(i)} \tag{3.5}$$

$$u_i^{(t)} = g(S_i^{(t)}), t \in [2, 3] \tag{3.6}$$

where

$$S_i^{(t)} = \sum_{j \in n_{(t-1)}} uj^{(t-1)} * \theta_{j,i}^{(t-1)} \tag{3.7}$$

and

$$g(S) = \frac{1}{1+\epsilon^{-S}} \tag{3.8}$$

Consider a $d-input$ $(x \in \mathbb{R}^d)$, $c-output$ $(\hat{y} \in \mathbb{R}^c)$ neural network that uses an augmented input vector $(\hat{x} \in \mathbb{R}^{d+1})$ and activation function g to compute the output. Let $\Theta_{i,j}$ denote the weight between input node $i \in \{1,...,d+1\}$ and output node $j \in \{1,...,c\}$. Suppose that we are given a training set $S = \{(x^p, y^p, p = 1,...,N)\}$ where $\in$ R^c denote the output of the network for augmented input $\hat{x}^p$. If the activation function g is a $differentiable$ function, we show how the gradient descent method applied to the $cost-function$:

$$J = \frac{1}{2N}\sum_{p=1}^{N}\sum_{j=1}^{c}(y_j^p - \hat{y}_j^p)^2 \tag{3.9}$$

may be used to obtain a $batch-training-rule$ for the $weights$ of the network.

$$= \frac{1}{2N}\sum_{p=1}^{N}\sum_{j=1}^{c}(y_j^p - g(\sum_{i=1}^{d+1}\theta_{i,j} - x_i^{(p)}))^2 \tag{3.10}$$

$$= \frac{\partial J}{\partial \theta_{i,j}} = \frac{1}{N}\sum_{p=1}^{N}\frac{\partial}{\partial \theta_{i,j}} g(\sum_{i=1}^{d+1}\theta_{i,j} - x_i^{(p)}(y_j^p - g(\sum_{i=1}^{d+1}\theta_{i,j} - x_i^{(p)}))) \tag{3.11}$$

$$\theta_{i,j} := \theta_{i,j} - \alpha\frac{1}{N}\sum_{p=1}^{N}\frac{\partial}{\partial \theta_{i,j}} g(\sum_{i=1}^{d+1}\theta_{i,j} - x_i^{(p)}) - x_i^{(p)}(y_j^p - g(\sum_{i=1}^{d+1}\theta_{i,j} - x_i^{(p)})) \tag{3.12}$$

3.3 Named Entity Recognition (NER)

NER has segmented entities - it has sought to locate and categorize different entities such as names of people, organizations, locations, time, and percentages (see Figure 3.8). In our research, NER detected and classified company/ brand names within textual content (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). Figure 3.9 portrays the NER distribution curves of negative N and positive features P - the decision threshold of NER is represented by a dotted line (Webster, 2001). The misclassification of some features are represented by FN and FP (Nicola, 2013).

3.3.1 Our NER evaluation

In our evaluation of the NER, entities are defined using two attributes (Lin et al., 2019). These attributes are represented by the name of the entity and the mark assigned by the NER module (Schwaiger, 2004). Table 3.1 explains our evaluation of the NER. We also use statistical methods for the NER by modeling the distribution probability $p(y|x)$ (Schwaiger, 2004). Where y represent the sequence of words (Schwaiger, 2004).

3.3.2 Feature Extraction (FE)

In Data Science, accurate FE methods have key roles for identifying relevant instances and augmenting classification Accuracy (Koncz and Paralic, 2011). ML based FE techniques require a few parameters to adjust (Lu and Zhai, 2008). The drawback of clustering is that it is difficult to extract minor features (Lu and Zhai, 2008).

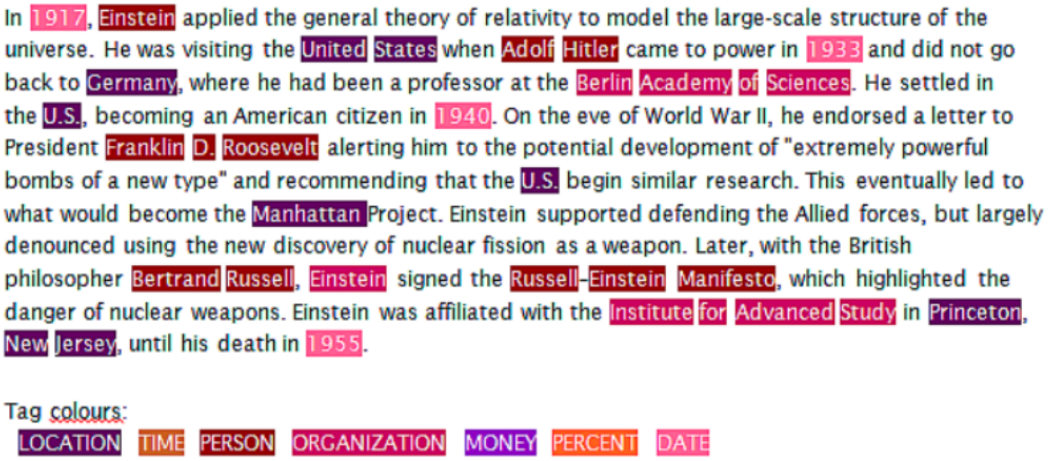


FIGURE 3.8: NER.

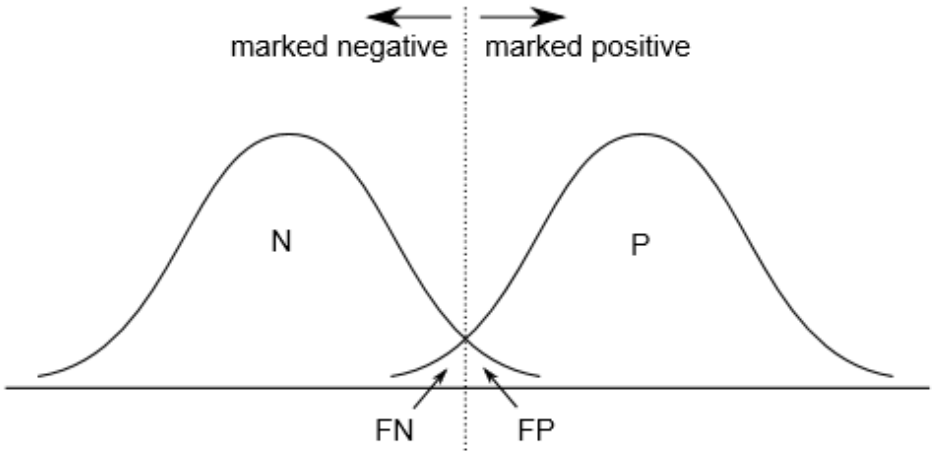


FIGURE 3.9: NER metrics.

Feature Extraction is often used in tasks corresponding to textual classification, notably in the classification of documents and the analysis of feelings (Lu and Zhai, 2008). For example, when we want to classify documents into different clusters, we generally assume that the words to be detected in one category of document is not in another category of document. This means that the frequency of distribution of words or sentences is able to characterize different categories of documents (Lu and Zhai, 2008).

In text-based applications, since textual elements are typically input data, the Feature Extraction process is very important for creating entities involving word frequencies (Lu and Zhai, 2008). To accomplish this task, a technique named features hashing has been implemented in order to effectively transform textual entities into index (Kent, 2014). In ML, feature-hashing is a feature *vectorization* technique (Li et al., 2019). Instead of associating each textual observation (words/ sentences) with

Entity	Marked as
Wits	University
South Africa	Country
United States Congress	Congress
IEEE	Organisation
1:PM	Time

TABLE 3.1: Entity marked.

a specific index, this method works by using a hash function on the entities and automatically using their hash values as indices (Li et al., 2019). The common method is to model a representation of the dictionary vocabulary $X(l)$ of the matrix $2 * 2$ at the training time (Lu and Zhai, 2008; Kent, 2014).

3.3.3 Feature Selection (FS)

The title of a page contains the keywords that describe the page itself (Shekyan et al., 2018) - following this, we implemented an approach which collects the header of a web page, ignores irrelevant features, categorizes terms based on their occurrence in a web page and output words with high occurrences (Nkongolo, 2018). Our framework comprises three stages (Data Collection, Data Mining, and Data Visualisation). We discuss these concepts in the following three Sections.

3.4 Information Retrieval (IR)

To access textual content of the World Wide Web (WWW) in an automated manner, we used a web crawler (Rawat and Malhan, 2019). The process implemented by this web crawler has the ability to automatically download a large amount of textual data (Kumar and Kumar Gupta, 2019). Many search engines make use of it to provide up-to-date information on the number of websites that are added to the WWW (Kumar and Kumar Gupta, 2019). We primarily chose the web crawler to make the processes of Data Collection automatic (Kumar et al., 2018). This was due to the fact that a manual data retrieval on the web is time-consuming (Kumar and Kumar Gupta, 2019; Hanson, Jiang, and Dahl, 2019). However, a web crawler has been used to easily locate and collect full or partial content from public websites (Figure 3.10).

Cho and Garcia-Molina (2002) presented in Figure 3.10 an hybrid architecture that includes more than one crawler. The user interacts with GUI by providing the query - this query is then processed by an Application Server Unit that utilises parallel crawling (if one crawler failed to retrieve content, this does not affect the other crawler to do so) (Cho and Garcia-Molina, 2002). The Data Master Unit stores query results and uses Crawler Agents to retrieve from the web page information pertaining to the user query - the query results are then display on the GUI (Cho and Garcia-Molina, 2002).

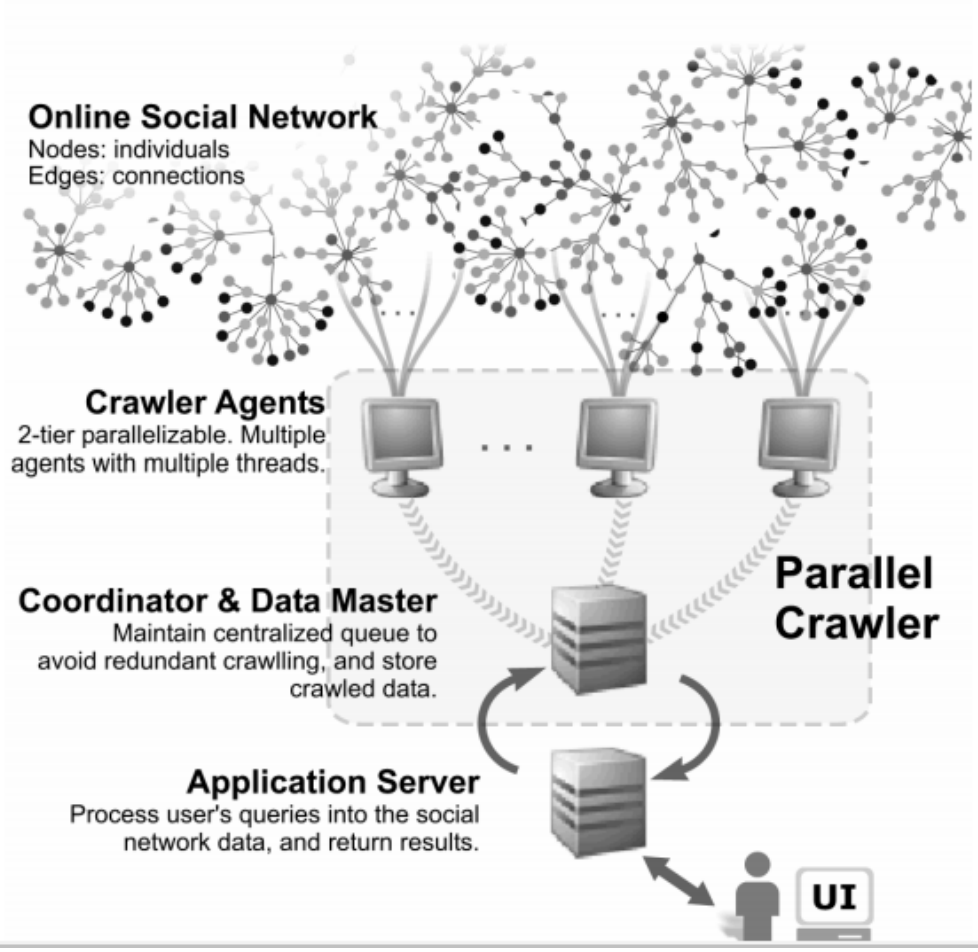


FIGURE 3.10: Web crawling (Cho and Garcia-Molina, 2002)

3.4.1 Feature Visualization

We present details on the visualization step of the Brand Reputation framework in this Section 3.4.1. Each visualization category is developed to specify a certain aspect of the Data Mining process (Figure 3.11).

3.4.2 Data Mining

The exploitation of textual data refers to SA. Methods of exploiting textual data are thus implemented (Gasc et al., 2019). First and foremost, we implemented sentimental classification algorithms (Duggan et al., 2015). These algorithms used a variety of class labels (positive, negative, and neutral), (Confente et al., 2019). The Data Mining process refers to extracting useful information from textual data set provided by the web crawler or existing data set.

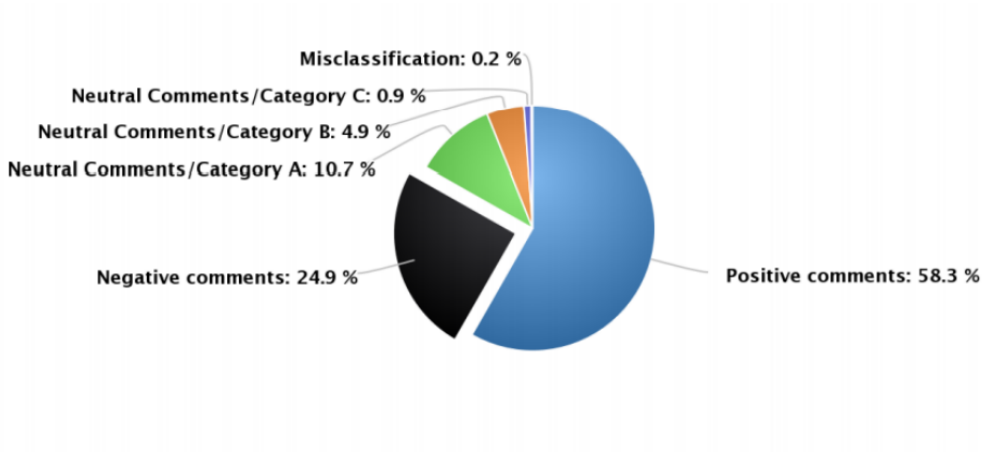


FIGURE 3.11: Visualisation Unit.

3.5 Evaluating the System

The system performance relies on sentiment analysis (Vural et al., 2013). For *binary* classifiers, there are different *metrics* to measure the performance and the testing phase can be split up into the following phases (Wang et al., 2012):

3.5.1 Evaluating Coverage

The *coverage* of the system refers to the percentage of the total observations that the system is able to make a classification for (Gonçalves et al., 2013). A high coverage value is desired to make sure that every feature used by the system is able to have classification generated for them (Gonçalves et al., 2013). The *formula* for calculating the coverage is shown below (Gonçalves et al., 2013):

$$Precision = \frac{TP}{TP + FP} \tag{3.13}$$

By using this metric, we are able to determine the reliability of the system in terms of making a classification (Fu et al., 2012).

3.5.2 Evaluating Accuracy

In order to evaluate the *Accuracy*, we used the metrics of *Precision* and *Recall*. We need a *baseline* to use as a reference (*groundtruth*). As false/ true negatives/ positives relate to the predictions, we must first establish a set of $data - points$ whose classification is known ahead of time (Nkongolo, 2018). The predicted outcomes are then compared to the known cases. The false positive is the case where a model predicts something that is known not to be so (Nkongolo, 2018). The false negative is the case where a model predicts something not to be so when the known classification is that it is so (Nkongolo, 2018). *Precision* calculates how many classifications generated were consistent. *Recall* compares the number of correct classifications generated to the total number that could have been generated based on the data set (Calefato et al., 2018). Both a high *Coverage* and a high *Recall* are desirable for the classification system. The *formulae* for calculating the *Recall* is shown below (Gonçalves et al., 2013):

$$Recall = \frac{TP}{TP + FN} \tag{3.14}$$

In Equation 3.14, True Positive stands for TP and False Positive stands for FP. False Negative is represented by FN (Lin et al., 2018). High $Recall$ means that the classifier can predict most of the Web pages accurately (Lin et al., 2018).

3.5.3 F-Measure

$F - measure$, also called harmonic means, is the combination of $Precision$ and $Recall$ that is also called balanced $F - score$ and is calculated as follows (Thelwall, Buckley, and Paltoglou, 2012):

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.15}$$

In Equation 3.15, $Precision$ and $Recall$ are weighted evenly (Nkongolo, 2018).

3.5.4 Accuracy

Another *statistical* metric of how well a binary classifier performs correctly on test data is $Accuracy$ (Thelwall, Buckley, and Paltoglou, 2012). To calculate the Accuracy, both TP and TN values amongst the total number of examined cases are considered (Lu and Zhai, 2008). The Accuracy formula is :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.16}$$

Initially, we have a classification algorithm $f(x|\Psi)$. After training, the prediction of x represents the positive polarity if $f(x|\psi) \geq \Theta$, at a certain level Θ (Massucci and Docampo, 2019). We suggest that $f(x|\Psi) \in [0,1]$ embodies *probabilistic* space if x has a positive polarity, this implies that $P(+|x) \equiv f(x|\Psi)$ (Michalski, Carbonell, and Mitchell, 2013). We say that x has a negative polarity if $f(x|\Psi) < \Theta$ and $P(-|x) \equiv 1 - f(x|\Psi)$. Then, based on the true label of x, we consider four cases in our $confusion\ matrix$ (Moraes, Valiati, and Neto, 2013). We also count the number of their occurrence: TP , FN , FP , and TN. For neural networks, we focused mainly on the classification of positive and negative polarity using a $binary - classification$ (Michalski, Carbonell, and Mitchell, 2013). In this case, the FN and FP represents a $misclassification$ (Massucci and Docampo, 2019).

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	TP	FP
	Negative (Predicted)	FN	TN

TABLE 3.2: Binary Confusion Matrix.

The words positive and negative refer to the polarity of sentiment, or more general, classifier's prediction or expectations (Kent, 2014). The terms *true* and $false$ show whether the classifier prediction or expectation corresponds to real observations or not, thus :

1. TP is the number of positive examples that are predicted as positive;

2. FP is the number of negative examples that are labeled as positive;
3. TN is the number of negative examples that are classified as negative;
4. FN is the number of positive examples that are tagged as negative.

3.5.5 Empirical error

The $empirical - error$ calculates the Accuracy of the ANN in terms of prediction outcome values based on previous unseen data (Jamal et al., 2018). The formula to compute the empirical error is shown in Equation 3.17:

$$Error = \frac{FP + FN}{FP + FN + TP + TN} \tag{3.17}$$

3.6 Sentiment Analysis Program

The open source *Sentiment Analysis* program is explained in this section. This program uses a *Lexical* method to *categorize* texts of *social networks* (Vural et al., 2013). This program is also available as a *Python* version. In this study, we used an $interface$ website to test *Sentistrength* in *real time*. The list of *emoticons* and the *lexicon* of sentiments are the *Sentistrength* resources which have been stored in a DB and the algorithm $refers$ to the *dictionary* when it is *loaded* and *processes* the text (Redhu et al., 2018). This *lexicon* is composed of 2.310 sentiments of human labeled words. The *polarity* scores is generated by the algorithm for each specific text (between minus one and minus five for negativity), one to five for positivity and the neutrality score is between negativity and positivity ranges (Vural et al., 2013; Zheng, Wang, and Gao, 2018). A *matching* process is then applied whereby each word $w_1, w_2, ..., w_n$ is matched in the dictionary D by assigning the appropriate polarity score with one of these three ranges (this score has been assigned manually) (Islam and Zibran, 2018).

When the algorithm is invoked, it reads texts, segments it into words, fragments punctuation and emoticons. A $verification$ process checks each word against the sentiment score (Islam and Zibran, 2018). The *Sentistrength* algorithm drawback is that it does not utilize *grammatical POS* to differentiate senses of the different words (Islam and Zibran, 2018). This is due to the fact that, the algorithm is designed to handle $informal$ social web text (Vural et al., 2013).

The algorithm optimizes its *lexicon* term *weights* w for specific texts and *iteratively increments* or *decrements* the w by one, one term at a time until no term *strength* changes to enhance the overall classification *Accuracy* (it is a *hill climbing* algorithm - see Figure 3.12) (Arif-Uz-Zaman et al., 2017). *Sentistrength* is *supervised* with *training* data-sets, *unsupervised* without a *training data − set* and *semi-supervised* (Nasukawa and Yi, 2003). In Vural et al. (2013) supervised mode overall Accuracy is similar to the unsupervised mode (Wang and Fikis, 2019).

In this research, the algorithm was evaluated by comparing the *Sentistrength* scores with *human* scores by using a ten fold cross validation (Lin et al., 2018). A single set of labelled text is split into testing parts (10% of texts) and training parts (90% of texts).

Input:= T
Output:= *Polarity*
Require: Textual Data T & The Sentiment Lexicon L ▷ T & L
Output:= s_i^+, s_i^-, s_i
Initialized S Where:
$s_i^+ \in L$
$s_i^- \in L$
$s_i \in L$
$t_i \in T$ ▷ t_i:=Text
Search For $t_i \in L$
if $t_i \in s_i^+ - List$ **then**
  $t_i = s_i^+$
  if $t_i \in s_i^- - List$ **then**
    $t_i = s_i^-$
    if $t_i \in s_i - List$ **then**
      $t_i = s_i$
    end if
  end if
end if
▷ Compute polarities for Web Pages W_p ▷ $\sum_{p=1}^{N} W_p$ ▷ $p := [1,2,3,4,...,N]$
if $|\sum_{i=1}^{N} s_i^+ > \sum_{i=1}^{N} s_i^- \in W_p|$ **then**
  $W_p = s_i^+$
  if $|\sum_{i=1}^{N} s_i^+ < \sum_{i=1}^{N} s_i^- \in W_p|$ **then**
    $W_p = s_i^-$
    if $|\sum_{i=1}^{N} s_i > \sum_{i=1}^{N} s_i^- \in W_p|$ **then**
      $W_p = si$
      if $|\sum_{i=1}^{N} s_i > \sum_{i=1}^{N} s_i^+ \in W_p|$ **then**
        $W_p = s_i$
      end if
    end if
  end if
end if

FIGURE 3.12: Sentistrength algorithm.

3.7 Framework architecture

The following Figure 3.13 illustrates the automation of brand reputation on the internet. To create the data set used for this research, a text was extracted from different links provided by the crawler (Fu et al., 2012). The crawler then populated the database with relevant information (Moraes, Valiati, and Neto, 2013). Prior to storing the textual data into the DB, a Named Entity Recognition assessed the validity of the data (Stuart and Majewski, 2015). This Named Entity Recognition (corporate/ brand name recognition) was able to detect texts which include a corporate/ brand name using keywords specified in the query (Farag, Lee, and Fox, 2018). This phase is related to NLP and ML (see Figure 3.13).

The DB contains four components separated into four tables: Label, sentiments, crawler, and raw text. We then extract observations, including keywords, then classify each observation using classifiers (Seth et al., 2019; Kim, Youn, and Lee, 2019;

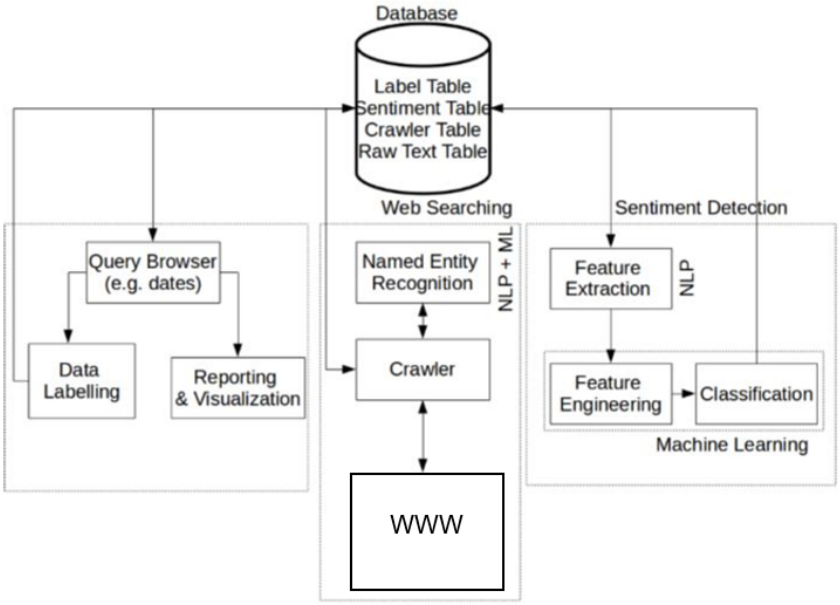


FIGURE 3.13: Platform hierarchy.

Hong, Lee, and Yu, 2019). The classification of extracted observations (Sentences) focused on their polarity (positive, negative, neutral) (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). This was achieved through iterative Feature Extraction. To perform the Sentiment Detection, Feature Extraction used NLP techniques such as N-gram and bag-of-words (Young et al., 2018).

The Feature Extraction utilized the auto-encoders approach by reconstructing a model of the extracted observations based on the keywords (Guggilla et al., 2018). This model was then utilized by the classifier. In this section, we considered the process of converting words into vectors (Giammarco and Giles, 2018). There are several methods for doing so. We considered a simplistic implementation, proposed by Farag, Lee, and Fox (2018), which uses auto-encoders to jointly learn representations for words and phrases. ANN and Sentistrength [1] was used to classify extracted observations/ sentences (Vural et al., 2013). Adhering to this, the results of the classification step were displayed using a Web-based interface (Donneau-Golencer et al., 2019).

We opted for Artificial Neural Networks because it produces accurate results via its confusion matrix (Moraes, Valiati, and Neto, 2013).

Sentistrength takes into account the morphology of textual data and produces a polarity close to that which a human-being could perform (Islam and Zibran, 2018). However, it has been noticed that this algorithm is efficient with sentences but inefficient with a panoply of data (Vural et al., 2013).

[1] Sentistrength is an algorithm for analyzing opinions. Taking as input any text written in English, the algorithm produces a score representing a sentimental polarity (positive/negative and neutral).

3.8 Data processing

As illustrated in Figure 3.14, datasets are given to ANN, Sentistrength, and BOW algorithms. These algorithms process data using different metrics :

1. Confusion matrix, Precision, Recall, Empirical error, F-measure and Accuracy are used by ANN,
2. Compound is used for Sentistrength,
3. Word frequency/ occurrences is utilised by the BOW algorithm.

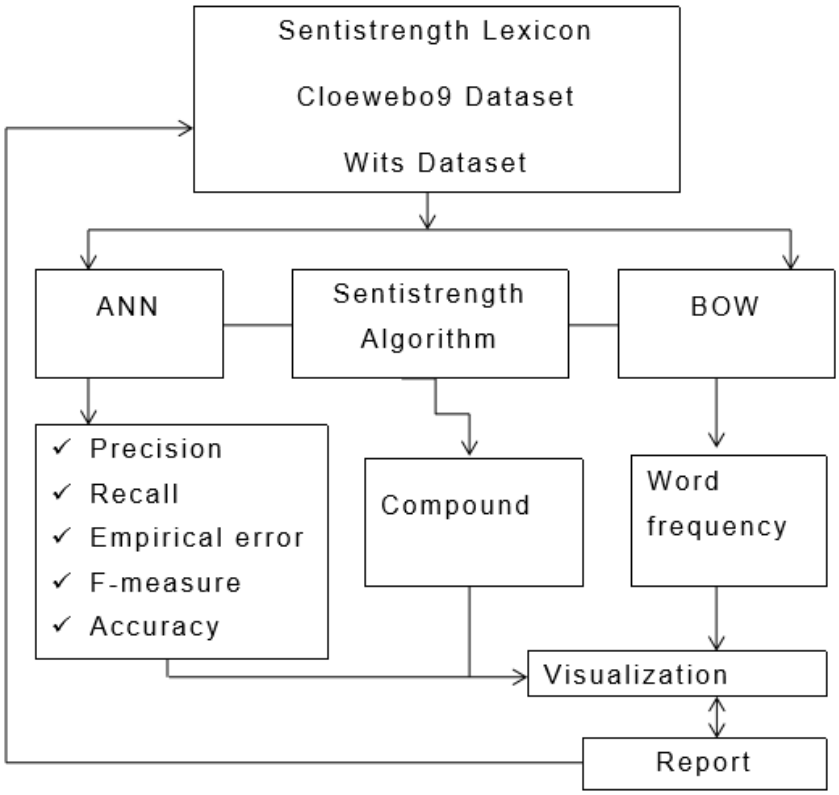


FIGURE 3.14: Data processing architecture.

Following this, experiment results are visualised for a better understanding of Data Analytic process (Chiranjeevi, Santosh, and Vishnuvardhan, 2019). Lastly, we report the experimental results and repeat the same process iteratively until we visualise and report results of the three data sets using selected algorithms. Basically, the procedural reading of the diagram illustrated in Figure 3.14 contains the natural contents of the existing datasets. We used a diverse set of metrics to evaluate selected algorithms (Jamal et al., 2018; Michalski, Carbonell, and Mitchell, 2013).

The choice of metrics affects how the performance of ML algorithms is measured and compared (Nikam, 2015). They influence how we clutter the importance of different characteristics and the ultimate choice of the algorithm to use therefore this leads to the natural evaluation measure of the algorithm, which should ideally correspond to the natural evaluation measure of the CRA framework (Das, Govardhan,

and Lakshmi, 2019). Rudimentary measures like Accuracy, Recall, and Precision consider only positive or negative decisions, with equal weight per instance (Hariri, Emami, and Malek, 2015). It is thus possible to average Precision and Recall, using an arithmetic, geometric or harmonic means as in Hariri, Emami, and Malek (2015).

3.9 Chapter overview

Textual data retrieval was addressed in Section 3.1 as well as SA. We then proceed with a discussion pertaining to Autoencoders that reconstruct learning models based on input features. In Section 3.2.1, ANN is explained in detail using back-propagation as well as the NER presented in Section 3.3. This research uses NER as a FE technique that provides relevant information within a textual dataset.

The information provided by the NER can be visualised as discussed in Section 3.4.1 - the visualisation is developed to analyse certain aspects or characteristics of the dataset (Morinaga et al., 2002). Evaluation metrics used in the experimentation process are introduce in Section 3.5, these include: Coverage, Accuracy, Precision, Recall, F-measure, and Empirical error. The SA algorithm is presented in Section 3.6 while Section 3.8 discusses the data processing architecture.

In the data processing stage, publicly available data are fed into the ML algorithms which process features for classification purposes. The result of the classification process is therefore visualised and analysed for CRA.

3.10 Conclusion

In this Chapter, we examined the feasibility of the Web Reputation Analysis framework in order to determine whether its implementation is justified. In order to achieve this, it was necessary to examine the reliability of the framework in terms of producing reputation that depends on the corporation. Research methods details as well as the learning algorithms were presented.

Chapter 4

Reputation Analysis framework

Before delving deeper into Reputation Analysis system, we have to start by addressing the concept of content extraction. Narwekar (2018) has provided a concise discussion on this topic, while Zheng, Wang, and Gao (2018) gives a more detailed explanation. The proposed sentiment focused crawler platform must include numerous components, to retrieve textual content from different web pages. The collection of textual content incorporates several processes such as, removing HTML tags and filtering textual content to retrieve only the relevant data from the content which is located in the body of the web pages (Narwekar, 2018).

4.1 Content Extraction

In the $cleanup - section$, we extracted the appropriate content, and the irrelevant information such as advertising and menu bars were ignored from the main content. To perform such an extraction and such a filtering, we used another tool named *BoilerPipe* (Narwekar, 2018; Nasukawa and Yi, 2003). *BoilerPipe* enabled the framework to retrieve the content located in the body of the web pages. It offers five options of the textual collection, among the five options CanolaExtractor is considered as the most efficient option because it has surpassed the extraction performance of the other four functions with regards to larger numbers of pages (see Table 4.1).

BoilerPipe Options	Size of features extracted
KeepEverythingExtractor	600 kB
ArticleSentencesExtractor	650 kB
NumWordsRulesExtractor	567 kB
CanolaExtractor	**1 MB**
LargestContentExtractor	700 kB

TABLE 4.1: BoilerPipe Extraction Performance

The textual extraction of a web page by *CanolaExtractor* from *BoilerPipe* is presented in Figure 4.1 - The visualisation of CanolaExtractor performance is also given in Figure 4.2.

4.1.1 Word Extraction

We also incorporate the $Stanford - Core - NLP$ library which is used extensively for textual segmentation from the content of the extracted page. The Stanford Core NLP is a set of Natural Language Processing tools, this library can provide all basic forms of words, the aggregation of speech and the structural tag of texts from a raw English text entry (Gardner et al., 2018).


```
The Apprentice
Read more about these links.
Mark Wright
Lord Sugar chose Mark Wright as the winner of series 10. Upon winning The Apprentice and
considering the prospect of going into business with Lord Sugar, Mark said "I am just
absolutely stoked and quite emotional." He added "I've worked so hard and to be sat here is
life changing and the best thing to ever happen to me."
About Mark
Australian-born Mark is a Sales Manager supporting internet marketing consultants and he has
received repeated recognition for the highest revenue growth within his company. He has over
nine years' experience in selling within the digital, health and fitness industries, though
his first job was washing cars at his dad's business.
Ambitious Mark would describe himself as a great communicator and manager of people and
claims he has no bad business skills, although he confesses that some may find him arrogant
as he appears to find it hard to accept others' opinions. He can also be grumpy in the
mornings! He says he is inspired in business by John D. Rockefeller, Andrew Carnegie and J.P.
Morgan as they are entrepreneurs that built their own empires.
```

FIGURE 4.1: A sample of textual data extracted by BoilerPipe using CanolaExtractor.

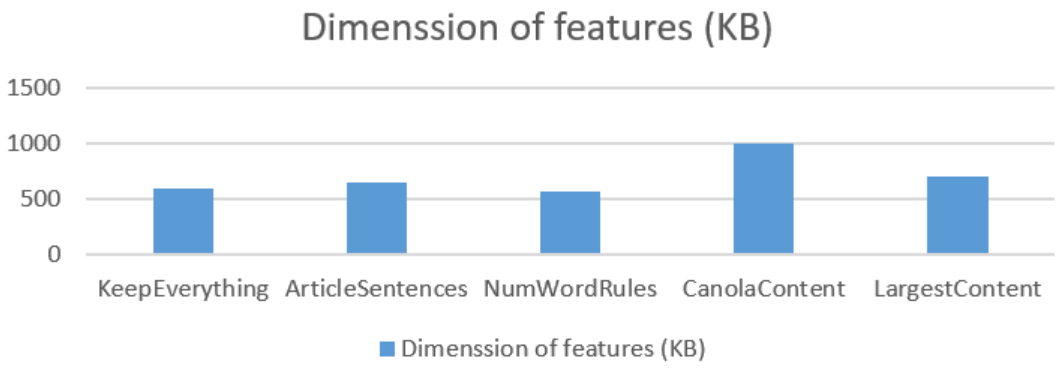


FIGURE 4.2: CanolaExtractor FE performance.

Our platform first sends the text to $Stanford - POS$ tagger for some preprocessing steps such as sentence segmentation, tokenization of sentences and tagged tokens (see Figure 4.3).

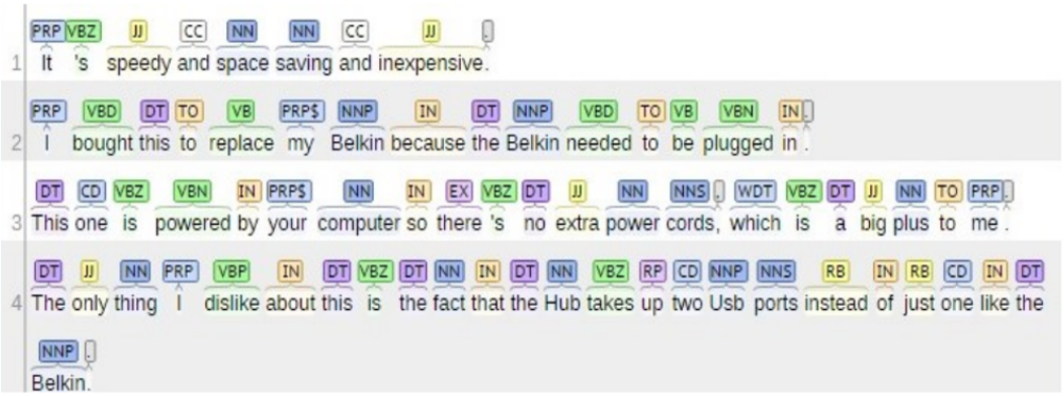


FIGURE 4.3: The pre-processing step of a web page using Stanford POS tagger.

4.1.2 Detecting Sentiment

As stated in Chapter 3, a Sentiment Detection process is required to perform the functional analysis, including the content of the page and the anchor text (Yılmazel and Arslan, 2019). To implement the latter, a Sentimental Analysis tool based on the lexicon, Sentistrength algorithm is used, as well as ANN. The interested reader should refer to the work presented in Islam and Zibran (2018) for a more detailed discussion in this domain.

We then proposed the use of an alternative which took into account a lexical subset of adjectives and adverbs (Taboada, 2016). A supervised learning algorithm which analyzed the learning data and produced a derived function that can be used to map new examples. This subset has 500 entries in total. A limitation has emerged when it comes to applied Sentistrength and ANN. We encountered challenges based on the quantification of sentimentality of a web page's content. Our philosophy is as follows, a web page is described as a textual plethora, hence the sentimentality of this page was calculated using the sentiment analysis of each sentence (Vural et al., 2013). The sentimentality of the page was formulated using the following Equation 4.1:

$$SS_p = \frac{1}{i} * \sum_{s_i \in p} (s_i^+ - s_i^- - s_i - 3) \tag{4.1}$$

The sentimental average score (SS): this symbol of sentimentality was formulated by Rout et al. (2018) and we then adapted the Equation by introducing the neutral polarity. This metric can be defined as follows: let s_i be the aggregation of sentences in a p page for which we want to determine polarities. The sentimental score p is determined by the mean of the sentiment scores of the sentence itself (Vural et al., 2013). In order to extract the Web page $words - list$, our framework builds a $array$ including all stemmed tokens (the words) along with their frequencies and part of speech tags (see Table 4.2).

Tokens	Word	Frequency	Tag
Bought	buy	1	VBD
Speedy	speedy	1	JJ
To	to	3	TO
's	is	5	VBZ
...	...	...	...

TABLE 4.2: The Words-List

After constructing the $words - list$, all words which are tagged as verbs and adjectives are sent to the classifier in order to extract their polarity scores (Massucci and Docampo, 2019). The total scores of all positive/ negative and neutral words are then calculated. Based on the higher score, our classifier predicts the web page polarity (Mehdad, Pappu, and Stent, 2018). If the classifier can correctly predict the polarity of the web page, it will be added to the DB. Our classifier goes through all real-time/ testing data-sets and predicts the polarity of extracted sentences/ observations (Musto et al., 2015). One of the strong points of this method is the ability for further expansion by considering more out-of-domain data sets as well as adding the additional sentiment lexicons (Lukyanenko et al., 2019). $Stop - Words$ are irrelevant words in Search Queries (Wang et al., 2018). In this research, these words are ignored by search queries. Table 4.3 shows a list of English stop words ignored.

Stop Words
a, about, above, across, after, afterwards again, against, all, almost, alone, along already, also, although, always, am, among amongst, amoungst, amount, an, and, another
...

TABLE 4.3: List of English Stop Words

4.2 Machine Learning

The system that we propose in this research, performs automatic learning and builds a model which is able to perform prediction by invoking extraction functions (Guggilla et al., 2018). We implemented Autoencoders in Keras.

4.2.1 Experimental Set-up

To deploy the system, we used a huge set of analytic data from various sources. In the scientific field, Clueweb09 was created to support information retrieval research and related human language technologies (Tsai et al., 2013). We utilized Clueweb09 as a an existing dataset. This dataset is collected between January and February 2009 and could contain up to one billion web pages with 25 TB content (Tsai et al., 2013). During the experiment, the system took into account the implementation of Python 3.6.0. MySQL is used to save the textual content of different web pages. All experiments are run on a computer with Ubuntu Linux as the main operating system (OS).

4.2.2 Web crawling

One ML sentiment-focused crawler was used, in which the polarity of a target web page is predicted by an algorithm (Arif-Uz-Zaman et al., 2017). The ML model adopted sentimentality prediction techniques and was evaluated. The following Table 4.4 portrays the sentiment-focused crawler utilized.

Rank	URL	Polarity
1	www.witsfoundation...	S_i^+ - Positive
2	www.wits.ac.za.feesmustfall2016...	S_i^- - Negative
3	https://www.wits...	S_i^+ - Positive
4	www.languageschool...	S_i - Neutral
5	www.news24.com/wits students/cry...	S_i^- - Negative

TABLE 4.4: Machine Learning based Sentiment-focused crawler

4.3 Training Phase

In the training phase, the classifier took a lexicon and a web page $words - list$ as input and computes the polarity of the Web page by querying for sentimental values of all its adjectives and verbs (Zheng, Wang, and Gao, 2018; Depren, Aşkın, and Öz, 2017). Figure 4.4 exhibits the data-set utilized.

4.3.1 Emoticons

To easily classify the polarity of a text/ message, it is necessary to focus on the emotions it contains (Asghar et al., 2018). We can define emoticons as a representation of happy or sad feelings (Asghar et al., 2018). To specify the polarity of emoticons, we considered an entire group of common emoticons. Figure 4.5 shows a sample of emoticons that we used to train the classification algorithm.

4.4 Training Approach

The entire textual data set was broken down into individual observations using *Python* code, with the most commonly appearing words being identified and recorded (Nkongolo, 2018). The particular number of common words chosen can be increased or decreased depending on the size of the data-set, or if results generated are not accurate enough (Jamal et al., 2018). Thus, the logic behind this is that if an observation's polarity is to be predicted it should contain at least some of the words commonly associated with past observations polarity examined (Redhu et al., 2018).

After building this list of observations, training and testing data-sets were then built from the obtained textual data. A multidimensional array was built for the data-set which utilized the previously obtained common words as features, and each record in the data-set as a separate row in the array (Nkongolo, 2018). By reading the text file containing the observations, each observation was read in one-by-one, using the % tags to separate each observation (see Figure 4.6).

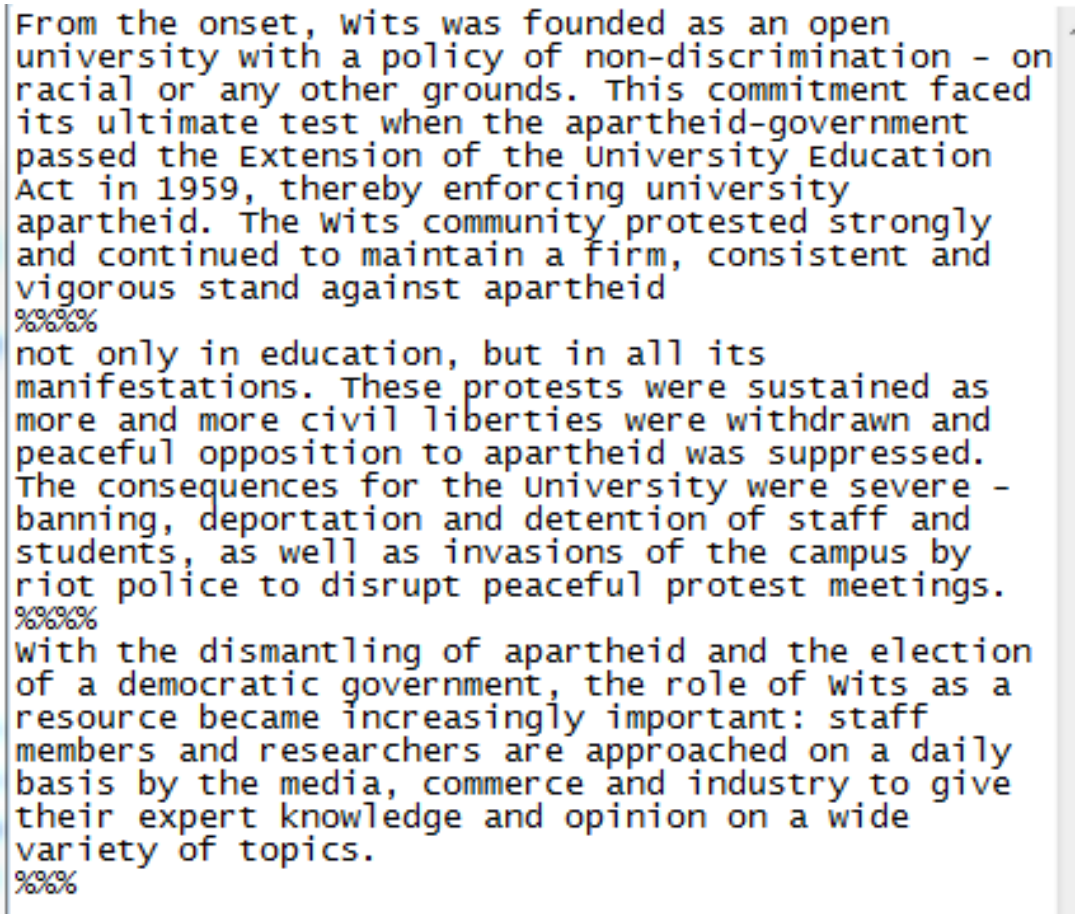


FIGURE 4.4: WITS marketing team dataset.

Each record was then compared, word-by-word, to the list of common words generated previously (Nkongolo, 2018). If any of the words in the observation matched with a common word, this was recorded in the multidimensional data-set array by *incrementing* the corresponding value in the array. Entire data-sets for the textual data were read in using this method, with a corresponding targets array being labeled "0" or "1", depending on which observation was currently being read (Nkongolo, 2018). The finalized array was then split into a training and a testing data-set, for the purposes of the neural network method. This particular system uses a four-layer total network (with two hidden layers), so three sets of random weights were first generated for the transitions between each layer (Nkongolo, 2018). Forward propagation is first applied to the network, using the following activation function (Zhang, 2018):

$$h(x) = \frac{1}{1 + \epsilon^{-W^T x}} \tag{4.2}$$

In Equation 4.2 W is the weight vector and x is the value of the input for that node (Das, Govardhan, and Lakshmi, 2019). As a result, hypotheses are generated for each node in the hidden layers (to be used as features for the next layer) as well as a hypothesis for the output in the final layer. $Back - propagation$ was then applied, in order to update the weights between the output layer and the second hidden layer, the second hidden layer, and first hidden layer, and first hidden layer

```
$:      -1.5    0.80623 [-1, -1, -1, -1, -3, -1, -3, -1, -2, -1]
%)      -0.4    1.0198  [-1, 0, -1, 0, 0, -2, -1, 2, -1, 0]
%-)     -1.5    1.43178 [-2, 0, -2, -2, -1, 2, -2, -3, -2, -3]
&-:     -0.4    1.42829 [-3, -1, 0, 0, -1, -1, -1, 2, -1, 2]
&:      -0.7    0.64031 [0, -1, -1, -1, 1, -1, -1, -1, -1, -1]
( '}{' )        1.6     0.66332 [1, 2, 2, 1, 1, 2, 2, 1, 3, 1]
(%      -0.9    0.9434  [0, 0, 1, -1, -1, -1, -2, -2, -1, -2]
('-:    2.2     1.16619 [4, 1, 4, 3, 1, 2, 3, 1, 2, 1]
(':     2.3     0.9     [1, 3, 3, 2, 2, 4, 2, 3, 1, 2]
((-:    2.1     0.53852 [2, 2, 2, 1, 2, 3, 2, 2, 3, 2]
(*      1.1     1.13578 [2, 1, 1, -1, 1, 2, 2, -1, 2, 2]
(-%     -0.7    1.26886 [-1, 2, 0, -1, -1, -2, 0, 0, -3, -1]
(-*     1.3     1.26886 [4, 1, 2, 0, 2, -1, 1, 2, 1, 1]
(-:     1.6     0.8     [2, 2, 1, 3, 1, 1, 1, 3, 1, 1]
(-:0    2.8     0.87178 [3, 2, 3, 4, 3, 2, 3, 1, 4, 3]
(-:<    -0.4    2.15407 [-3, 3, -1, -1, 2, -1, -2, 3, -3, -1]
(-:o    1.5     0.67082 [3, 1, 1, 2, 2, 2, 1, 1, 1, 1]
(-:O    1.5     0.67082 [3, 1, 1, 2, 2, 2, 1, 1, 1, 1]
(-:{    -0.1    1.57797 [-2, -3, 1, -2, 1, 1, 0, 0, 2, 1]
```

FIGURE 4.5: Labeled emoticons and their variations.

```
This is part of Wits' suite of online offerings that will be made available over th
enhance access to higher education. These include MOOCs, short online courses, and
online degree programmes
%%%%
The year 2016 has been incredible for the digital revolution at Wits. The Universit
R500 million in a new ICT system which will be implemented in the coming years.
%%%%
Wits is also transforming Braamfontein into a young, vibrant precinct for its notab
innovative community. Together with partners in government, business and industry,
launched the Tshimologong Digital Innovation Precinct in Braamfontein, Johannesburg
Setswana for "new beginnings", Tshimologong is Johannesburg's newest high-tech addr
the incubation of start-ups, the commercialisation of research and the development
digital skills for students, working professionals and unemployed youth will take p
%%%%
In Parktown, Wits has played an integral role in the establishment of the Nelson Ma
Hospital launched on Friday, 2 December 2016. The hospital will serve as a world-cl
advanced specialised children's hospital which will save lives. The University is t
partner that is located on land donated by Wits in Parktown. It will serve as a hos
training and research.
```

FIGURE 4.6: Feature separation using the % tags.

and input layer, respectively (Jamal et al., 2018). This is done to improve hypothesis generation accuracy, as mentioned by Hothorn (2018). Following this, the parameter with which to split the data into positive/ negative was obtained by testing the previously obtained factors on a combined portion of the data-set with positive/ negative labels (Nkongolo, 2018). This test was done repeatedly, each time updating the parameter value until the optimal separation is found (Seth et al., 2019).

4.5 Training Results

The implementation created for the intake and preprocessing of the data-set was found to proceed quickly enough to get results in a reasonable amount of time (Agichtein, Brill, and Dumais, 2006). This was based on the current amount of data collected for the tests. Obviously, it is expected that the more data that is added to the data-set, the longer the code will take to generate results (Al Marouf, Hasan, and Mahmud, 2019; Webster, 2001; Nikam, 2015). The ANN algorithm, when tested on the basis of TP, TN, FP, and FN, was found to produce approximately 11% FP and 2.7% FN. This is shown in Table 4.5. ANN accurately classified each observation using a binary polarity with an Empirical error of 0.1379%.

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	22 = 15.172413%	16 = 11.034482%
	Negative (Predicted)	4 = 2.7586206 %	103 = 71.034482%

TABLE 4.5: Evaluation metrics for ANN

In terms of approximations, the range of the ANN's confusion matrix is given in Table 4.6.

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	20 to 33	4 to 18
	Negative (Predicted)	4 to 16	89 to 103

TABLE 4.6: Confusion Matrix approximation

Empirical error = 0.13 to 0.17%.

As we can see from the confusion matrix approximation, there is more or less 33 positive observations that are predicted as positive; with at least 103 negative observations that are classified as negative. In contrast, there were at least 34 observations that are misclassified. ANN is an adaptive algorithm which changes its inner structure based on the information passing through it (Zhang, 2018).

Therefore, learning in ANN means that a processing unit could update its input/ outputs due to the change in environment (Haixiang et al., 2017). For training, we used some training samples with unique features (words, sentences); and to performed testing, we used some testing sample with other unique features. From the Graphs of ANN given in Figure 4.7, we note that there is a strong correlation between the volume of the data-set and the classifier prediction. The implication is that for a Big Data with multiple observations, the algorithm will decrease the value of the cost function, thus improve its learning (Stuart and Majewski, 2015).

However, if the volume of the data-set is minimized, the algorithm will increase the value of the cost function, thus prone to errors (Volcani and Fogel, 2006).

The cost function of the ANN reached a minimum, thus showing that the algorithm was able to converge correctly (Lewandowski, 2020). The graph in Figure 4.7 shows the value of the cost function decreasing with the number of iterations.

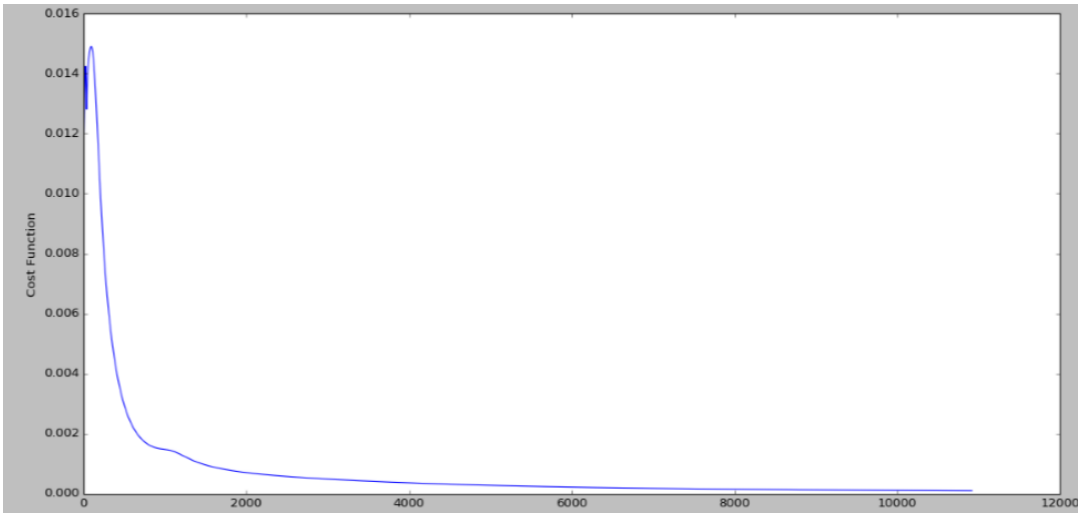


FIGURE 4.7: The value of the cost function decreasing for positive features.

4.6 Analysis

From the results obtained from the ANN implementation, it can be seen that most of the data is able to be classified correctly. This is likely due to the textual polarity data not having too much overlap which results in the boundary value being able to separate them properly (O'Leary, 2013). Due to the varied nature of the feature data, with the features potentially changing as the data-set gets larger, it can be seen that this classification method could possibly have a reduction in prediction accuracy as the sample size increases, due to more observations potentially overlapping (Nkongolo, 2018). However, a larger data-set could, instead, increase hypothesis generation accuracy as the algorithm would have more data to work with and the boundary between positivity and negativity could be better understood.

The quality of the data is thus important in this respect. In addition, for the neural networks implementation, it was found that the algorithm only misclassified a minor number of observations, with an empirical error of 0.13 or 0.17%, suggesting a correct classification rate of 71%. The vast majority of observations were thus classified correctly, based on this result. The algorithm was thus able to handle the complex relationships between the various features, making use of the hidden layers to account for these relationships (Lewandowski, 2020). It was largely successful in classifying each observation as either positive or negative. Once again, a larger data-set could potentially increase output accuracy for this algorithm (Nkongolo, 2018).

4.6.1 Bag-of-Words Results

The results of the bag-of-words is shown is Table 4.7 with the key word *student*(*s*) having the highest occurrence within the textual data-set.[1] The next Section 4.7, concentrates on detecting negation in Sentiment Analysis.

Words	Occurrence
students	128
student	104
south	96
universities	88
university	82
education	73
higher	69
protest	68
2016	63
CPUT	62

TABLE 4.7: BOW Results. The content of this WITS dataset is negative due to the occurrence of the keyword *protest*

4.7 Negation Keywords

One of the frequent ways to state an opinion is negation that completely changes the polarity of words and statements (see Table 4.8). There are varied manners and several grammatical rules to express the same opinions or emotions that a writer can use. Negation is one of the most common and yet complex issues in linguistics (Nyman et al., 2018). Negation detection can help classifiers to categorize a text, a sentence or a phrase more accurately as positive, negative or neutral by taking into account all relevant negation factors that have influence on the polarity. Therefore, identifying the negation itself and more importantly, its scope, is one of the important tasks to find out the orientation of text in sentiment detection (Taboada, 2016). Since there are different negation terms, defining negation keywords and their type is a good starting point before embarking on the next sections.

Sentence	Negative Token	Prediction
I don't understand French	**don't**	S_i^-
No one likes you	**No one**	S_i^-
They will not fail	**not, fail**	S_i^+
Students cannot protest	**cannot, protest**	S_i^+

TABLE 4.8: A typical set of labelled sentences and their polarity after detecting negation

4.7.1 Negation Scope

Konstantinova and De Sousa (2011) presented six common typical categories for negation keywords as shown in the following Table 4.9. Since we work with textual

[1] 128+104=232

data, it is common that writers use ungrammatical structures and words in their writing. Thus considering these cases is crucial to computing the correct scope of a negation term in a sentence or a phrase. For instance, using "ain't" instead of "aren't" or "isn't".

Category	Example
Adjectives and adverbs	impossible, nowhere etc.
Nouns	nothing
Conjunctions	neither ... nor etc.
Negation keywords	"no", "not"
Auxiliary verbs	cannot, shouldn't
Prepositions	"without"

TABLE 4.9: Categories for Negation keywords

4.7.2 Contextual Valence Shifters

Generally, terms that can alter or modify the semantic orientation of polarized words are called "contextual valence shifters" (Kiritchenko and Mohammad, 2018). In terms of grammar, adjectives modify nouns whereas adverbs shift the polarity of verbs or adverbs. For instance, in the sentence "I am very excited to play piano", the phrase "very excited" is more positive than "excited" alone. In contrast, in the phrase "little efficient", the term "little" decreases the intensity of the adjective "efficient" (Almatarneh and Gamallo, 2017). For instance, in the phrase "barely sufficient", the term "barely" changes the positive level of the word "sufficient". Kennedy and Inkpen (2006) use the contextual valence shifters in their term-counting method for sentiment classification of movie reviews and report the improvement in the Accuracy of the classifier. Similarly, Taboada (2016) propose a method that associates different percentages to the valence shifters instead of adding or subtracting a fixed value as presented by Kiritchenko and Mohammad (2018).

Our approach is inspired by Blanco and Moldovan (2011) that propose some heuristics to identify the negation detection role in statements. They study both the scope and focus of negation keywords. According to them, the scope is all items in a statement that are negated by individual negation marker, and focus is a part of scope that carries the most prominent negated burden in a statement (Blanco and Moldovan, 2011). Usually identifying the focus is more difficult than scope, while some linguistic aspects such as stress and intonation have influences on it (Blanco and Moldovan, 2011).

We mention here that addressing the focus is beyond the scope of this research. In this research, we expand the above approach by adding more negation bearings and patterns. In addition to "not" and its affixed form "n't", modal and auxiliary verbs, we consider more negation makers (see Table 4.10).

To better illustrate our approach we show it in an example. Consider the following sentence S :

Negation Makers
"no" "none", "nobody" "nowhere", "never" "barely", "scarcely" "hardly" "nothing" "neither...nor" "neither ... nor" "no one" "no person" "anyone" "anybody" "without", "lack" "lacked" "lacking","lacks
...

TABLE 4.10: Negation patterns

Tokenized *S*: [there, is, no, best, music, in, this, album, !]
In the second step, the methods look for negation keyword, and if it finds anyone, then applies a negation analyzer. In this example, the word "no" is a negation marker. The negation analyzer sends the tokenized sentence to Stanford Parser in order to create its parse tree.

4.7.3 Main Tree

(**ROOT** (S (NP (EX there)) (VP (VBZ is) (NP (NP (DT no) (JJ best) (NN music)) (PP (IN in) (NP (DT this) (NN album))))) (. !)))

In the third phase, the negation analyzer uses a negation processor to detect the corresponding pattern that matches with the parse tree (see Figure 4.8). It finds the following pattern

Pattern: VP « NP=np : (=np < (DT < no))

In the fourth step, based on the negation pattern, the negated sub-tree (s) are extracted from the main parse tree. Like Blanco and Moldovan (2011), we prune the negated sub-tree by eliminating prepositional phrases (PP) and moreover parentheses. All the rest of the leaves are marked as negated items and their polarity scores multiplied by -1.

4.7.4 Negated sub-tree

(VP (VBZ is) (NP (NP (DT no) (JJ best) (NN music)) (PP (IN in) (NP (DT this) (NN album)))))

4.7.5 Negated items

[is/VBZ, best/JJ, music/NN, in/IN, this/DT, album/NN]

We mention again that we just consider verbs and adjectives as polar terms. So the final negated terms are "is"and"best". In Table 4.11, we presented the complete list of tags with their meanings.

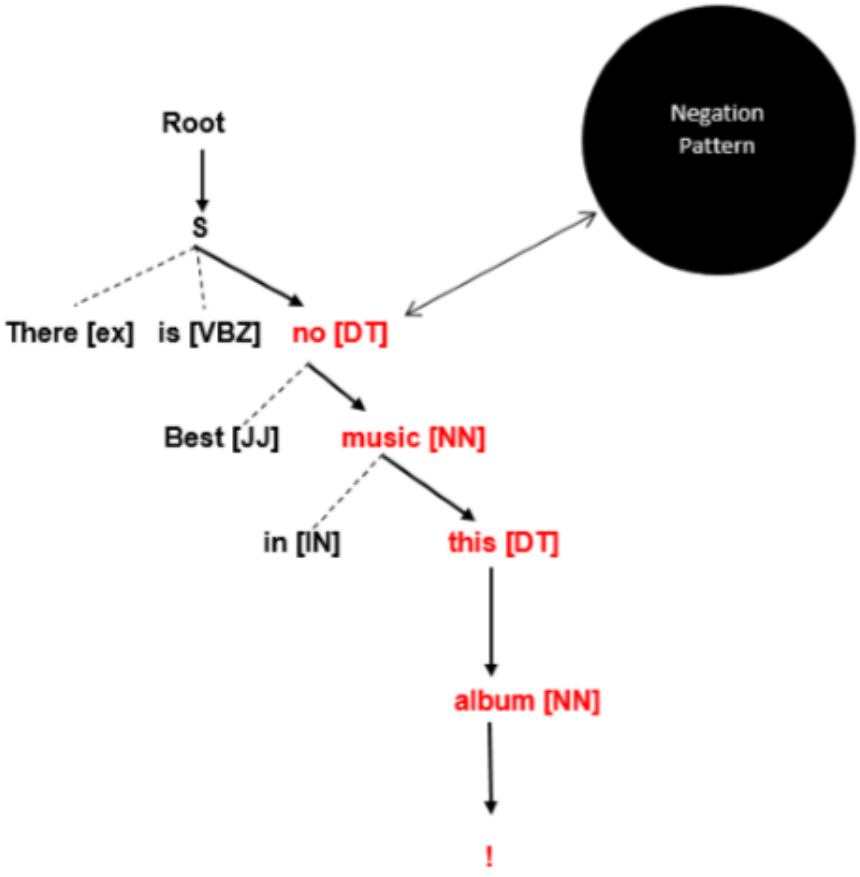


FIGURE 4.8: Negated sub-tree derived from the main parse tree

Tags	Meaning
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
IN	Preposition
JJ	Adjective
JJR	Adjective comparative
JJS	Adjective, superlative
MD	Modal
NN	Noun, singular
NNP	Proper noun, singular
NNS	Noun, plural
NNP	Proper noun, singular
PRP	Personal pronoun
SYM	Symbol
FW	Foreign words
LS	List item marker

TABLE 4.11: Meaning for POS

4.8 GUI

The computational framework introduced in Chapter 1 is presented in this Section. This framework automatically created searches for the terms that a user wants to track, such as brand names and corporate terms (Zheng, Wang, and Gao, 2018). Textual data was extracted from social media and other sources of information. Our design presents the information in an understandable way, which makes it possible to understand the results produced by the system. The color allowed us to classify web pages crawled with respect to their polarities (Kim, Youn, and Lee, 2019). Table 4.12 is a list of some meanings about colors and how they are interpreted by the GUI framework:

Colour	Meaning
Red	S_i^-
Yellow	S_i^+
Green	S_i

TABLE 4.12: The meanings of colors for the GUI

"NA" is used to indicate that the textual contents of a certain web page is not available. "Date" is used to indicate the date a website was written. "Sentiment" indicates the polarity prediction. "Source" is used to determine the source of web pages. The Figure 4.11 displays the result for the query "Wits University" and displays the result for the query "Wits students protest". The user name and password would have been given to the user by the system administrator. The back-end MySQL DB contains a table for all users that stores credentials for both their usernames and passwords. If both the user name and password match, then the user is logged into the system. Otherwise, the user is presented with an error message and is asked to re-enter their credentials.

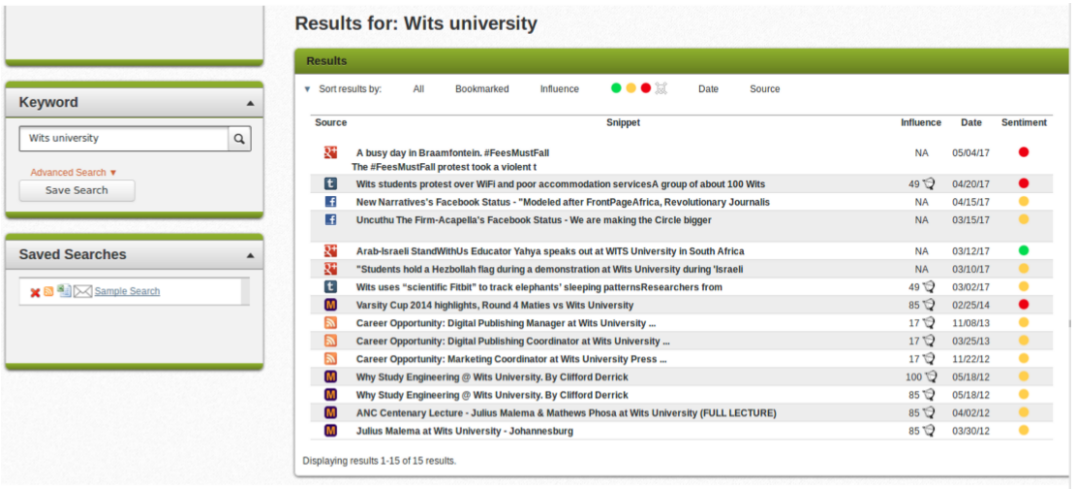


FIGURE 4.9: Results for: Wits University.

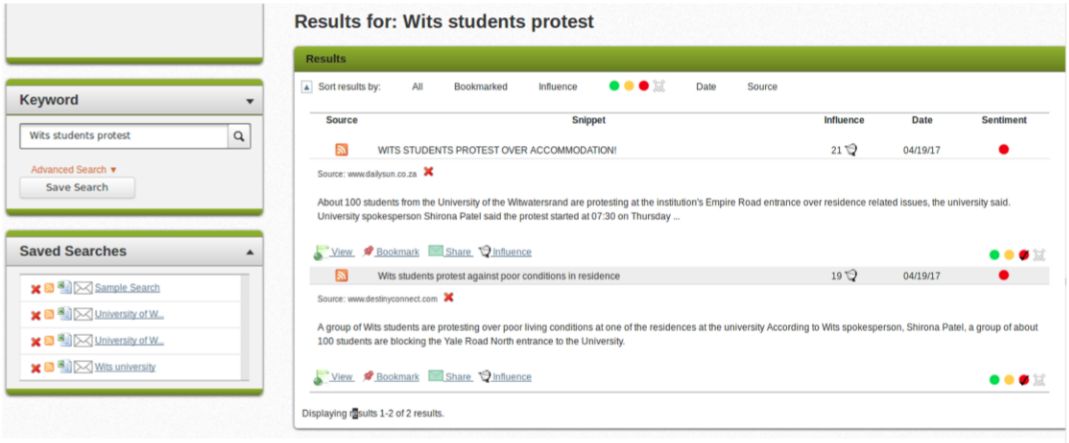


FIGURE 4.10: Results for: Wits students protest.

FIGURE 4.11: Reputational framework.

4.9 Storage of Data in a Database

A MySQL database was set up in order to store textual data. We chose MySQL for this purpose as it is easy to use with the Python programming language and supports the major SQL functions that are needed, such as creating tables, inserting and updating data (Szárnyas et al., 2018).

It was necessary to create a "crawler" table in order to store Web pages, their contents, and additional information. A "raw text" table was also necessary in order to store extracted texts related to Web pages. We chose a relationship of "One-to-many" between the "crawler" table and the "raw text" table because we could extract various textual contents from one Web page. The label and sentiment tables could then be linked via an intermediate table in order to record the polarity of classified observations/ sentences (Szárnyas et al., 2018). We implemented a relationship of "One-to-one" between the label and sentiment tables. This was based on the fact that one observation/ sentence recorded in the "label" table can only have one polarity stored in the "sentiment" table. This is shown in Table 4.13.

Table	Relationship
"Crawler" and "Raw text	1 : ∞
"Label" and "Sentiment"	1:1

TABLE 4.13: Tables Relationships

The polarity of the entire web page was displayed in the final reputation analysis application.

4.10 Back-End Development and Web Browser Specifications

The framework was developed with a range of web browsers in mind. The layouts of each individual page within the reputation analysis application were designed so that the pages would look presentable no matter which web browser is used to access the system.

In addition, the elements on the web pages respond to the resolution of the monitor being used, ensuring that each part of the application can be viewed regardless of screen size. The recommended browsers to use when accessing the reputation analysis system consist of the following, as displayed in Table 4.14.

Mozilla Firefox	Google Chrome	Microsoft Edge	Opera

TABLE 4.14: Latest Versions of Recommended Browsers for the reputation analysis System

The server-side application shown in Figure 4.12 is connected to the DB via the web crawler, which pulls, saves, or changes data. We used the Pyramid web framework and incorporated external Python libraries, HTML, CSS, and JavaScript.

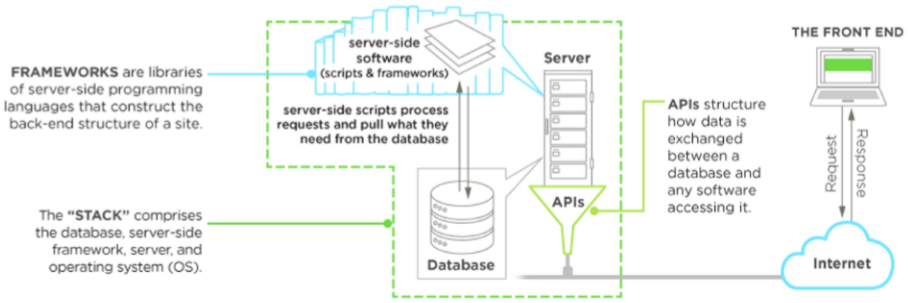


FIGURE 4.12: Back-End Development.

4.11 Problem

Our vision is to detect any sequence $\{p_{j1}, ..., p_{ji}, ..., p_{jn}\}$ of n pages so that the general sentimentality $S(C_n)$ of the crawled web pages C_n is significant, this implies that we

want to increment (Vural et al., 2013):

$$S(C_n) = \sum_{i=1}^{n} S_{j_i} \tag{4.3}$$

The objective is to optimize the general sentimentality of crawled pages while the platform is performing web crawling in real-time (see Figure 4.13).

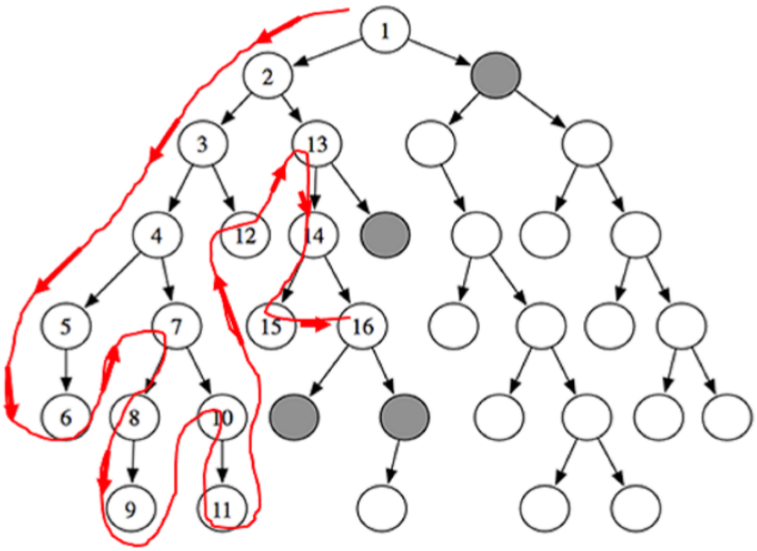


FIGURE 4.13: Schematic representation of crawling. Each node represents a single Web page, the red line represents the crawling direction.

4.12 Discussion

This Chapter 4 addresses the concept of content extraction. In Section 4.1, we addressed data preparation process that clean-up irrelevant features such as advertising and menu bars - a tool named BoilerPipe has been used to accomplish this task. Similarly, Section 4.1, also discussed the incorporation of the Standard -core-NLP library into the CRA framework - this component performs token segmentation of extracted text using POS. In Section 4.1.2, we present the algorithms used for Sentiment Detection such as, ANN and Sentistrength. We then presented the lexical corpus that has 50 entries in total. The experimental set-up is also explained in Section 4.2.1 as well as the dataset utilized for preliminary experiments.

The crawling mechanism is presented in Section 4.2.2 where each web page is ranked by the crawler and the sentiment score is assigned to find the overall opinion of a specific web site. Section 4.3.1 presents Emoticons used in the training process while Section 4.4 discussed the training approach in which training and testing datasets were built - the % symbol is used to read each observation line by line in the dataset.

A discussion on how to find the optimal separation in the dataset is also carry out. Additionally, the training results are discussed in Section 4.5 - ANN produced approximately 11% FP, 2.7 % FN and 0.1 % of Empirical error. The results revealed that

most data can be classified correctly with ANN. The BOW results are presented in Section 4.6.1 with the keyword "students" having the highest occurrence - this indicates that the dataset is made of educational content.

The discussion pertaining to negation detection process is given in Section 4.7.1 - this process does improve Sentiment Detection as discussed by Taboada (2016). Lastly, we presented the computational framework in Section 4.8 and a discussion with regard to data storage is undertaken - we used MySQL as it is easy to used and it is compatible with Python programming language in terms of Data Manipulation Language (DML) and Data Definition Language (DDL).

4.13 Conclusion

Throughout this Chapter, the results pertaining to the research question we posed in Chapter 1 were detailed. We first addressed the concept of content extraction. We proceeded to then consider the Sentiment Detection methods utilized. Next, to find out the orientation of text in Sentiment Detection, we identified the negation and its scope because negation can completely change the polarity of words and statements. Experiments, results, and discussion are given in the next Chapter 5.

Chapter 5

Exploratory results

This Chapter 5 evaluates different methods addressed in Chapter 3 and Chapter 4. The Chapter describes the datasets used and presents results in terms of measures that have been utilised to classify textual data. This Chapter compares different performance metrics obtained from the datasets used. The experiment starts with a discussion of results obtained from the WITS marketing team datasets. This is followed by a discussion about the performance of algorithms used on Clueweb09 dataset as well as the performance of the Sentistrength algorithm.

5.1 An analysis of ANN, BOW, and NER

ANN, BOW, and Sentistrength are coded and tested in this Chapter. The results provided by these algorithms are presented. We plotted certain results and illustrated them throughout this Chapter. Words and sentences as well as their prediction (positive, negative, and neutral) are quantified and visualised for a deep Corporate Reputation Analysis (Donneau-Golencer et al., 2019). The computing environment is presented in the following Section 5.1.1.

5.1.1 Anaconda-Spyder

Anaconda has been used as a tool of implementation. We utilised thousands of sentences to assess and train the framework (Guggilla et al., 2018). A whole corpus for training and evaluation has been used to handle sentiments of textual data (Nikam, 2015). Therefore, we evaluated each algorithm one by one on different datasets. Each algorithm is tested with both the corpus for testing and training as in Mehdad, Pappu, and Stent (2018). The NER is assessed based on the strength of an entity presence on the internet (Mehdad, Pappu, and Stent, 2018).

5.1.2 Evaluation criteria

In order to compare classification algorithms, classification measures are needed (Yılmazel and Arslan, 2019). This, for example, measures the prediction performance of textual data (Mehdad, Pappu, and Stent, 2018). These measures help in terms of determining whether the content of a specific textual subset is positive, negative, or neutral (Mehdad, Pappu, and Stent, 2018).

5.1.3 Neural Network initialisation

Figure 5.1 illustrates the initialisation process of the computing environment. In Figure 5.1, the ANN algorithm is analysing the WITS dataset on an exponential level-Eigenvectors maintains the same direction. The confusion matrix data type (float)

is presented. On the left side of Figure 5.1, a function evaluates hypothesis on the input dataset and returns the corresponding output of the hypothesis. The function learns a hypothesis of ANN types on training datasets and returned the learned parameters of the hypothesis (Huang, Capretz, and Ho, 2019). The cross validation type, size, and values are illustrated on the right side of Figure 5.1.

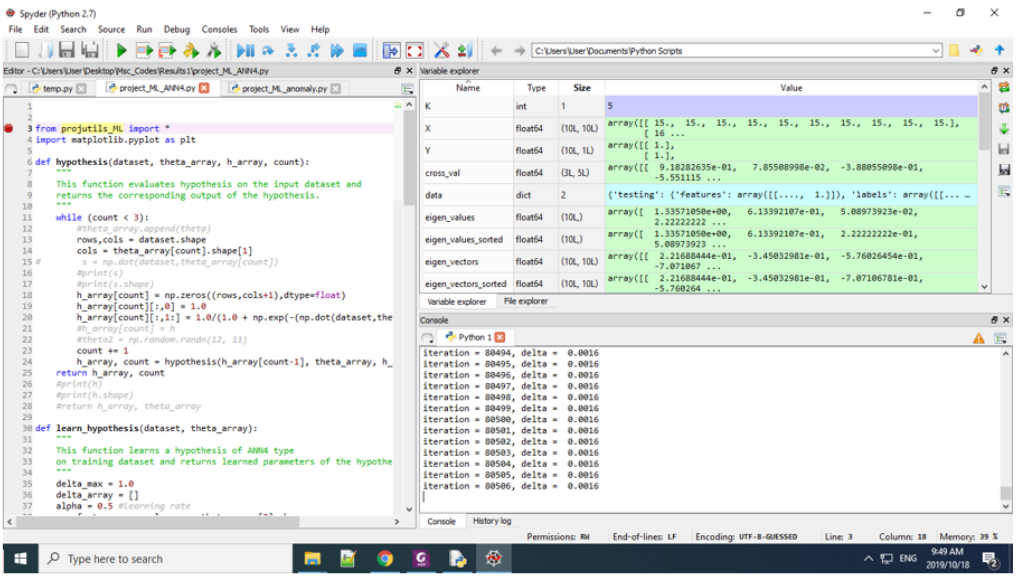


FIGURE 5.1: Anaconda-python IDE running ANN. This process used 39% of the RAM (random access memory).

5.1.4 Prolongation of number of iteration

Figure 5.2 physically illustrates that the ANN algorithm is constantly learning due to the fact that the cost function is decreasing without direct contact on the x axis (Michalski, Carbonell, and Mitchell, 2013). In depth, the graph strictly dictates that the number of iterations will prolong as the algorithm does constantly reanalyse the textual data set (Das, Govardhan, and Lakshmi, 2019).

The confusion matrix in Table 5.1 has accurately accumulated 89 features as TP and an occurrence of 25 features as TN. The over calculation of the ANN algorithm accounted 114 features (89+25) that have been correctly classified.

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	89 = 72.3577235%	2 = 1.62601626 %
	Negative (Predicted)	7 = 5.6910569 %	25 = 20.3252032 %

TABLE 5.1: Evaluation metrics for ANN

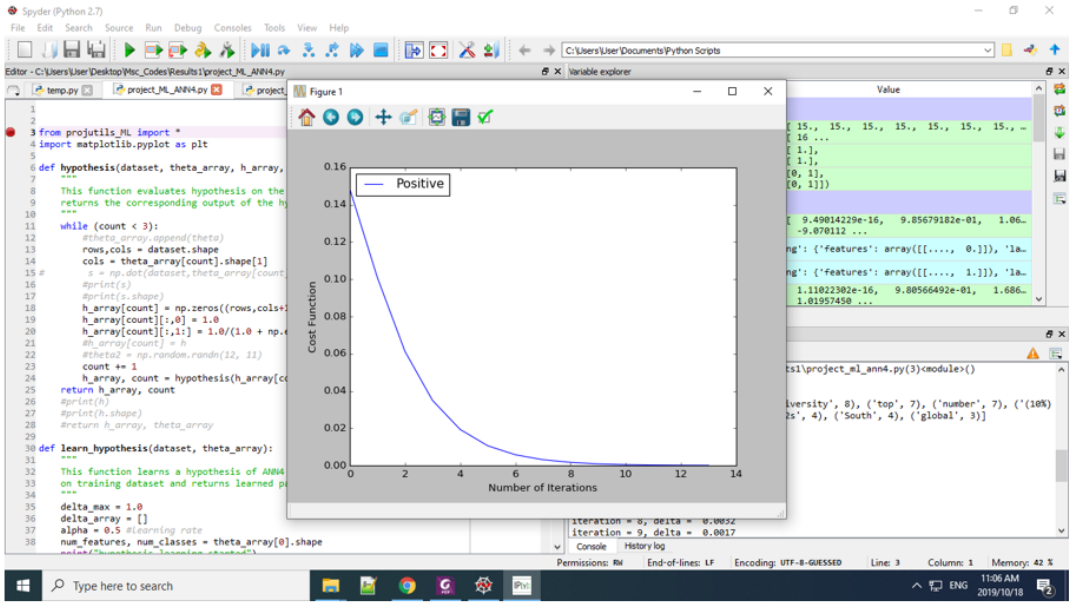


FIGURE 5.2: Visualisation of the learning process using ANN.

The probability of the confusion matrix gives seven and two features incorrectly classified as FN and FP. With additional data to the previous data set (see Table 5.2), the accumulated data set has proven to be more functional as the results have become more accurate and have increased the occurrence of features detected (see Figure 5.3) - this was also proven in Das, Govardhan, and Lakshmi (2019).

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	100	50
	Negative (Predicted)	30	14

TABLE 5.2: Evaluation metrics for ANN

5.2 Data paucity

In NLP, data sparseness/ paucity is utilised to describe the phenomenon of not learning accurately from data in a corpus (Tsai et al., 2013). This Section uses the confusion matrix to assess the performance of ANN with regard to data sparseness (Krahel and Titera, 2015). We visualized evaluation metrics in Figure 5.3. It has been proven that ANN has successfully classified 100 sentences as positive. Hence, this algorithm has proven its functionality and has solved the data paucity problem as discussed in Krahel and Titera (2015). In comparison to TN, the difference is much greater than TP - this was also proven by Ingre and Yadav (2015). Thus, the classification of FN has a lesser value as compared to FP (Ingre and Yadav, 2015). As demonstrated by Ingre and Yadav (2015), the natural reading of the learning algorithm attains an entity of higher volume of positive occurrence/ prediction as to of negative occurrence. We thus argue that the content of this WITS sample dataset is positive and the algorithm minimises FN/ FP.

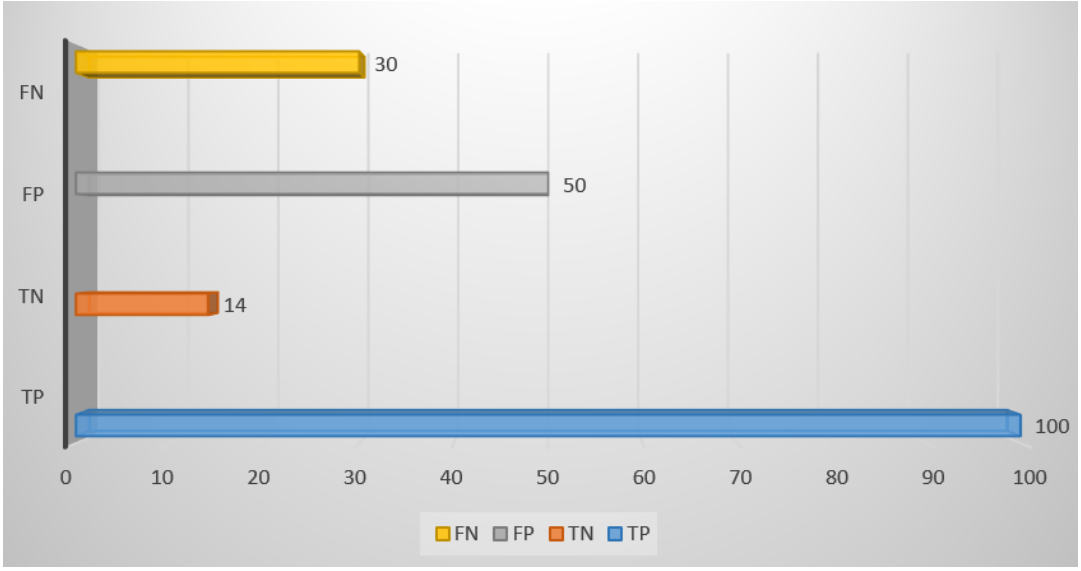


FIGURE 5.3: Confusion matrix visualisation (with additional dataset).

Words	Occurrence
Wits	200
University	100
Research	45
Opportunity	10
Science	30
Marketing	20
Leading	10
students	10

TABLE 5.3: BOW Results for WITS dataset

Table 5.3 is a representation of the BOW results retrieved from the WITS dataset. The Table 5.3 keeps a count of occurrences of the most frequent keywords detected in the dataset (Redhu et al., 2018). In Figure 5.5, the recognition unit provides the result of the BOW. This recognition unit outputs results with numerical values. This feature quantification technique enabled us to study the overall polarity of this WITS dataset sample which is more neutral than positive (see Figure 5.4).

5.3 Central Process Unit (CPU) time consumption

Consumption time is used to study the long CPU time of ANN (Seth et al., 2019). This concept gives an indication of the algorithmic efficiency (see Figure 5.6, the greater the data point, the more time consuming it becomes for ANN to provide better Accuracy).

ANN is guaranteed to terminate as well as produce correct results after a finite time (Seth et al., 2019). Unfortunately, results from the experiment demonstrated in Figure 5.6 illustrates the limiting factor of ANN which suffered from significant

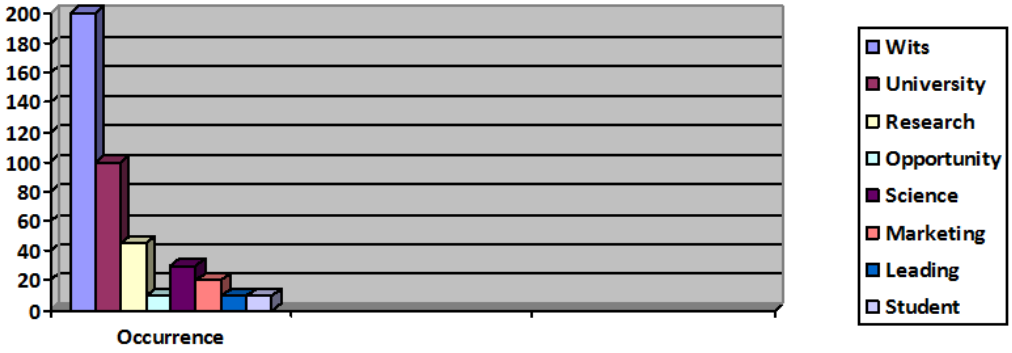


FIGURE 5.4: Data presentation. The overall polarity of this dataset is neutral as no negative keywords have been identify by the NER.

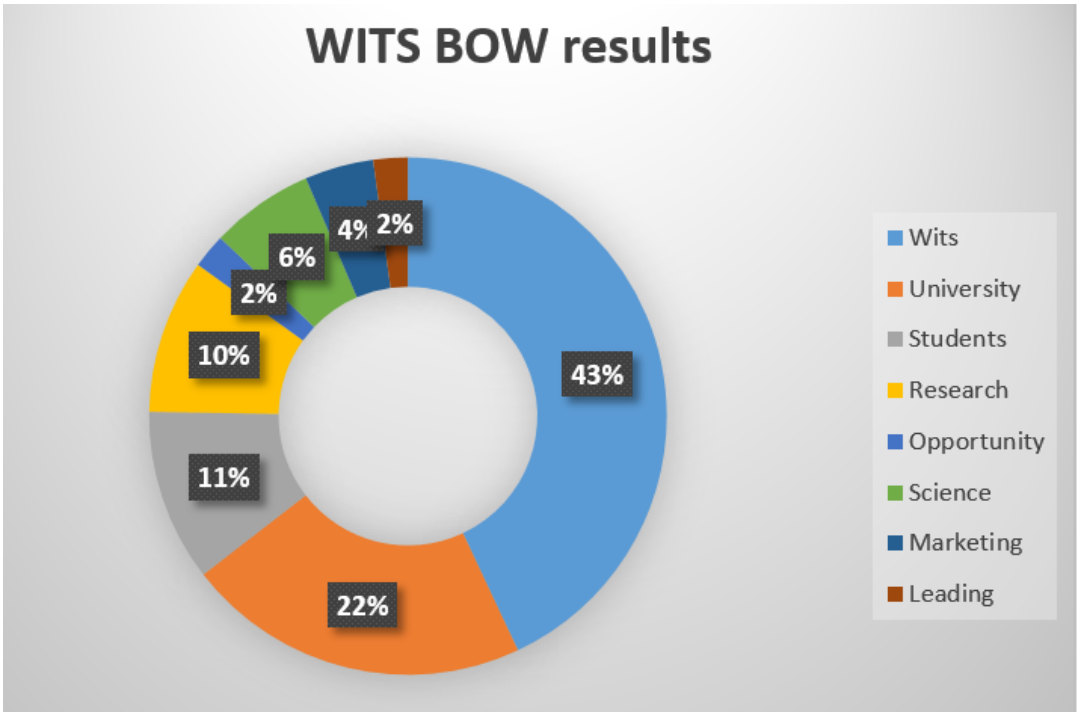


FIGURE 5.5: Visualisation of the WITS BOW results.

slowdowns with larger datasets (Nikam, 2015). An increasing amount of data to process as well as additional features exponentially increased the processing time required for the learning algorithm to terminate (Nkongolo, 2018). However, since the algorithm tries to find an exact fit for the textual data, the Accuracy of hypotheses generated is likely to be high (Michalski, Carbonell, and Mitchell, 2013). Thus, an individual considering this algorithm for use may have to look at the trade-off between processing time and Accuracy and make a determination based on their needs and wants (Guggilla et al., 2018). In Chapter 1, it has been proven that SVM and DT produced less efficient and accurate results (Jadav and Vaghela, 2016; Rasool et al., 2019).

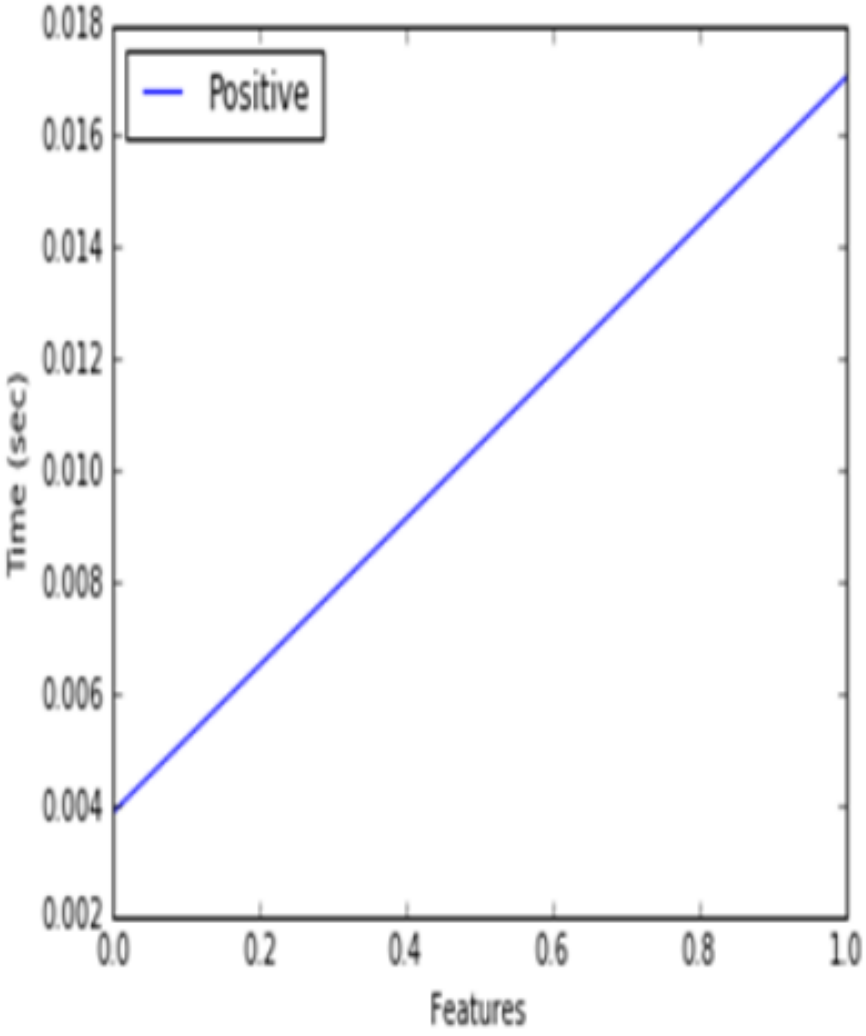


FIGURE 5.6: Illustration of time consumption.

Jadav and Vaghela (2016) and Rasool et al. (2019) argued that using NB when selection features are dependable yield low Accuracy. Thus, there is a need of using ANN in Data Mining - especially in text processing to improve Accuracy, optimise the running time, and minimise the misclassification percentage (Nikam, 2015).

A better visualisation of an improvement in Accuracy during the testing process is given in Figure 5.7. The graph in Figure 5.8 represents different phases of the learning and training process.

In the training phase, the rate of FN and FP - which represents incorrect classification - starts of low due to the fact that the algorithm has already learned with previous features (Yılmazel and Arslan, 2019). In the training phase, the content that was used exhibited more negative connotations than positive. However, when ANN was tested, there was an increase in TP (positivity) and a decrease in TN (Ingre and Yadav, 2015). This was due to the fact that the textual dataset's content used in the training phase differs from the textual dataset's content used in the testing phase. The average of the Accuracy remains fairly consistent throughout all phases

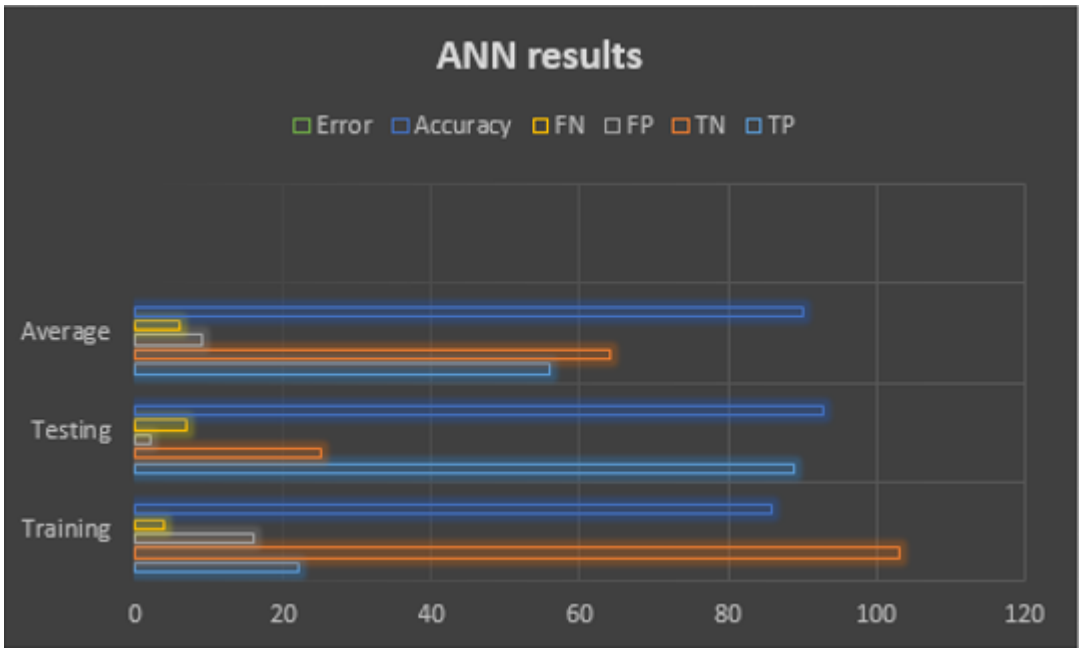


FIGURE 5.7: ANN performance visualisation.

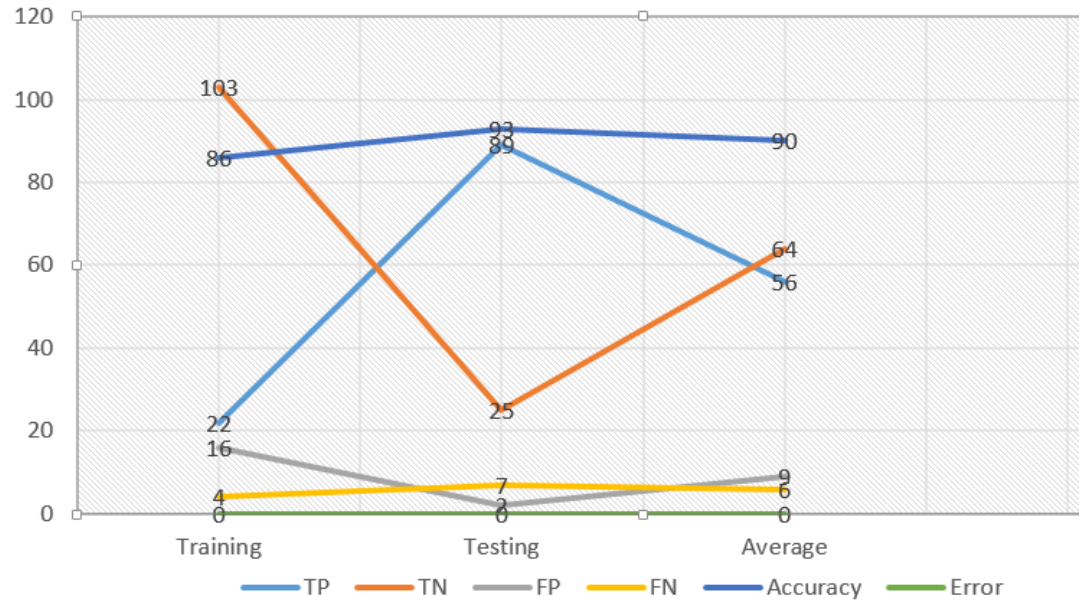


FIGURE 5.8: ANN Accuracy visualisation.

of our learning and training process which exhibits a fairly Accurate usage of the algorithm.

The Empirical error remains at zero for each phase of our learning and training process and, as a result, we can conclude that our algorithm is not prone to errors as discussed by Arif-Uz-Zaman et al. (2017). To enhance text classification numerically, we extracted the most optimal features from the input dataset. These features are represented by the BOW results. Depending on the characteristics of the textual

dataset, it can be difficult to interpret classification results (Lukyanenko et al., 2019). Thus, heuristics techniques namely, Classes- Similarity Measure (CSM), Cost-Based Measure (CBM), and Classes-Number Measure (CNM) can be used as in Tsytsarau and Palpanas (2012) with regard to Accuracy improvement.

5.3.1 Evaluating ANN confusion matrix

Looking at Table 5.4, we can deduce that ANN has better performance on textual datasets (Hariri, Emami, and Malek, 2015). ANN accurately classified 185 sentences (134+51) as TP and TN. On this WITS sample dataset, the algorithm has considerable rates of misclassification (94 + 55 = 149).

		Predicted	
		Positive (Actual)	Negative (Actual)
Actual	Positive (Predicted)	51 = 15.2694%	94 = 28.1437%
	Negative (Predicted)	55 = 16.4670%	134 = 40.1197%

TABLE 5.4: The difference between accurate classification and misclassification.

However, when we consider features that have been correctly classified and compare them to features that have been incorrectly classified, we have a ratio value of 36 (185-149). This value can thus represent the difference between correct and incorrect classification visualized in Figure 5.9 where WITS datasets one, two, and three are samples of different size (imbalanced data).

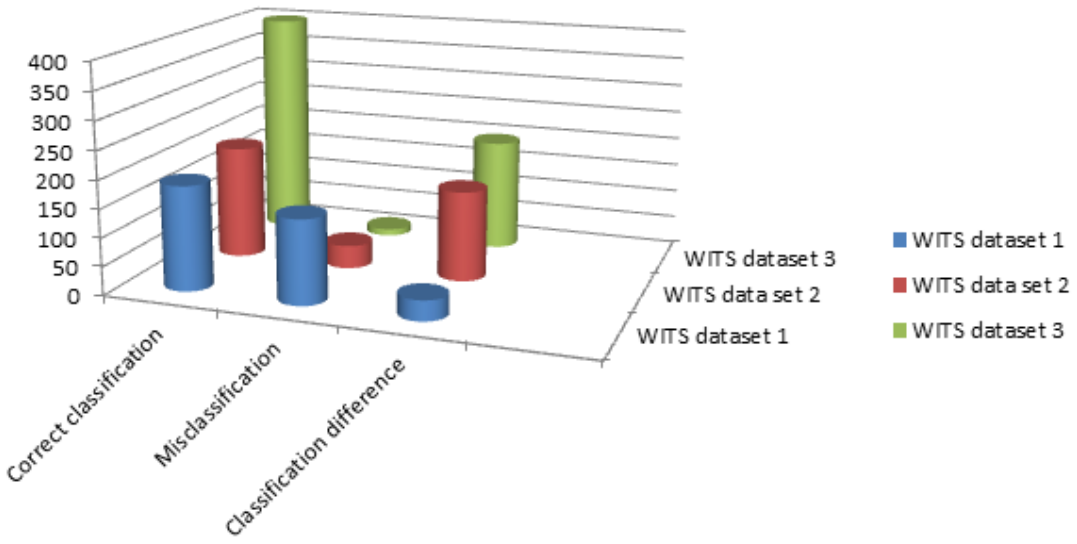


FIGURE 5.9: Imbalanced data.

The graph shows inconsistency between the classification ratios of the various datasets due to an imbalance of data points (Arif-Uz-Zaman et al., 2017).

5.3.2 Evaluating Precision

Figure 5.10 shows the maximal Precision value of 100% yielded by the WITS dataset 12 in the training and testing processes. This Figure 5.10 also indicates a decrease in terms of classification ratios - this is a good indication of Precision. The classification ratio is inversely proportional to the Precision value (Depren, Aşkın, and Öz, 2017). It shows that ANN accurately classifies features with *great* Precision.

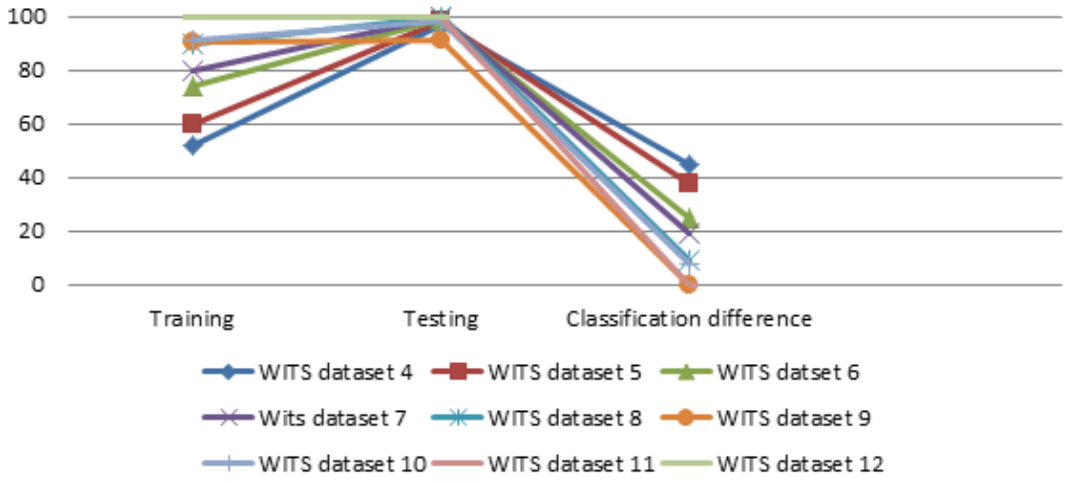


FIGURE 5.10: Precision performance

5.3.3 Evaluating Recall

In Figure 5.11, the Recall value reached 92% while testing the algorithm. However, this value decreased by eight (92-84) while training ANN. On average, the Recall value is 88%. This represents the completeness of ANN as discussed by Haixiang et al. (2017). In this Figure 5.11, higher Recall means less FP and FN (Seth et al., 2019). As indicated, improving the Recall value increases the TN and TP as the sample space is augmenting (Mehdad, Pappu, and Stent, 2018).

5.3.4 F-measure Vs Precision

In general as indicated in Figure 5.12, both the F-measure and Precision values increased during the testing phase.

This increase can be seen as an indicator that portrays exceptional classification accuracy as demonstrated by Szárnyas et al. (2018).

5.4 BOW and NER evaluation

Figure 5.13 indicates the relationship between the entity, category, and the BOW algorithm performance (Baradwaj and Pal, 2012). Entities were extracted from web pages then classified under appropriate categories followed by the entity's word counts (Guggilla et al., 2018). For instance, IBM is categorised as an organisation with a word count of 14.

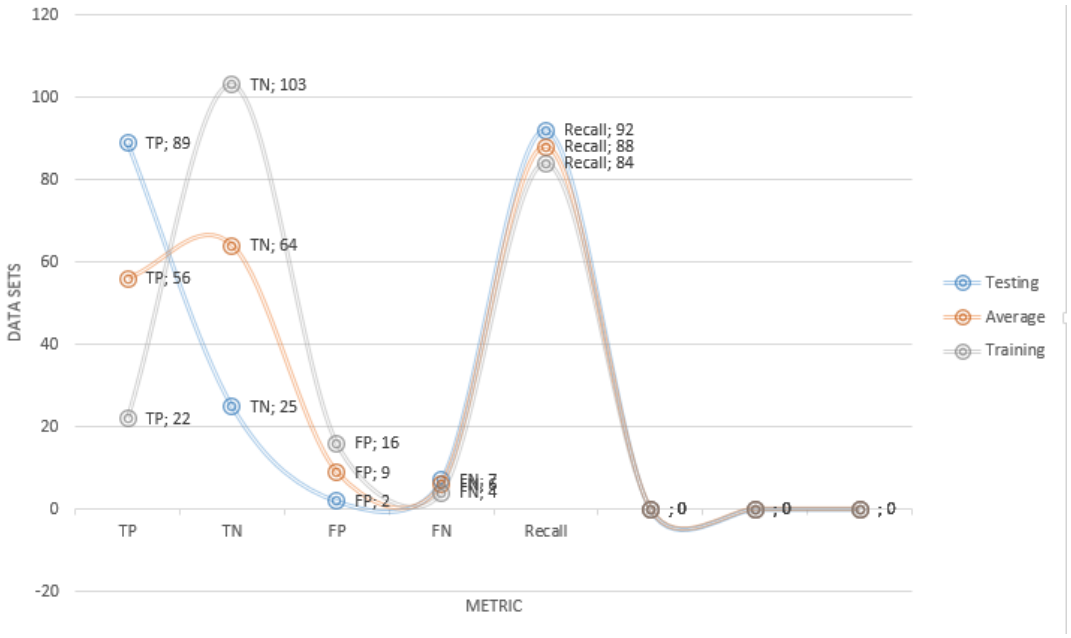


FIGURE 5.11: Recall values improvement

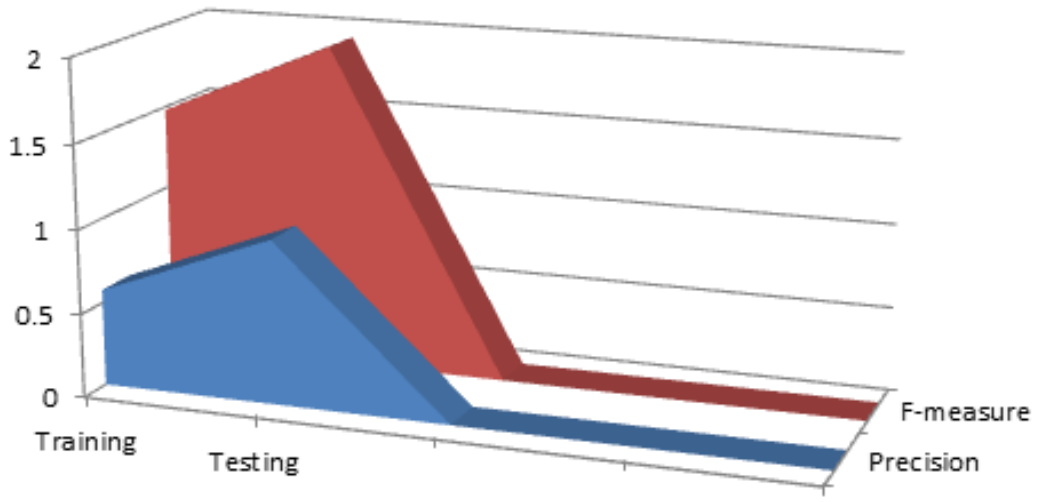


FIGURE 5.12: F-measure improvement

5.5 Sentistrength algorithm evaluation

In Figure 5.14, Sentistrength algorithm shows the Highest polarity and Compound of each sentence in the WITS dataset. Nine sentences were classified as negative - as presented in red with 0.479 score as the lowest polarity and -0.1531 as the lowest Compound score.

This is a result of the following words: "poor", "FeesMustFall", "failed", "bombs", "lost", "negatively", etc. Positivity is represented by yellow with the Highest polarity being one and its highest Compound being 0.6369.

Entity	Categorised as	BOW
University of the Witwatersrand	University name	300
South Africa	Country	278
Braamfontein	Location	200
1:PM	Time	152
Prof Vilakazi	person	134
10000R	Money	122
11-12-2019	Date	111
University of Johannesburg	University name	105
Johannesburg	Location	100
1 January 2005	Date	95
Soweto	Location	90
Adam Habib	Person	50
1 Jan Smuts	Location	45
Judy Dlamini	Person	30
1922	Date	26
Jorissen Street	Location	20
IBM	Organisation	14

FIGURE 5.13: Entity categorised by the NER and BOW

This was due because of the following words: "scholarship", "highest", "Prestigious", "smart" , "clever", "peace", etc. The green representing neutral text with a mostly constant polarity of 0.1 and Compound value of 0.

Sentence	Highest polarity	Compound
over poor living conditions	neg: 0.479	-0.557
FeesMustFall	neg: 0.474	-0.4019
failed the examination	neg: 0.398	-0.5106
are not ready to write the test	neg: 0.19	-0.2263
bombs found at Wits University	neg: 0.192	-0.2263
lost is first place	neg: 0.215	- 0.2323
negatively affected	neg : 0.151	-0.1531
accused the university	neg: 0.239	-0.296
do not like the exam time !	neg: 0.325	-0.3404
violence appear to be rooted	neg: 0.24	-0.6249
The Absa scholarship	pos: 0.545	0.5563
The highest number of endowed professors	pos: 0.085	0.0772
Prestigious World Universities	pos: 0.0	0.0
Smart professors	pos: 0.915	0.0772
The clever boys	pos: 0.0	0.0
I will make it at wits	pos: 1.0	0.0
Passed the exams and enjoyed	pos: 1.0	0.0
Restore peace at Wits	pos: 0.348	0.5873
Nice place to be	pos: 0.333	0.4588
Johannesburg, is the best	pos: 0.344	0.6369
Wits is a nice place to be	pos: 0.359	0.4215
I love programming	pos: 0.488	0.6369
Let go at wits	neu: 1.0	0.0
Witsies went in Germany	neu: 1.0	0.0
Students are inside the venue	neu: 0.1	0.0
Your supervisor called you	neu: 0.1	0.00
We are going at Kambule building	neu: 0.1	0.0
Try more maybe	meu: 0.1	0.0
Students did it	neu: 0.1	0.0
I am neutral	neu: 0.1	0.0
After graduating	neu: 0.1	0.0
My subjects are here	neu: 0.1	0.0
The university of the Witwatersand	neu: 0.1	0.0
Prof Adam Habib is the vice chancellor	neu: 0.1	0.0
The more you try, the luckier you get	neu: 0.4404	0.707
I can play chess	neu: 0.455	0.34

FIGURE 5.14: Sentistrength performance evaluation

5.6 Clueweb09 dataset

We present and analyse the Clueweb09 dataset in this Section. Clueweb09 has 240 unique topics. Thus, the dataset is suitable for experiment (Gasc et al., 2019). To get enough data, we only consider plain text which allowed us to build the corpus and

the dataset is 15 GB in size.

As illustrated in Figure 5.15, "Unique Topics" dataset has accumulated 600 features accurately classified as TP in comparison to "Topic-Doc-Number" dataset which accumulated 800 features correctly classified as TP.

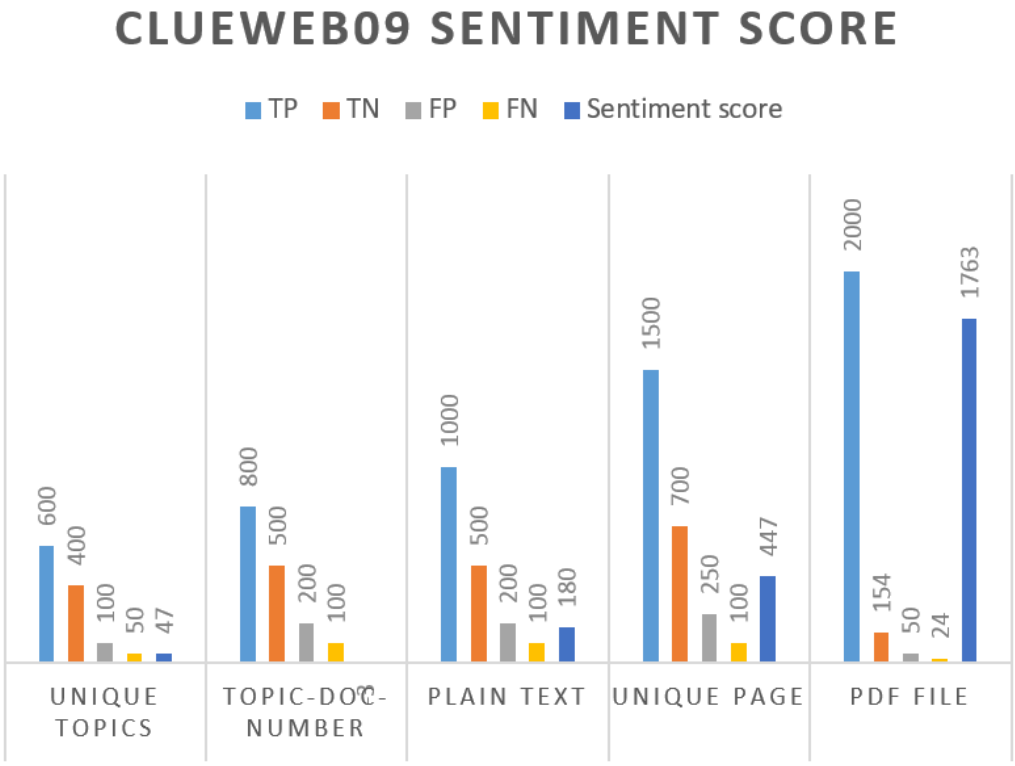


FIGURE 5.15: Clueweb09 sentiment score performance

Additionally, "Pdf file" yielded the best TP accumulation with 2000. This applied for TN as well where "Unique page" produced an accumulation of 700 for TN. Overall, the FN and FP values have been decreased. This is an indication demonstrating that ANN did not suffer from data paucity (Zishumba, 2019). Lastly, the sentiment score produced by Sentistrength algorithm on this dataset has increased for the "Pdf file" subset with 1763. In conclusion, "Pdf file" is the most suitable dataset to be used to train any CRA framework because the SA algorithm yielded the highest sentiment score on this dataset.

5.6.1 Cross-Validation

Cross validation can be defined as a technique for assessing how the statistical analysis generalises to an independent data set (Massucci and Docampo, 2019). It is a technique for evaluating ML models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data (Stuart and Majewski, 2015). Using cross-validation, there are high chances that we can detect over-fitting with ease.

Cross-Validation is also a data mining approach which requires reserving a specific sample of a dataset on which we do not train the model (Maerz and Puschmann, 2020)-later, we tested the model on this sample before finalizing it. We initially split up the entire data we have got into two sets, one is used to *train* the model upon and the other is kept as a *holdout* set which is used to check how the model behaves with completely unseen data (Maerz and Puschmann, 2020; Massucci and Docampo, 2019). The Figure 5.16 summarises the entire idea of performing the split.

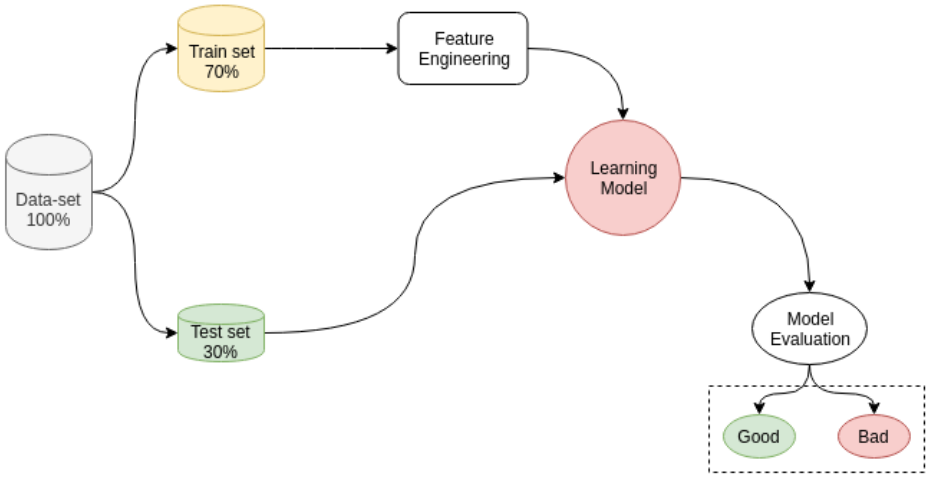


FIGURE 5.16: A very high-level view of the model building and Croos-validation pipeline

In this Section we present the steps involved in cross validation (Maerz and Puschmann, 2020):

1. Sample data set reservation.
2. Train the model-using the remaining part of the dataset.
3. **Use the reserve sample of the test validation set**. This enabled us in gauging the effectiveness of the model's performance (Maerz and Puschmann, 2020).

The first step isolated the *test* data-set and use it only for final evaluation. Cross-validation is then performed on the training set (see Figure 5.17).

Initially, the entire training data set is broken up in k equal parts (Maerz and Puschmann, 2020). The first part is kept as the hold out (testing) set and the remaining k-1 parts are used to train the model (Maerz and Puschmann, 2020). Then the trained model is then tested on the holdout set. The above process is repeated k times, in each case we keep on changing the holdout set. Thus, every data point get an equal opportunity to be included in the test set. Usually, k is equal to three or five. It can be extended even to higher values like 10 or 15 but this research revealed that by doing so, it becomes extremely computationally expensive and time-consuming (Li et al., 2019).

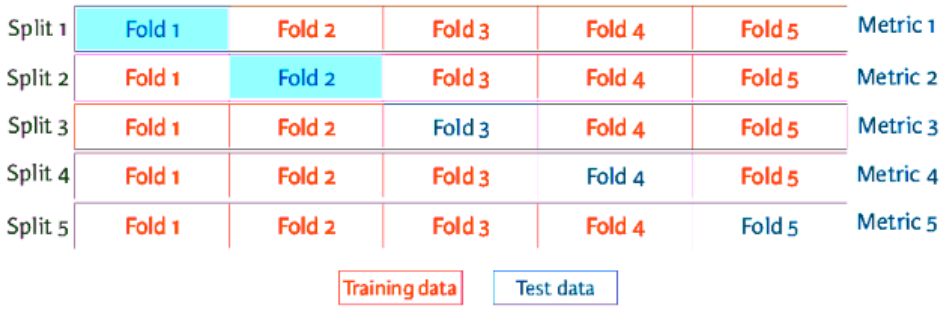


FIGURE 5.17: K-Fold Cross Validation

This research reveals that the model delivers a positive result on validation data as demonstrated in the following experimentation. Separate data sets for training and evaluation were collected (Maerz and Puschmann, 2020). The evaluation data set represents the samples to which the prediction model built from the training set would have been applied (Maerz and Puschmann, 2020).

The ML models are *blinded* to the evaluation data and any circumstances that affected the collection of that data during the intervening three years of this dissertation-including changes in search strategies or process (Maerz and Puschmann, 2020). Therefore, the test data used for *evaluation* is the same that a deployed computational framework would have been applied to during that time period. Thus, we can be confident that the performance obtained reflects the performance that would have been seen in a deployed computational framework over this time period.

Descriptive statistics on the prior and new data sets are shown in Table 5.5. As can be seen over 11 samples/ topics, 16.900 sentences had been collected in the prior data[1], with over 16.900 sentences of new data[2] collected across these 11 samples/ topic.

	Topic	Prior Data-1	Prior Data-2	Prior Data-3	Total	New-Data-1	New-Data-2	**Total**
1	education	200	150	300	650	400	250	650
2	research	100	350	100	550	200	350	550
3	Alumni	400	600	100	1100	500	600	1100
4	science	700	400	100	1200	600	600	1200
5	sport	200	200	100	500	250	250	500
6	religion	100	40	60	200	50	150	200
7	society	700	300	500	1500	1000	500	1500
8	ranking	800	200	300	1300	600	700	1300
9	protest	300	100	400	800	300	500	800
10	students	3000	4000	1000	8000	6000	2000	8000
11	Facebook	700	300	100	1100	600	500	1100
	Total	7200	6640	3060	16900	10500	6400	16900

TABLE 5.5: Data preparation for Cross Validation

[1]Prior data means training data
[2]New data means Testing data

5.6.2 Receiver operating characteristic curve (ROC) - Uncertainty values for Accuracy

We conducted two types of evaluation on the prior and new data sets. Since the sub-goals of this study was also to determine how well predictive models created with the prior data set perform on prospectively collected data, for making comparison, both sets of measurements utilize the area under the receiver operating curve (ROC) (Mall, 2018).

Given a set of textual data set, where some are positive and some are negative or neutral for a classification task, ROC is a good measure of the system performance (Brabazon, 2016), where 1.0 is a perfect score (Das, Govardhan, and Lakshmi, 2019).

The ROC curve is a performance measurement for classification problem at various thresholds settings - it is a *probability* curve that represents degree or measure of *separability*. It tells how much the model is capable of distinguishing between classes. In this research the Higher the ROC value, better the model is at predicting (Massucci and Docampo, 2019). By analogy, the Higher the ROC value is, the better the model is at distinguishing between positivity and negativity of sentences.

The ROC curve is plotted with TP rate (TPR) against the FP rate (FPR) where TPR is on y-axis and FPR is on the x-axis.

ROC is independent of class prevalence and is thus a good measure to use when comparing performance, which typically have different class prevalences.

We performed five repetitions of two-way (5*2) cross-validation with $stratification$ on the prior data set, for each topic - to keep the ratio of positive to negative samples consistent between training and test splits (Chiranjeevi, Santosh, and Vishnuvardhan, 2019; Oshikawa, Qian, and Wang, 2018). Each two-way cross-validation was randomly and independently split.

The resulting ROC measurements where then averaged together to create a mean ROC measurement for each topic/sample - This process results in a *baseline* set of performance for each topic/ sample, and allows comparison with our prior results on this data (Shukri et al., 2015; Lewandowski, 2020). We trained our classification system on the prior data set to conduct the prospective performance evaluation and evaluated the predictions of this model when applied to the new data set - the prior data set was used as the training collection and the new data set served as the test collection.

ROC per topic was computed for the resulting predictions, we then compared the results obtained from applying 5*2 cross-validation on the prior data set with the results obtained by training on the prior data and testing on the new data.

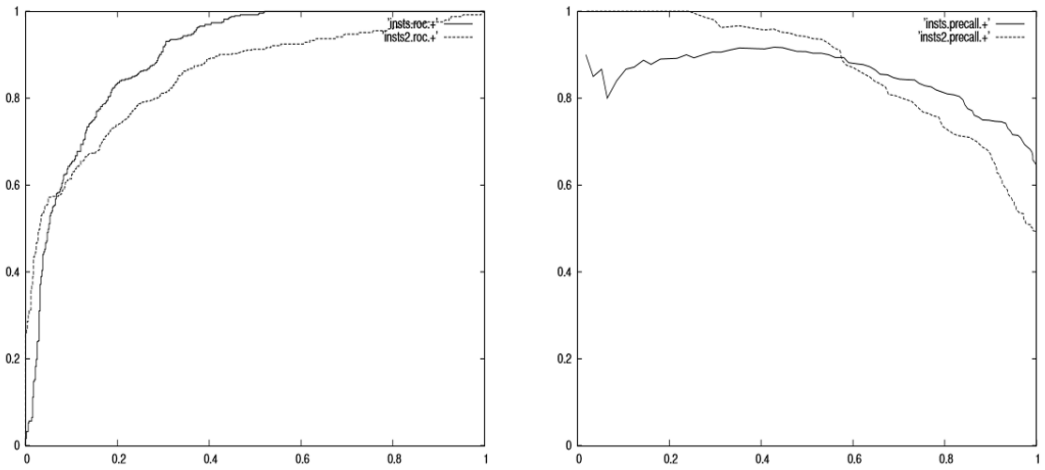
We then conduct a statistical comparison across topics. This experiment reveals that if a ML process is being used to make predictions for uncertain value of Accuracy, then it is possible that performance will suffer over time as the classification task diverges from its original definition or purpose (Michalski, Carbonell, and Mitchell, 2013). Table 5.6 depicts a Cross-validation test on the 11 sample collected previously.

	Topic	Prior Data-1	Prior Data-2	Prior Data-3	ROC average	New-Data-1	New-Data-2	**ROC average**
1	education	0.976	0.987	0.876	0.888	0.976	0.95	1.0
2	research	0.987	0.887	0.989	0.999	0.98	0980	0.888
3	Alumni	0.875	0.984	0.875	1.0	0.800	0.987	0.976
4	science	0.876	0.1	0.86	0.874	0.600	0.600	1.0
5	sport	0. 76	0.876	0.976	0.900	0.9250	0.50	0.8500
6	religion	0.876	0.986	0.856	0.9200	0.850	0.1	0.9200
7	society	0.876	0.986	0.876	0.879	0.898	0.9500	0.888
8	ranking	0.996	0.8200	0.956	0.876	0.9600	0.8700	0.8765
9	protest	0.926	0.876	0.877	0.9800	0.9300	0.8500	0.8800
10	students	0.877	0.600	1.0	0.9800	6000	0.820	0.899
11	Facebook	0.899	0.876	0.876	1.0	0.960	0.850	0.880
12	education	0.888	0.999	0.999	0.960	0.840	0.870	0.860
13	research	0.885	0.850	0.765	0.765	1.0	1.0	1.0
14	Alumni	0.987	0.765	0.989	0.896	0.884	0.8600	0.8976
15	science	0.999	0.888	1.0	0.900	0.960	0.960	0.987
16	sport	0.888	0.985	0.809	0.8500	0.9250	0.9250	0.8500
17	religion	1.0	0.940	0.960	0.8200	0.850	0.8150	0.8200
18	society	0.800	0.900	0.8700	0.8500	1.0	0.876	0.890
19	ranking	0.980	0.890	0.8300	0.800	0.8600	0.8700	0.999
20	protest	0.800	1.0	0.8400	0.8800	0.9300	0.8500	0.8800
21	students	1.0	0.765	0.999	0.800	0.90	0.800	0.870
22	Facebook	0.890	0.780	1.0	0.899	0.860	0.850	1.0

TABLE 5.6: Cross Validation results

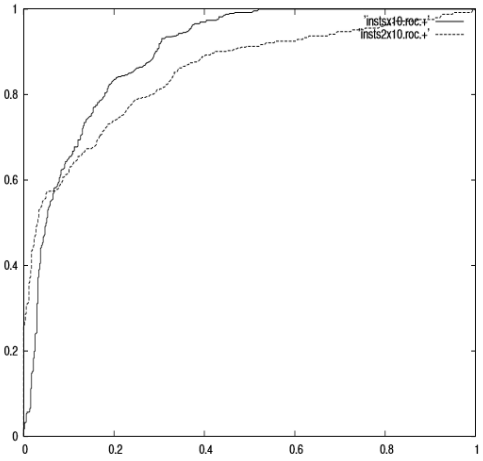
The results of the cross-validation and train/test experiments are shown in Table 5.6. Cross-validation performance is almost identical to that achieved in prior works (see for instance, (Shukri et al., 2015; Annervaz, Chowdhury, and Dukkipati, 2018; Yılmazel and Arslan, 2019)), indicating that the Overall performance is *high*, and on par with the results we have previously achieved. Cross-validation Mean is 1.0 for *science* sample. Performance on the test collection varied between a low of 70% for prior data and a high of 100% for new data - with an average of 80%. For some topics or samples, decreases in performance seen between the evaluations for each topic are in general very small. This difference is larger, with both increases and decreases in performance seen on some topics. The biggest increase in performance was observed in both prior and new data.

In Figure 5.18, the test set has a balanced 1:1 class distribution.

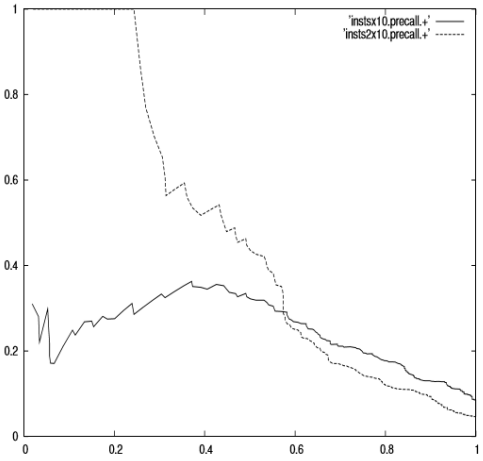


(a) ROC curves, 1:1 (b) Precision-recall curves, 1:1

FIGURE 5.18: Classifiers evaluated using ROC curves and precision-recall curves (a)



(c) ROC curves, 1:10



(d) Precision-recall curves, 1:10

FIGURE 5.19: Classifiers evaluated using ROC curves and precision-recall curves (b)

Figure 5.18 and 5.19 shows the same two classifiers (ANN and sentiStrength) on the same domain, but the number of negative instances has been increased ten-fold. We noted that the classifiers and the underlying concept has not changed - only the class distribution is different.

We also observed that the ROC graphs in Figure 5.18 and 5.19 5*a* and 5*c* are *identical*, while the Precision-Recall graphs in Figure 5.18 and 5.19 5*b* and 5*d* differ dramatically. In some cases, the conclusion of which classifier has superior performance can change with a *shifted* distribution.

5.6.3 Data correction for American and British English

How do we correct data that is not in plain American or British English? Searches are based on English keywords that the user enters into the search box. For example, entering the word *Wits* would prompt the search engine to return web pages that are related to the definition of *Wits* as a noun in some ways, and discount any other unrelated pages (Nkongolo, 2018). However, it is not always possible to convey a person's exact needs through a set number of keywords. In addition, the results obtained by a search algorithm may not be optimal. For example, if a person wanted information or videos about the University of the Witwatersrand, entering only the word *Wits* may also return other types of results, such as the meaning of *Wits* according to the Cambridge Dictionary[3] - refining this search to a phrase such as the University of the Witwatersrand *news* may eliminate a number of unrelated pages, but it is still not likely to give completely perfect results (Nkongolo, 2018). A few pages related to *Wits* synonyms, for example, may have been included in the search results.

This type of query correction required the computational framework to understand the semantic of the query that the user provide to the web crawler (Khan and Digout, 2018). Unfortunately if the user does not provide an optimal query this is likely to

[3]The capacity for inventive thought and quick understanding; keen intelligence

yield bias results due to a misunderstanding of the query *morphology* on the computer side. In NLP, the computer program utilises the *morphology* of a text to make prediction. As such, attempting to implement intelligent system capable of capturing the *semantic* of given text is a subject which requires intensive research in strong *AI*[4], the author argued.

In general the performance of the computational framework is very similar to that obtained in previous cross-validation studies (Chiranjeevi, Santosh, and Vishnuvardhan, 2019; Hirve and Reddy, 2019; Chiranjeevi, Santosh, and Vishnuvardhan, 2019). Overall, the ROC measures are high, well over 80% and 90%. We can argue that statistically, the cross-validation and prospective evaluations performed identically - this is a very encouraging result, and provides strong confirmation that the computational framework could be aided by the addition of an ML-based system to the classification process.

5.7 Experimental discussion

The objective of this experiment was to evaluate different techniques discussed in Chapter three and four. We presented results with regard to the metrics used to classify the textual data set in terms of assessing the CRA framework. An illustration of the computing environment was presented where ANN analysed the WITS marketing team dataset on an exponential level. A prolongation of iteration has been detected in terms of reanalysing the WITS dataset and the confusion matrix shows an accurate accumulation of 114 features correctly classified. The incorrect classification (FN and FP) have been reduced as the sample size increased (Nikam, 2015).

To assess the sparsity of the dataset, a confusion matrix was visualised with additional datasets (Asghar et al., 2018). This visualisation shows an increase of higher volume of TP as compared to TN and the FP - FN did not have a considerable ratio. We then argued that this was due to the fact that the content of the selected dataset was positive than negative. Following this, the BOW performance was presented where the recognition unit enabled us to study the overall polarity of the WITS dataset sample (Al Marouf, Hasan, and Mahmud, 2019). The results revealed that the content of this dataset is more neutral than positive after applying Sentistrength algorithm. The time consumption is analysed by studying the long CPU time of ANN as an indication of efficiency (Guggilla et al., 2018).

It has been found that the greater the dataset, the longer ANN takes time to accurately classify data as discussed by Tsytsarau and Palpanas (2012). This is a limitation of ANN which suffers from significant slowdowns with Big Data. The Accuracy and Empirical error are assessed - there is an improvement of Accuracy of 93% during the training process. A minimisation of the Empirical error has been detected with 0.07%. An improvement of Precision evaluation values are presented and ANN depicts a maximal value of 100% for both training and testing stages.

Therefore Accuracy is not an efficient metric to be used with regard to assessing

[4]Strong artificial intelligence (strong AI) is an artificial intelligence construct that has mental capabilities and functions that mimic the human brain. In the philosophy of strong AI, there is no essential difference between the piece of software, which is the AI, exactly emulating the actions of the human brain, and actions of a human being, including its power of understanding and even its consciousness.

ANN (Depren, Aşkın, and Öz, 2017). The Recall values are presented and the experiment yielded 86% - this demonstrated the efficiency of ANN (Nilizadeh et al., 2019). We then compared the F-measure value with the Recall value - the experiment shows that there is an augmentation of F-measure percentage during the testing phase. Additionally, the BOW results are presented with entities categorised by the NER - the Sentistrength algorithm results are also presented in a tabular format with 0.479 as the lowest polarity for the WITS dataset and 0.6369 as the highest Compound.

ROC graphs are a very useful tool for visualizing and evaluating classifiers (Vural et al., 2013). In this research they are able to provide a richer measure of classification performance than Accuracy or empirical error can, and they have advantages over other evaluation measures such as Precision-Recall graphs and lift curves. As such, as with any evaluation metric, using them wisely requires knowing their characteristics and limitations (Hariri, Emami, and Malek, 2015; Farag, Lee, and Fox, 2018). It is hoped that this research advances the general knowledge about ROC graphs and helps to promote better evaluation practices in the Data Analytic community.

5.8 Discussion in relation to the research question

Having completed this experimentation, the research question set out in Chapter 1 can now be answered:

1. Is publicly available data suitable to address the problem of Reputation Analysis? This research demonstrates that publicly available data can be used to address the problem of Corporate Reputation Analysis by using ANN, Sentistrength, BOW, and NER. These algorithms improve the overall classification Accuracy and reduced misclassification significantly (Vural et al., 2013).

2. Sentiment Analysis is used to address the problem of implementing techniques for predicting the polarity scores of web pages.

3. The development of automatic textual data collection in real time was implemented by a web crawler. NER and BOW algorithms were used to improve the Feature Extraction while ANN was used to improve the classification Accuracy (Jamal et al., 2018).

4. We used different metrics such as Precision and Recall to evaluate the proposed framework systematically (Hariri, Emami, and Malek, 2015).

5. Lastly, our results outperformed results obtained by Shukri et al. (2015), Jadav and Vaghela (2016), and Rasool et al. (2019).

5.9 Conclusion

The aim of this Chapter was to assess and discuss results obtained during the experimentation. To accomplish this, it was of paramount importance to assess the completeness and efficiency of the reputational framework introduced in Chapter one by comparing ANN, BOW, NER, and Sentistrength algorithm with regard to Precision, Accuracy, F-measure, Recall, word counting and the sentiment score. In this Chapter, we explored and analysed different techniques to derive meaning from textual data in terms of producing an accurate Corporate Reputation Analysis. Results revealed that BOW and NER are efficient for Feature Extraction and Sentistrength predictions are of great Precision. In the next Chapter (Chapter six), we conclude this research and discuss challenges faced in Corporate Reputation Analysis - we also present future works in Chapter six.

Chapter 6

Discussion and future works

Donneau-Golencer et al. (2019) argued that the availability of textual data on the internet makes it possible to study users opinions on various services offered by certain corporations or brands. In this perspective, entrepreneurs need to receive feedback from customers in order to improve their services (Brabazon, 2016). This will allow them to improve the quality of services provided. The challenge is that the Big Data available on the internet makes it difficult to detect user opinions (Depren, Aşkın, and Öz, 2017). This difficulty is demonstrated by the inability of classifiers to produce accurate results during the testing process (Shukri et al., 2015).

Therefore, this research attempts to solve problems related to processing publicly and existing available data for a Reputation Analysis of a company based on user opinion. In addition, we have implemented text extraction and document based on pre-processed textual data via ML and NLP techniques (Nikam, 2015).

Additionally, this dissertation also introduces an approach to use computational methods to analyse the web presence of an entity to establish their reputation (Musto et al., 2015). In this case, Sentiment Analysis was used as a proxy to assess the reputation of the University of the Witwatersrand on the internet. The use of ANN, Sentistrength, NER, and BOW algorithms have been presented, implemented, and analysed in this research. The computational approach objectives is to develop robust classification techniques that would be more accurate and minimise misclassification (Paramita et al., 2019). The textual dataset used for experimentation is defined - selected classifiers are used to determine the sentiment polarity of a given text extracted form the internet (Fu et al., 2012).

Existing datasets as well as datasets that were extracted manually were used in the experimentation. The methodology relied on training selected classifiers with balanced features having different subsets of textual features (Taboada, 2016). In the training phase, we used different subsets iteratively. In general, this work showed that FN and FP can be reduced to improve Accuracy and Precision in Corporate Reputation Analysis and our results outperformed results presented by Jadav and Vaghela (2016) and Rasool et al. (2019). This research is completely different from Jadav and Vaghela (2016) and Rasool et al. (2019)'s work with regards to the techniques used in terms of Information Retrieval, Feature Extraction, and Feature Visualisation. Jadav and Vaghela (2016) and Rasool et al. (2019) used SVM, PCA, and DT. However, these algorithms did not classify features with great Precision. Most of the ML techniques utilised by Jadav and Vaghela (2016) and Rasool et al. (2019) could not minimise the misclassification rate - this may be due to imbalanced data and a difference in feature characteristics.

The computational framework presented in this work has improved Accuracy and Precision percentage by reducing FN and FP rate. A Big Data of textual features was used during the training and testing process - this enabled us to perform substantial experiments. In the first experiment, we used balanced features having the same size and ANN yielded an Accuracy of 93% and 100% of Precision (Huang, Capretz, and Ho, 2019). This work also demonstrated that Accuracy is not suitable to assess classifiers in Reputation Analysis as discussed by Kim, Youn, and Lee (2019) - this was due to the fact that the percentage produced by the Precision metric outperformed Accuracy. Therefore, Precision is the most suitable metric that depicts the computational framework completeness. Similarly, the Recall values were presented with 86% - this also demonstrated classifiers classification efficiencies (Gardner et al., 2018).

A comparison study between F-measure and Recall performance revealed that F-measure increased in value during the training and testing process - this demonstrated that classifiers are able to learn from publicly available data whether they are structured or unstructured (Bottou, Curtis, and Nocedal, 2018). Feature Extraction using NER and BOW are quite satisfactory - the NER was able to categorise entities and the BOW produced word counting which was visualized during the Data Mining process.

Similarly, results produced by ANN outperformed Shukri et al. (2015) results where, NB were used to study the reputation of Mercedes and Audi - the polarity classification has 83% which is much lower than our results. Finally, this research has shown that combining ANN, Sentistrenghth, NER, and BOW algorithms can improve the prediction and detection Accuracy of classifiers in Reputation Analysis. The robustness of selected algorithms have been tested in real-time on Clueweb09 dataset and WITS marketing team datasets. Based on the results obtained, the classification prediction Precision is much better on both datasets and provides details in terms of Reputation Analysis.

6.1 Challenges in Corporate Reputation Analysis

Corporate Reputation Analysis is complex. Some of the issues in Corporate Reputation Analysis are listed below:

1. The implementation of new recognition techniques and features to enhance performance Accuracy for other languages such as French or Spanish (Vural et al., 2013).

2. The development of semi-supervised methods to enhance the adaptability of classifiers in real time (Zuo, Zhao, and Xu, 2016).

3. Create standard GUI, which will improve the quality of the visualisation unit.

4. Implement parallel web crawling to improve Data Collection and Feature Extraction.

5. Corporate Reputation Analysis requires reducing misclassification and ameliorating the classification coverage.

6. Insufficient datasets (Nkongolo, 2018).

7. Protection of personal information - AI carries with it many ethical and social ramifications. Therefore, the condition of the lawful processing of personal information is of paramount importance.

8. Classifiers performance might differ due to the dynamic structure of the internet - crawling is based on keywords used, thus affecting the prediction performance of classifiers when the query is not optimal.

Overall, both algorithms used in this research can be combined to address the problem of Corporate Reputation Analysis.

6.2 Future work

This research attempted to develop a reputational framework based on supervised learning. Moreover, this research might be extended by improving certain aspects of the computational framework. Below is a list of suggestions for a promising CRA framework:

1. K Nearest Neighbor (KNN) and fuzzy search algorithms can also be compared to ML techniques presented in this research (Kent, 2014).

2. The identification of new features capable of improving the performance of classifiers is another possibility for future research. These features might be pictures or videos.

3. The framework can be expanded to other contexts, such as a demographic driven web crawler to optimise the amount of relevant content crawled (Seth et al., 2019).

We truly believe that the proposed computational framework could be used as a commercial sentimental search engine.

Bibliography

Agichtein, Eugene, Eric Brill, and Susan Dumais (2006). "Improving web search ranking by incorporating user behavior information". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 19–26.

Al Marouf, Ahmed, Md Kamrul Hasan, and Hasan Mahmud (2019). "Identifying Neuroticism from User Generated Content of Social Media based on Psycholinguistic Cues". In: *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, pp. 1–5.

Almatarneh, Sattam and Pablo Gamallo (2017). "Automatic construction of domain-specific sentiment lexicons for polarity classification". In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, pp. 175–182.

Annervaz, KM, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati (2018). "Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing". In: *arXiv preprint arXiv:1802.05930*.

Arif-Uz-Zaman, Kazi et al. (2017). "Extracting failure time data from industrial maintenance records using text mining". In: *Advanced Engineering Informatics* 33, pp. 388–396.

Arpat, Guven Burc, Saiyad Shah, and Srikant Ramakrishna Ayyar (2018). *Sentiment polarity for users of a social networking system*. US Patent App. 15/693,235.

Asghar, Muhammad Zubair et al. (2018). "T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme". In: *Expert Systems* 35.1, e12233.

Baradwaj, Brijesh Kumar and Saurabh Pal (2012). "Mining educational data to analyze students' performance". In: *arXiv preprint arXiv:1201.3417*.

Blanco, Eduardo and Dan I Moldovan (2011). "Some Issues on Detecting Negation from Text." In: *FLAIRS Conference*, pp. 228–233.

Boit, James and Omar El-Gayar (2019). "Topical Mining of Malaria Using Social Media. A Text Mining Approach". In:

Bottou, Léon, Frank E Curtis, and Jorge Nocedal (2018). "Optimization methods for large-scale machine learning". In: *Siam Review* 60.2, pp. 223–311.

Brabazon, Tara (2016). *The University of Google: Education in the (post) information age*. Routledge.

Calefato, Fabio et al. (2018). "Sentiment polarity detection for software development". In: *Empirical Software Engineering* 23.3, pp. 1352–1382.

Chen, Juan et al. (2019). "Trust architecture and reputation evaluation for internet of things". In: *Journal of Ambient Intelligence and Humanized Computing* 10.8, pp. 3099–3107.

Chiranjeevi, P, D Teja Santosh, and B Vishnuvardhan (2019). "Survey on Sentiment Analysis Methods for Reputation Evaluation". In: *Cognitive Informatics and Soft Computing*. Springer, pp. 53–66.

Cho, Junghoo and Hector Garcia-Molina (2002). "Parallel crawlers". In: *Proceedings of the 11th international conference on World Wide Web*. ACM, pp. 124–135.

Chou, Yi-Chi, Chun-Yen Chao, and Han-Yen Yu (2019). "A Résumé Evaluation System Based on Text Mining". In: *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, pp. 052–057.

Confente, Ilenia et al. (2019). "Effects of data breaches from user-generated content: A corporate reputation analysis". In: *European Management Journal*.

Das, M Swami, A Govardhan, and D Vijaya Lakshmi (2019). "Classification of web services using data mining algorithms and improved learning model." In: *Telkomnika* 17.6.

Depren, Serpil Kılıç, Öyküm Esra Aşkın, and Ersoy Öz (2017). "Identifying the classification performances of educational data mining methods: a case study for TIMSS". In: *Educ Sci Theory Pract* 17.5, pp. 1605–1623.

Donneau-Golencer, Thierry et al. (2019). *Tools and techniques for extracting knowledge from unstructured data retrieved from personal data sources*. US Patent App. 16/165,242.

Duggan, Maeve et al. (2015). "Social media update 2014". In: *Pew research center* 19.

Farag, Mohamed MG, Sunshin Lee, and Edward A Fox (2018). "Focused crawler for events". In: *International Journal on Digital Libraries* 19.1, pp. 3–19.

Fu, Tianjun et al. (2012). "Sentimental spidering: leveraging opinion information in focused crawlers". In: *ACM Transactions on Information Systems (TOIS)* 30.4, p. 24.

Gardner, Matt et al. (2018). "Allennlp: A deep semantic natural language processing platform". In: *arXiv preprint arXiv:1803.07640*.

Gasc, Luis et al. (2019). "Noytext: A Web platform to annotate social media documents on noise perception for their use in opinion mining research". In: *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*. Vol. 259. 8. Institute of Noise Control Engineering, pp. 1779–1786.

Giachanou, Anastasia, Julio Gonzalo, and Fabio Crestani (2019). "Propagating sentiment signals for estimating reputation polarity". In: *Information Processing & Management* 56.6, p. 102079.

Giammarco, Kristin and Kathleen Giles (2018). "Verification and validation of behavior models using lightweight formal methods". In: *Disciplinary convergence in systems engineering research*. Springer, pp. 431–447.

Gonçalves, Pollyanna et al. (2013). "Comparing and combining sentiment analysis methods". In: *Proceedings of the first ACM conference on Online social networks*. ACM, pp. 27–38.

Goswami, Sanjay, Satrajit Nandi, and Sucheta Chatterjee (2019). "Sentiment Analysis Based Potential Customer Base Identification in Social Media". In: *Contemporary Advances in Innovative and Applicable Information Technology*. Springer, pp. 237–243.

Guggilla, Chinnappa et al. (2018). "Mining entities and their values from semi-structured documents in business process outsourcing". In: *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. ACM, pp. 282–288.

Haixiang, Guo et al. (2017). "Learning from class-imbalanced data: Review of methods and applications". In: *Expert Systems with Applications* 73, pp. 220–239.

Hanson, Sara, Lan Jiang, and Darren Dahl (2019). "Enhancing consumer engagement in an online brand community via user reputation signals: a multi-method analysis". In: *Journal of the Academy of Marketing Science* 47.2, pp. 349–367.

Hariri, Nadjla, Zahra Emami, and Mojtaba Malek (2015). "The precision and recall of general search engines in retreival of images related to endocrine diseases". In: *Iranian Journal of Endocrinology and Metabolism* 17.2, pp. 97–104.

Hirve, Sumit and CH Pradeep Reddy (2019). "A Survey on Visualization Techniques Used for Big Data Analytics". In: *Advances in Computer Communication and Computational Sciences*. Springer, pp. 447–459.

Hong, Sim-Hee, Seul-Ki Lee, and Jung-Ho Yu (2019). "Automated management of green building material information using web crawling and ontology". In: *Automation in Construction* 102, pp. 230–244.

Hothorn, Torsten (2018). "CRAN task view: Machine learning & statistical learning". In:

Huang, Yanrong and Min Chen (2019). "Improve Reputation Evaluation of Crowdsourcing Participants Using Multidimensional Index and Machine Learning Techniques". In: *IEEE Access* 7, pp. 118055–118067.

Huang, Yuxuan, Luiz Fernando Capretz, and Danny Ho (2019). "Neural Network Models for Stock Selection Based on Fundamental Analysis". In: *arXiv preprint arXiv:1906.05327*.

Hussein, Doaa Mohey El-Din Mohamed (2018). "A survey on sentiment analysis challenges". In: *Journal of King Saud University-Engineering Sciences* 30.4, pp. 330–338.

Ingre, Bhupendra and Anamika Yadav (2015). "Performance analysis of NSL-KDD dataset using ANN". In: *2015 International Conference on Signal Processing and Communication Engineering Systems*. IEEE, pp. 92–96.

Islam, Md Rakibul and Minhaz F Zibran (2018). "SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text". In: *Journal of Systems and Software* 145, pp. 125–146.

Jadav, Bhumika M and Vimalkumar B Vaghela (2016). "Sentiment analysis using support vector machine based on feature selection and semantic analysis". In: *International Journal of Computer Applications* 146.13.

Jamal, Salma et al. (2018). "Machine Learning: What, Why, and How?" In: *Bioinformatics: Sequences, Structures, Phylogeny*. Springer, pp. 359–374.

Jo, Bogeun, KyungBae Park, and Sung Ho Ha (2019). "The BRAVO: A Framework of Building Reputation Analytics from Voice Online". In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, pp. 777–790.

Jones, Brian, John Temperley, and Anderson Lima (2009). "Corporate reputation in the era of Web 2.0: the case of Primark". In: *Journal of marketing management* 25.9-10, pp. 927–939.

Keim, Daniel A (2002). "Information visualization and visual data mining". In: *IEEE Transactions on Visualization & Computer Graphics* 1, pp. 1–8.

Kennedy, Alistair and Diana Inkpen (2006). "Sentiment classification of movie reviews using contextual valence shifters". In: *Computational intelligence* 22.2, pp. 110–125.

Kent, Allen (2014). *Encyclopedia of Library and Information Science Volume 35*. Crc Press.

Khan, Salman and Jacques Digout (2018). "The corporate reputation reporting framework (CRRF)". In: *Corporate Reputation Review* 21.1, pp. 22–36.

Kim, Hyuksoo, Seounmi Youn, and Doohwang Lee (2019). "The effect of corporate social responsibility reputation on consumer support for cause-related marketing". In: *Total Quality Management & Business Excellence* 30.5-6, pp. 682–707.

Kiritchenko, Svetlana and Saif M Mohammad (2018). "Sentiment composition of words with opposing polarities". In: *arXiv preprint arXiv:1805.04542*.

Koncz, Peter and Jan Paralic (2011). "An approach to feature selection for sentiment analysis". In: *Intelligent Engineering Systems (INES), 2011 15th IEEE International Conference on*. IEEE, pp. 357–362.

Konstantinova, Natalia and Sheila CM De Sousa (2011). "Annotating negation and speculation: the case of the review domain". In: *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pp. 139–144.

Kosala, Raymond and Hendrik Blockeel (2000). "Web mining research: A survey". In: *ACM Sigkdd Explorations Newsletter* 2.1, pp. 1–15.

Krahel, John Peter and William R Titera (2015). "Consequences of Big Data and formalization on accounting and auditing standards". In: *Accounting Horizons* 29.2, pp. 409–422.

Kumar, Akshi and Dhiraj Kumar Gupta (2019). "Empirical Analysis of Supervised Machine Learning Techniques for Expert Mining in CQA System". In: *International Journal of Information Systems & Management Science* 2.2.

Kumar, Manish et al. (2018). "Keyword query based focused Web crawler". In: *Procedia Computer Science* 125, pp. 584–590.

Lam, Sophia et al. (2019). "Optimizing Customer-Agent Interactions with Natural Language Processing and Machine Learning". In: *2019 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, pp. 1–6.

Lewandowski, Mateusz (2020). "Public organizations and business model innovation: The role of public service design". In: *Disruptive Technology: Concepts, Methodologies, Tools, and Applications*. IGI Global, pp. 1404–1428.

Li, Zuhe et al. (2019). "A survey on sentiment analysis and opinion mining for social multimedia". In: *Multimedia Tools and Applications* 78.6, pp. 6939–6967.

Lin, Bin et al. (2018). "Sentiment Analysis for Software Engineering: How Far Can We Go?" In: *Proceedings of 40th International Conference on Software Engineering*.

Lin, Bin et al. (2019). "Pattern-based mining of opinions in Q&A websites". In: *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press, pp. 548–559.

Lu, Yue and Chengxiang Zhai (2008). "Opinion integration through semi-supervised topic modeling". In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 121–130.

Lukyanenko, Roman et al. (2019). "Expecting the unexpected: effects of data collection design choices on the quality of crowdsourced user-generated content". In: *MIS Quarterly* 43.2, pp. 623–647.

Luo, Xiong et al. (2018). "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy". In: *Journal of The Franklin Institute* 355.4, pp. 1945–1966.

Maerz, Seraphine F and Cornelius Puschmann (2020). "Text as Data for Conflict Research: A Literature Survey". In: *Computational Conflict Research*. Springer, pp. 43–65.

Makhzani, Alireza et al. (2015). "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644*.

Mall, Rajib (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd.

Massucci, Francesco Alessandro and Domingo Docampo (2019). "Measuring the academic reputation through citation networks via PageRank". In: *Journal of Informetrics* 13.1, pp. 185–201.

Mehdad, Yashar, Aasish Pappu, and Amanda Stent (2018). *Scalable Multilingual Named-Entity Recognition*. US Patent App. 15/406,586.

Mescheder, Lars, Sebastian Nowozin, and Andreas Geiger (2017). "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks". In: *arXiv preprint arXiv:1701.04722*.

Michalski, Ryszard S, Jaime G Carbonell, and Tom M Mitchell (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

Moraes, Rodrigo, JoãO Francisco Valiati, and Wilson P GaviãO Neto (2013). "Document-level sentiment classification: An empirical comparison between SVM and ANN". In: *Expert Systems with Applications* 40.2, pp. 621–633.

Morinaga, Satoshi et al. (2002). "Mining product reputations on the web". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 341–349.

Musto, Cataldo et al. (2015). "CrowdPulse: A framework for real-time semantic analysis of social streams". In: *Information Systems* 54, pp. 127–146.

Narwekar, Abhishek Avinash (2018). "Affective analysis of text in tweets". PhD thesis.

Nasukawa, Tetsuya and Jeonghee Yi (2003). "Sentiment analysis: Capturing favorability using natural language processing". In: *Proceedings of the 2nd international conference on Knowledge capture*. ACM, pp. 70–77.

Nicola, Raluca Georgeta (2013). "Categorization and visualization of Twitter data". PhD thesis. Technical University of Dresden.

Nikam, Sagar S (2015). "A comparative study of classification techniques in data mining algorithms". In: *Oriental journal of computer science & technology* 8.1, pp. 13–19.

Nilizadeh, Shirin et al. (2019). "Think Outside the Dataset: Finding Fraudulent Reviews using Cross-Dataset Analysis". In: *The World Wide Web Conference*. ACM, pp. 3108–3115.

Nkongolo, Mike (2018). "Classifying Searh Results Using Neural Networks and Anomaly Detection". In: *Educor Multidisciplinary Journal* 2.1, pp. 102–127.

Nyman, Rickard et al. (2018). "News and narratives in financial systems: exploiting big data for systemic risk assessment". In:

O'Leary, Daniel E (2013). "Artificial intelligence and big data". In: *IEEE Intelligent Systems* 28.2, pp. 96–99.

Oshikawa, Ray, Jing Qian, and William Yang Wang (2018). "A survey on natural language processing for fake news detection". In: *arXiv preprint arXiv:1811.00770*.

Paramita, Monica Lestari et al. (2019). "Collecting Comparable Corpora". In: *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*. Springer, pp. 55–87.

Peters, Michael A (2019). *Global university rankings: Metrics, performance, governance*.

Ramanathan, Vallikannu and T Meyyappan (2019). "Twitter Text Mining for Sentiment Analysis on People's Feedback about Oman Tourism". In: *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*. IEEE, pp. 1–5.

Rasool, Abdur et al. (2019). "Twitter Sentiment Analysis: A Case Study for Apparel Brands". In: *Journal of Physics: Conference Series*. Vol. 1176. 2. IOP Publishing, p. 022015.

Rawat, Keshav Singh and IV Malhan (2019). "A Hybrid Classification Method Based on Machine Learning Classifiers to Predict Performance in Educational Data Mining". In: *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer, pp. 677–684.

Redhu, Swati et al. (2018). "Sentiment analysis using text mining: a review". In: *International Journal on Data Science and Technology* 4.2, p. 49.

Rehman, Anwar Ur et al. (2019). "A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis". In: *Multimedia Tools and Applications*, pp. 1–17.

Rout, Jitendra Kumar et al. (2018). "A model for sentiment and emotion analysis of unstructured social media text". In: *Electronic Commerce Research* 18.1, pp. 181–199.

Sänger, Johannes and Günther Pernul (2018). "Interactive Reputation Systems". In: *Business & Information Systems Engineering* 60.4, pp. 273–287.

Schwaiger, Manfred (2004). "Components and parameters of corporate reputation—An empirical study". In: *Schmalenbach business review* 56.1, pp. 46–71.

Seth, Neeraj et al. (2019). "Comparative Analysis of Major Jacobian and Gradient Backpropagation Optimizers of ANN on SVPWM". In: *Soft Computing and Signal Processing*. Springer, pp. 345–357.

Shekyan, Sergey et al. (2018). *Securing web page content*. US Patent App. 15/791,291.

Shukri, Sarah E et al. (2015). "Twitter sentiment analysis: A case study in the automotive industry". In: *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*. IEEE, pp. 1–5.

Stuart, Keith Douglas and Maciej Majewski (2015). "Intelligent opinion mining and sentiment analysis using artificial neural networks". In: *International Conference on Neural Information Processing*. Springer, pp. 103–110.

Sugiyama, Yuji (2019). *Analysis of evaluations from internet media*. US Patent App. 10/185,915.

Sundaresan, Neelakantan et al. (2019). *System and method for topic extraction and opinion mining*. US Patent App. 10/339,184.

Szárnyas, Gábor et al. (2018). "The Train Benchmark: cross-technology performance evaluation of continuous model queries". In: *Software & Systems Modeling* 17.4, pp. 1365–1393.

Tabassum, Nusrath and Muhammad Ibrahim Khan (2019). "Design an Empirical Framework for Sentiment Analysis from Bangla Text using Machine Learning". In: *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, pp. 1–5.

Taboada, Maite (2016). "Sentiment analysis: an overview from linguistics". In:

Thelwall, Mike, Kevan Buckley, and Georgios Paltoglou (2012). "Sentiment strength detection for the social web". In: *Journal of the American Society for Information Science and Technology* 63.1, pp. 163–173.

Tsai, Chun-Wei et al. (2013). "Data mining for internet of things: A survey". In: *IEEE Communications Surveys & Tutorials* 16.1, pp. 77–97.

Tsytsarau, Mikalai and Themis Palpanas (2012). "Survey on mining subjective data on the web". In: *Data Mining and Knowledge Discovery* 24.3, pp. 478–514.

Unnisa, Muqtar, Ayesha Ameen, and Syed Raziuddin (2016). "Opinion mining on Twitter data using unsupervised learning technique". In: *International Journal of Computer Applications* 148.12, pp. 975–8887.

Unwin, Antony (2000). "Visualisation for data mining". In: *International Conference on Data Mining, Visualization and Statistical System, Séoul, Korea*.

Volcani, Yanon and David Fogel (2006). *System and method for determining and controlling the impact of text*. US Patent 7,136,877.

Vural, A Gural et al. (2013). "A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish". In: *Computer and Information Sciences III*. Springer, pp. 437–445.

Wang, Hao et al. (2012). "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle". In: *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, pp. 115–120.

Wang, Yanshan et al. (2018). "A comparison of word embeddings for the biomedical natural language processing". In: *Journal of biomedical informatics* 87, pp. 12–20.

Wang, Yinying and David J Fikis (2019). "Common core state standards on Twitter: Public sentiment and opinion leaders". In: *Educational Policy* 33.4, pp. 650–683.

Webster, Thomas J (2001). "A principal component analysis of the US News & World Report tier rankings of colleges and universities". In: *Economics of Education Review* 20.3, pp. 235–244.

Welbl, Johannes, Pontus Stenetorp, and Sebastian Riedel (2018). "Constructing datasets for multi-hop reading comprehension across documents". In: *Transactions of the Association of Computational Linguistics* 6, pp. 287–302.

Xianghua, Fu et al. (2004). "Focused crawling method with online-incremental adaptive learning". In: *JOURNAL-XIAN JIAOTONG UNIVERSITY.* 38.6, pp. 599–602.

Xu, Keyang, Kyle Yingkai Gao, and Jamie Callan (2018). "A Structure-Oriented Unsupervised Crawling Strategy for Social Media Sites". In: *arXiv preprint arXiv:1804.02734*.

Yılmazel, İbrahim Barış and Ahmet Arslan (2019). "An intrinsic evaluation of the Waterloo spam rankings of the ClueWeb09 and ClueWeb12 datasets". In: *Journal of Information Science*, p. 0165551519866551.

Young, Tom et al. (2018). "Recent trends in deep learning based natural language processing". In: *ieee Computational intelligenCe magazine* 13.3, pp. 55–75.

Zhang, Na et al. (2019). "Exploring the sources of contractors' competitive advantage on international HSR construction projects". In: *International Journal of Civil Engineering* 17.7, pp. 1115–1129.

Zhang, Zhihua (2018). "Artificial neural network". In: *Multivariate Time Series Analysis in Climate and Environmental Research*. Springer, pp. 1–35.

Zheng, Lijuan, Hongwei Wang, and Song Gao (2018). "Sentimental feature selection for sentiment analysis of Chinese online reviews". In: *International journal of machine learning and cybernetics* 9.1, pp. 75–84.

Zishumba, Kudzai (2019). "Sentiment Analysis Based on Social Media Data". PhD thesis.

Zuo, Yuan, Jichang Zhao, and Ke Xu (2016). "Word network topic model: a simple but general solution for short and imbalanced texts". In: *Knowledge and Information Systems* 48.2, pp. 379–398.