PREDICTING USER PERSONALITY FROM PUBLIC PERCEPTIONS ON SOCIAL MEDIA

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF COMPUTER AND INFORMATION SCIENCES

Supervisors
Dr. Muhammad Asif Naeem
Dr. Farhaan Mirza

April 2019

By
Adi Darliansyah
School of Engineering, Computer and Mathematical Sciences

Copyright

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Acknowledgements

All praises to Allah for the strengths and His blessing in completing this thesis. I would like to express my sincere gratitude to the people who made more than their share of contributions. Special appreciation goes to my supervisors, Dr. M. Asif Naeem and Dr. Farhaan Mirza, for their invaluable advice and constructive comments throughout the course of the study. Your guidance incredibly empowered me to have the confidence to move ahead with my project. I humbly thank you.

This research would not have been possible without the financial support of the New Zealand's Ministry of Foreign Affairs and Trade, for which I am very grateful. I would also like to acknowledge my lecturers, for their dedication in imparting their knowledge to me, and to the Scholarships Office team, for making my journey through the university a pleasant one.

My heartfelt and deepest gratitude is extended to my family members and friends, for their utmost moral support and prayers during the ups and downs of my study. I truly appreciate the friendship I have received from my best friends. Thank you for the many trips down memory lane.

I would like to dedicate this thesis to my dear parents, Sarmili Hidayat and Sri Idayati, who taught me the merits of patience and perseverance, and to my younger brother Bentar Choirul Huda, who offered humour as the best form of help. Finally, I dedicate this thesis with love in memory of my late sister,

Wulan Anggraeni, who really encouraged me to follow my dreams and reach for opportunities that seemed unattainable. I know she would be so proud of what I have accomplished.

Abstract

Personality distinctively characterises an individual and profoundly influences behaviours. Social media offer the virtual community an unprecedented opportunity to generate content and share aspects of their life which often reflect their personalities. The interest in using deep learning to infer traits from digital footprints has grown recently; however, very limited work has been presented which explores the sentiment information conveyed. The present study, therefore, used a computational approach to classify personality from social media by gauging public perceptions underlying factors encompassing traits.

In the research reported in this thesis, a Sentiment-based Personality Detection system was developed to infer trait from short texts based on the 'Big Five' personality dimensions. We exploited the spirit of Neural Network Language Model (NNLM) by using a unified model that combines a Recurrent Neural Network named Long Short-Term Memory (LSTM) with a Convolutional Neural Network (CNN). The proposed system is threefold: It commences with sentiment classification by grouping short messages harvested online into three categories, namely positive, negative, and nonpartisan. This is followed by employing Global Vectors (GloVe) to build vectorial word representations. As such, this step aims to add external knowledge to short texts. We apply CNN and LSTM during the learning process. Finally, we trained each variant of the models to

compute prediction scores across the five traits. Experimental study indicated the effectiveness of our system.

As part of our investigation, a case study was carried out which employed the proposed system. We opted for Uber, a renowned global hail-sharing company, as the subject of our examination. The selected study was set up to investigate the existing correlation of personality traits and opinion polarities. The results support the prior findings of the tendency of persons with the same traits to express sentiments in similar ways.

Publications

Long-term trends in public sentiment in Indian demonetisation policy

Darliansyah, A., Wandabwa, H. M., Naeem, M. A., Mirza, F. & Pears, R. In: I. S. Bajwa, F. Kamareddine & A. Costa (Eds.), *Proceeding of the International Conference on Intelligent Technologies and Applications* (INTAP), pp. 65-75. Singapore: Springer, 2018.

SENTIPEDE: A smart system for sentiment-based personality detection from short texts

Darliansyah, A., Naeem, M. A., Mirza, F. & Pears, R. Article accepted for publication in the special issue on Intelligent Computing for Society, *Journal of Universal Computer Science*, 2019.

Contents

List of Tables

List of Figures

Glossary

16PF	Sixteen Personality Factors. 42
AMI	Amazon Machine Image. 81
ANN	Artificial Neural Network. 30
API	Application Programming Interface. 52, 55, 56, 83
ASCII	American Standard Code for Information Interchange. 58
BFI	Big Five Inventory. 43, 107, 108
CBOW	Continuous Bag-of-Words. 36, 37
CD	Continuous Delivery. 70, 80, 83
CNN	Convolutional Neural Network. 21, 23, 24, 31, 32, 47, 48, 52, 53, 64, 66, 87, 91–94, 100, 103, 111, 112
EBS	Amazon Elastic Block Store. 81
EC2	Amazon Elastic Compute Cloud. 73, 81, 82, 84
FFM	Five-Factor Model. 20, 43, 58
FFN	Feed-Forward Network. 32–34
GloVe	Global Vectors. 23, 36–38, 64, 66, 87, 91–94, 100, 103, 104, 111, 112
HTML	Hypertext Markup Language. 58–60, 65
HTTP	Hypertext Transfer Protocol. 77, 81, 83
IoT	Internet of Things. 38
LIWC	Linguistic Inquiry and Word Count. 20, 44
LSTM	Long Short-Term Memory. 23, 24, 35, 47, 48, 52, 64, 66, 87, 91–94, 100, 103, 111, 112

MPQA	Multi-Perspective Question Answering. 40
MSE	Mean Squared Error. 45
MVC	Model View Controller. 71, 72, 81
NBC	Naive Bayes Classifier. 40, 41, 45
NLP	Natural Language Processing. 21, 28, 30, 32, 38, 47, 87
NLTK	Natural Language Toolkit. 52, 61
NNLM	Neural Network Language Model. 23, 24, 30, 31, 35, 50, 87, 110, 112
OCEAN	Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism. 20, 43, 101
OS	Operating System. 80, 81, 84
PERSOMA	Personality Prediction in Social Media Data. 45, 68
REST	Representational State Transfer. 56
RNN	Recurrent Neural Network. 31, 33, 34, 48, 53
RWD	Responsive Web Design. 73, 74
SENTIPEDE	Sentiment-based Personality Detection. 23, 25, 65, 68, 74, 97, 100, 102, 111, 112
SoC	Separation of Concerns. 71
SSH	Secure Shell. 81
STS	Stanford Twitter Sentiment. 25, 40, 55, 56, 61, 63, 90, 110
SVM	Support Vector Machine. 40, 41, 44, 45, 53
TDD	Test-Driven Development. 69
TSA	Twitter Sentiment Analysis. 41
UI	User Interface. 71, 73
URL	Uniform Resource Locator. 97
UTF-8	Universal Character Set Transformation Format 8-bit. 59
VADER	Valence Aware Dictionary and Sentiment Reasoner. 63, 86, 90, 91, 94, 98
WSD	Word Sense Disambiguation. 40

Chapter 1

Introduction

For decades, research instruments such as surveys, questionnaires, and interviews have been standard practice for eliciting information from the public. It was not until the early 2000s, with the widespread presence of the social web, that researchers began to extract the digital footprint of users to infer public perceptions and user behaviours (Dave, Lawrence & Pennock, 2003; Yi, Nasukawa, Bunescu & Niblack, 2003). The advent of social media has opened a promising new avenue for multidisciplinary researchers to collect a networked, massive amount of openly available data. Social media, including a wide variety of technologies from social networking sites such as Facebook[1] to microblogging services such as Twitter,[2] allow individuals to interact and engage with user-generated content in virtual spaces (Fuchs, 2013). Hence, social media are presently considered a prominent source for comments, opinionated texts, feedback, and emotional expressions.

Twitter, for instance, with nearly 330 million active users, has become one of the most popular social media platforms today (Kemp, 2018), providing

[1] https://www.facebook.com/
[2] https://twitter.com/

services focused on short updates called *tweets*. This microblogging site has been shaping the social media landscape since its launch in 2006. Twitter has promoted the *hashtag*, a string of characters preceded by the number sign, as a way to label and categorise tweets. Through this social tag, one can quickly find messages on relevant topics. Given the *#Uber* in a tweet, any person searches for related information will retrieve up-to-the-minute news about Uber.[3] In such a situation, Twitter has given the *Twitterverse*—as the collective of its heterogeneous customers—a medium to broadcast their opinions towards the global renowned ride-sharing company, making it a prolific space of real-time feedback.

Since a customer's view is valuable and can directly affect the brand's image and loyalty, gathering social media data thus has become more prominent. To such an extent that, with regards to users and content, sentiment analysis is one such powerful tool. In the past few years, with the emergence of data-driven approaches, sentiment analysis has had much more visibility. Its application has greatly increased across various sectors. In the government sector, for example, sentiment analysis has been employed for examining public policy (Darliansyah, Wandabwa, Naeem, Mirza & Pears, 2019) and predicting electoral results (Bermingham & Smeaton, 2011; Joyce & Deng, 2017). Similarly, in the commercial area, such an analysis can help business to derive valuable insights into their products or services according to the wisdom of the crowd (Nam, Joshi & Kannan, 2017).

The open nature of social media in which users can contribute and share interests has also made its platforms a flourishing space of personal expression. Online communities subconsciously share aspects of their real life. This

[3] https://www.uber.com

often includes thoughts, feelings, and behaviour, which signal their personalities (Carducci, Rizzo, Monti, Palumbo & Morisio, 2018). Referring to the combination of the aforementioned characteristics, personality defines a unique individual. A person hence can be described as shy, open, or friendly as determined by a relatively stable features called *traits*. Among the available measurements, the Five-Factor Model (FFM) (McCrae & John, 1992) emerged as the most broadly accepted personality traits model today. Each trait describes an individual's personality over five dimensions, namely *Openness, Conscientiousness, Extroversion, Agreeableness*, and *Neuroticism*, also known as the Big Five, or the OCEAN.

A strong relationship between the Big Five traits and the number of connections in social media has been reported in the literature (Golbeck, Robles, Edmondson & Turner, 2011; Schrammel, Hochleitner & Tscheligi, 2009). Individuals with higher Openness, Extroversion, and Agreeableness, for example, tended to have more friends in Facebook and persistently maintain the connections (Rosen et al., as cited in Carducci et al., 2018). Interestingly, the findings observed in the literature mirror the actual conditions in real life. This observation has led to extensive studies on personality prediction from self-authored text posted online.

The prior modelling on trait inference from social media was dominated by algorithms on word usage patterns recognition. Among these are the Linguistic Inquiry and Word Count (LIWC) (Tausczik & Pennebaker, 2010), a transparent text count analysis program that counts words in psychology-relevant categories. Studies have shown that the LIWC categories correspond with the FFM (Golbeck et al., 2011; Schwartz et al., 2013; Sumner, Byers, Boochever & Park, 2012). However, this bag-of-linguistic-features approach is usually language-dependent and comprises of intensive processing and thus takes time (F. Liu, Perez &

Nowson, 2016). In addition, the approach often requires vast amounts of data to learn, e.g., around 200 posts from a Facebook user (Schwartz et al., 2013), or 100,000 words, in predicting a user's personality (Yarkoni, 2010); consequently, they might not entirely applicable in the real usage scenario of social media, in particular, in Twitter, where every tweet has a cap of 280 characters. Researchers have therefore moved towards deep learning methods.

Over the last few years, the interest in using deep learning for user profiling has grown. For example, it has been used in the business sector to build up a customer demographic profile for each type of user (Smith, 1999). Marketers have attempted to analyse the consumer's buying pattern and its relation with geographical, demographic, and psychological characteristics. Neural network learning approaches, which provide a robust method to compute such behaviour patterns on a nonlinear, parallel task (Mitchell, 1997), are able to uncover that valuable information. The approach has been successfully applied to problems entailing real-world sensor data such as face recognition (Lawrence, Giles, Tsoi & Back, 1997; Parkhi, Vedaldi & Zisserman, 2015) and handwritten character classification (Cireşan, Meier, Gambardella & Schmidhuber, 2011). Furthermore, in Natural Language Processing (NLP) applications, neural network learning has been shown to be effective in text classification (Conneau, Schwenk, Barrault & LeCun, 2016; Kim, 2014). In regard to personality detection from self-authored text, a variant of neural networks known as Convolutional Neural Network (CNN) (LeCun & Bengio, 1998) has demonstrated promising performance (Majumder, Poria, Gelbukh & Cambria, 2017; Kalghatgi, Ramannavar & Sidnal, 2015). Although the research has been devoted to entailing document-level features, rather less attention has been paid to infer trait at the sentence-level. Taken together, the results thus far reveal the need for further empirical study.

1.1 Research Motivation

Personality trait assessment can be a valuable resource and has been used in a wide range of studies. This is exemplified in the work undertaken by Chamorro-Premuzic and Furnham (2003) in examining students' academic performance, and studies in the workplace to investigate the correlation between an applicant's aptitude and achievement (Goldberg, 1993; Judge, Thoresen, Bono & Patton, 2001). In some cases, personality dimensions are related to the types of products or services that are offered, such as a game to match a player's personality (Yang, Lin, Huang & Tsai, 2017). However, similar to the eliciting of public perception, the conventional personality assessment requires adequate time and resources which rely on self-report and empirical investigation through questionnaires (John & Srivastava, 1999). In spite of the fact that it has a profound theoretical significance, such an approach can be tedious. Social media, on the other hand, unprecedentedly provide the digital footprint of human behaviours and social interactions that were not previously possible in both scale and extent. For this reason alone, it is imperative to harness the potential of social media as a tool or method with the intention of understanding user behaviours within the platform.

Personality has been found to influence an individual's choice of words. As highlighted by Stemmler and Wacker (2010), persons with same personality traits tend to express similar sentiments. While this observation has already drawn attention to investigating sentiment analysis based on personality traits, such as the work of Lin, Mao and Zeng (2017), there is a general lack of research in exploring the role of opinion polarity in trait inference. Besides, in practice the existing models tend to ignore the sentiment information in sentences (Carducci et al., 2018).

Driven by above-mentioned motives, this work presents a smart system called SENTIPEDE, stands for *Sentiment-based Personality Detection*. The term SENTIPEDE is used to refer to the proposed system in the rest of the thesis. This new system employs Neural Network Language Model (NNLM) to predict user personality from a self-authored text incorporating sentiment information conveyed. Moreover, to better understand the existing correlation of personality and public perceptions, we further conduct a case study-based investigation.

1.2 Contributions of the Study

This research makes the following contributions:

- **SENTIPEDE: A smart system for personality detection.** We develop a smart system using a *Python* web framework for extracting user personality traits from short texts. The main tasks of the system include *Twitter data scraping*, *Sentiment analysis*, and *Personality detection*. We use pre-trained word representations named Global Vectors (GloVe) to transform the given texts into an embedding matrix, and later feed them onto a neural network with CNN and a recurrent network called Long Short-Term Memory (LSTM). The system returns prediction scores across the five board personality dimensions. SENTIPEDE can be accessed online at the following link: http://sentipede.dsrg.ac.nz.

- **The case study of Uber.** A case study-based investigation is conducted employing the recommended system. We opted for a ride-sharing company of Uber as the subject of this study. The topic is selected on the basis of a degree of attention received from the online community which provides us with enough variability to be explored. The selected case study, therefore, is expected to provide an insight into the relationship between personality traits and opinion polarity.

- **Performance evaluation.** Several well-known deep learning approaches under the umbrella of NNLM are implemented in this work: CNN, LSTM, and a unified model combining the two models. We compare the performance of each variant under both sentiment classification and personality detection tasks, and determine the best models to predict the personality traits from social media.

1.3 Thesis Structure

In this section, we provide a brief explanation of the chapters included in this research. Beginning with the present introductory chapter that gives background to and key concepts of the study, laying the groundwork for the research, the rest of the thesis is organised as follows:

Chapter 2 *Literature Review*, presents an overview of the prior research that set the stage for the current work in the context of the following topics: (1) *natural language processing*; (2) *neural network language modelling*, covering basic understanding of language models in the realm of neural network learning; (3) *social media mining*, particularly in extracting relevant information from digital traces, e.g., opinion mining and sentiment analysis; and (4) *personality traits*, explained under the broadly-accepted personality dimension of the Five-Factor Model.

Chapter 3 *Methodology*, covers the practical understanding of the system architecture. The substance of the chapter is threefold: *data retrieval*, *Twitter sentiment classification*, and *personality detection*. We discuss various approaches to developing neural network language models. In that account, a set of experiments is conducted utilising some well-known corpora: Stanford Twitter Sentiment (STS) and the myPersonality corpus.

Chapter 4 *System Development,* demonstrates the development of the proposed system (SENTIPEDE). It commences with mapping out the models discussed in Chapter 3, builds a responsive web-based system, delivers the software and, finally, integrates a pipeline to deploy the website to production.

Chapter 5 *Results and Evaluation,* reviews the experimental results obtained on each of the learning models explained in Chapter 4. In addition, a case study—investigating the ride-sharing company of Uber—is set out in align to examine the relationship between personality traits and public perceptions. We perform sentiment analysis and personality detection on Uber's users adopting SENTIPEDE.

Chapter 6 *Discussion*, is dedicated to discussing the findings of the research and the experimental validation of the proposed system. It ends by highlighting some restrictions and limitations.

Chapter 7 *Conclusions and Future Research*, is devoted to the conclusions drawn from this study and suggestions for future work.

Chapter 2

Literature Review

The following chapter presents an overview of the related literature used to support the current study, and is divided into five parts. We first provide the introduction to the basics of natural languages processing in Section 2.1. The second part, Section 2.2, discusses language modelling techniques under the umbrella of neural networks. In Section 2.3, we include a survey of the literature on social media mining and continue to review the computational linguistics techniques for eliciting public opinion expressed online known as sentiment analysis. This is followed by an insight into the personality theory covering the Five-Factor Model of personality traits, which is outlined in Section 2.4. Several studies on computational personality recognition from social media are also discussed. This chapter concludes by identifying the research gap as the starting point for further examination.

2.1 Natural Language Processing

Language is fundamentally a means for humans to express and exchange ideas, thoughts and feelings (Allen, 1995). The term has been widely applied in various semiotics and linguistics-related fields such as mathematics and computer programming languages. Since the emergence of digital technology, for instance, computers were given instructions in the form of the language in which machines are programmed: a set of standard codes that have a well-defined purpose (Steele, 1999, p. 223), also called an artificial language. Despite being based on specific syntactic and semantic rules, this is not something considered as to be a so-called *natural language*.

According to the *Concise Oxford Dictionary of Linguistics* (Matthews, 2007, p. 109), natural language is "a language in the ordinary sense, which is or has been learned and spoken naturally by a community". Following this definition, understanding natural language is not a conscious effort for humans; conversely for computers, the task is not trivial. This common language rather shows complexity and varied structures in actual practice. Coming in many forms including writing and speech, natural language is melded with diverse meanings of words and phrases which co-occur in different context (Palmer, 2010). To such a degree that it can be notoriously ambiguous on various levels. *Natural language processing*, or NLP, thus was constituted from a statistical modelling approach and linguistic theory to automatically derive meaning from human language (Jurafsky & Martin, 2008).

An essential task in any NLP system is text processing, in which a raw text file or a set of digital files, such as corpora harvested from the Internet, are converted into a well-defined sequence of linguistically meaningful units such as characters, words, and sentences (Palmer, 2010). As the task determines how all further processing stages will be best achieved, defining which techniques are to be implied in the earlier-established stages in NLP is therefore crucial. Some very widely used methods in the processing of natural language texts are described detail in the following subsections.

2.1.1 Word Tokenisation

Tokenisation defined as a process of dividing character streams into words, phrases, or meaningful strings. Given a sentence, the task will break it into a list of words. In this context, words and other things, such as numbers or punctuation marks are called *tokens*. The whole process is then referred to as *word tokenisation* (Manning & Schütze, 1999). In the English language which standardly written with white spaces between words, tokenisation infers intuitively the space-separated words (Bonzanini, 2016). This process can be seen in the sample sentence: "This sentence is short, simple and to the point" ⇒ ['This', 'sentence', 'is', 'short', ',', 'simple', 'and', 'to', 'the', 'point'].

2.1.2 Lemmatising and Text Stemming

The task of mapping the morphological variants of a word into the root base form, or *stem*, can be described as stemming (Manning & Schütze, 1999). This mapping process is used to match words which bear the same meaning. For instance, English speakers recognise that the words 'drive', 'driving', and 'driver' are derivatively related. The most common algorithm for stemming is suffix stripping, and one that is empirically effective is the Porter Stemmer (Porter, as cited in Bonzanini, 2016). However, for some irregular verbs, this algorithm would not work, thus lemmatisation is involved. An approximation to stemming, the goal of lemmatisation is to reduce inflectional forms to common conceptual form. In 'go', 'went', and 'gone', for example, while the stemming algorithm failed to infer 'went' as 'go', lemmatising successfully recognised the lemma. Such an approach can be implemented with the use of WordNet[1], a lexical resource for English language that groups words into sets of synonyms called *synsets*.

[1]https://wordnet.princeton.edu

2.2 Neural Network Language Model

As previously mentioned, NLP embodies the statistical modelling approach. This can be found in most language models today. According to Kirchhoff (2012), a language model computes the probability of a sentence. Given a vocabulary Σ, commonly in a list of unique words encountered in the training data set, and a sequence $W = w_1 w_2 ... w_t \in \Sigma$, the language model applied to estimate joint probability of words in sentence based on the chain rule which is represented as:

$$P(W) = \prod_{i=1}^{t} P(w_i \mid w_{i-1} w_{i-2} ... w_2 w_1) \tag{2.1}$$

The most conventional statistical language modelling approach is the sequence of n words technique known as n-gram model, which can be referred as unigrams ($n = 1$), bigrams ($n = 2$), trigrams ($n = 3$), 4-grams, and so on, depending on the count of words. The n-gram model has been applied in various NLP applications such as for text retrieval (Mayfield & McNamee, 2003) and sentence classification (Tripathy, Agrawal & Rath, 2016); however, one drawback of this approach is data sparseness, particularly in dealing with a large vocabulary. As performance on a statistical language processing task relies upon the information accuracy found in a corpus, this phenomenon occurs due to not enough data being observed (Allison, Guthrie & Guthrie, 2006).

The Neural Network Language Model (NNLM), nevertheless, has the ability to process sparse data. Designed to model the capability of the human brain to performs certain computations, a neural network, often termed as Artificial Neural Network (ANN), is a massive interconnection of *neurons* which act as information-processing units (Haykin, 2009). Resembling the neurons, ANN emulates their functionality to adapt based on the input, output and feedback.

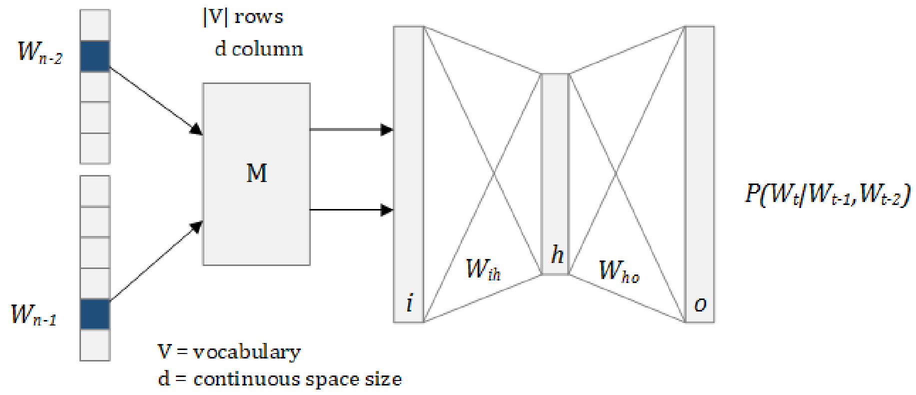


Figure 2.1: Neural network language model. From "*Language Modeling*", by K. Kirchhoff, 2012, in D. Bikel & I. Zitouni (Eds.), *Multilingual natural language processing applications* (pp. 169-198). Upper Saddle River, NJ: IBM Press.

In the context of machine learning, a simple model of a neuron, also referred to as a computation node or *perceptron*, has a fundamental role in the learning process (Mitchell, 1997). In a single-layer perceptron network, an input layer is projected onto an output layer of nodes. This type of network is also designated as a *feed-forward* or *acyclic* type (Haykin, 2009). Meanwhile, a multilayer feed-forward network, along with an input layer and a single layer of output nodes, distinguishes itself by the presence of at least one *hidden layer*. Drawing from this, according to Kirchhoff (2012), NNLM resembles the latter. Also termed a neural probabilistic language model, a graphical representation of Neural Network Language Model architecture is shown in Figure 2.1.

From the figure, in the NNLM architecture, each word in $W \in \mathbb{R}^{d \times |V|}$ projects to a row of vocabulary V in the matrix M. The output is the probability of the next word. In this present work, two common algorithms in neural networks were included: the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN). We used the two models and their variants for personality prediction from short texts. Key to the methods is the use of a dense distributed representation for each word as explained fully in the following subsections.

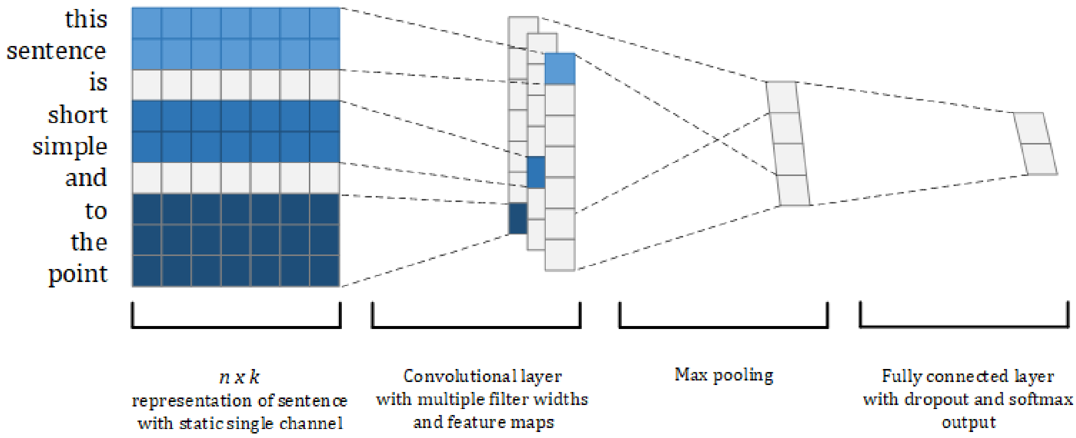


Figure 2.2: Model architecture for an example sentence. Adapted from "Convolutional Neural Networks for Sentence Classification", by Y. Kim, 2014, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746-1751). doi: 10.3115/v1/D14-1181.

2.2.1 Convolutional Neural Networks

Intentionally designed for computer vision (LeCun & Bengio, 1998; Lawrence et al., 1997), Convolutional Neural Network (CNN) has been refined and become prevalent among natural language processing researchers (Collobert et al., 2011; Shi, 2017). The model performs very well on various of NLP tasks including speech recognition (Abdel-Hamid et al., 2014), sentence classification (Kim, 2014), and sentiment analysis (dos Santos & Gatti, 2014).

A typical CNN is a multilayer *feed-forward network* (FFN) which consists of a set of layers with convolving filters that are applied to local features (Kim, 2014), Figure 2.2 gives an illustration. In regard to applying CNN in a language model, the network takes as input the sequence of words in a sentence. Each column $x_i \in \mathbb{R}^k$ corresponds to the i-th word in the sentence of length n. Its inputs are passed through a convolution layer which breaks them into small windows of h words and apply the same transformation to each of these windows involving a filter $w \in \mathbb{R}^{hk}$ to produce a new feature. Commonly, this layer is followed by a

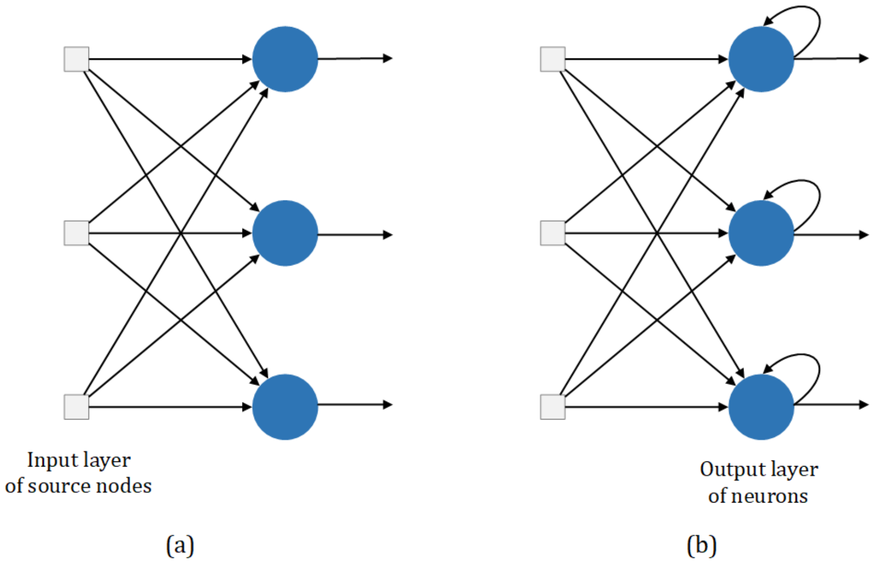


Figure 2.3: The difference in the information flow between (a) a feed-forward or acyclic network with a single layer of neurons, and (b) a recurrent network with a self-feedback loop. From *Neural Networks: A Comprehensive Foundation* (2nd ed., pp. 21-23), by S. Haykin, 1999, Upper Saddle River, NJ: Prentice Hall.

pooling layer which captures the most important feature for each feature map. These features form the penultimate layer and are fed into a fully connected layer (Collobert et al., as cited in Kim, 2014).

2.2.2 Recurrent Neural Networks

Language implies more complexity than understanding individual words. In deriving meaning, humans do not discharge all the previous words in a sentence and start processing from the beginning of a new sentence; instead the previous words are kept to allow information to persist. This is important as the sequence of words holds crucial information to allow the prediction of the up-coming words (Shi, 2017). While FFN could not do this, a Recurrent Neural Network (RNN) addresses this issue by adding the immediate past information to the present tasks. An RNN differentiates itself from FFN in that it has one or more *feedback loops* (Haykin, 2009), as illustrated in Figure 2.3.

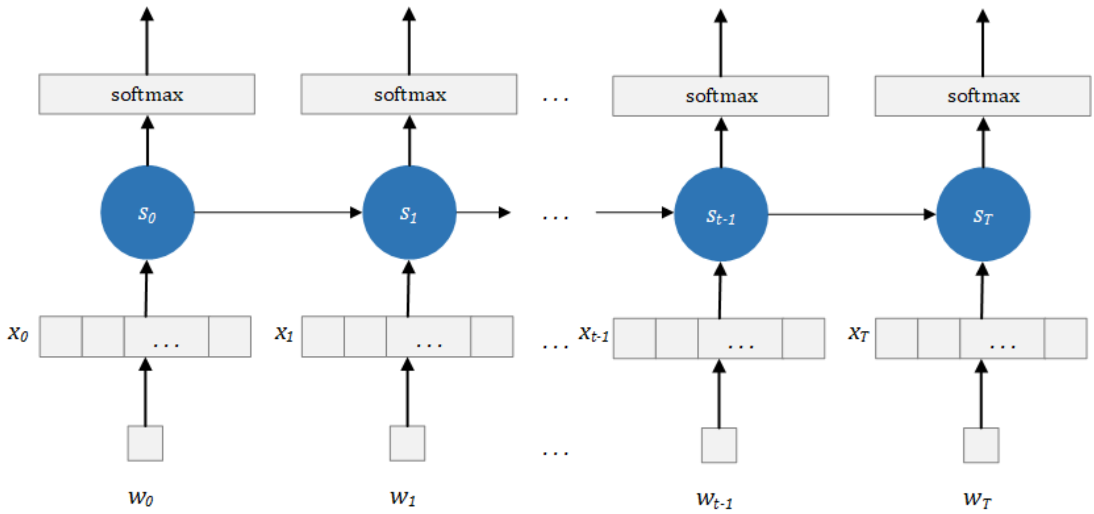


Figure 2.4: Recurrent neural network language model. Adapted from "A Study on Neural Network Language Modeling", by D. Shi, 2017, *Computing Research Repository (CoRR)*, abs/1708.07252.

The early implementation of RNN in language models was proposed by Bengio et al., and later was refined by Mikolov et al. (as cited in Shi, 2017). The representation of words in RNN is similar to that in FFN; however, unlike FFN in which a weight matrix is assigned to its inputs to produce the output, RNN applies weights to both the current and the previous input. As shown in Figure 2.4, the inputs of RNN is the feature vector of a direct previous word. At every step, RNN involves the previous internal state. As the outputs are non-normalised probabilities, they need to be regularised such as using a *softmax layer*.

2.2.2.1 Long Short-Term Memory

In processing sequential data, FFN only considers the current input and has no notion of order in time. Although RNN is capable of coping with such a short-coming, when dealing with long sequences, the task can be extremely difficult due to the vanishing gradient problem (Shi, 2017). Introduced by Hochreiter

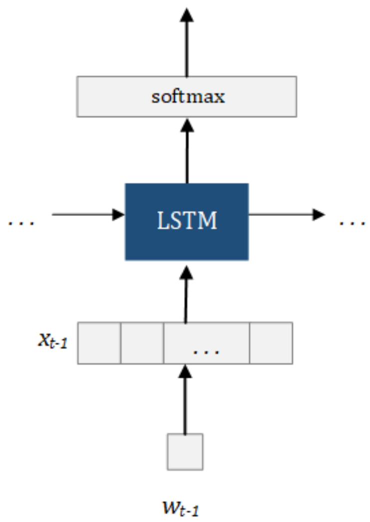
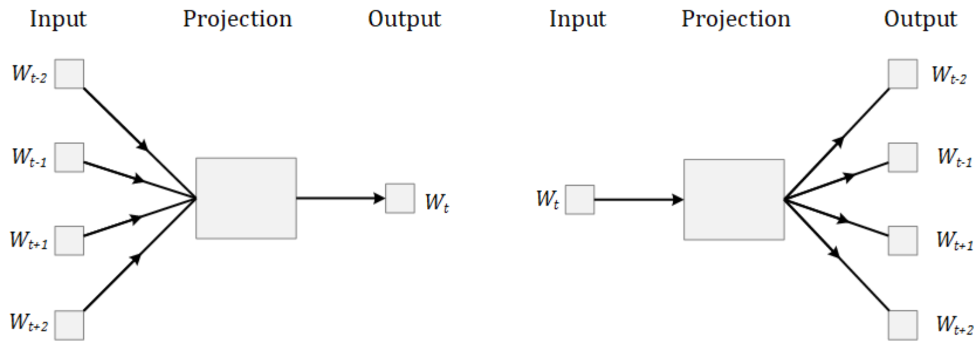


Figure 2.5: Long short-term memory architecture. Adapted from "A Study on Neural Network Language Modeling", by D. Shi, 2017, *Computing Research Repository (CoRR)*, abs/1708.07252.

and Schmidhuber (1997), the Long Short-Term Memory (LSTM), is explicitly designed to avoid the long-term dependencies issue. As a variant of recurrent networks which have the form of a chain of repeating modules of a neural network, LSTM is equipped with a cell state that runs straight down the entire chain (see Figure 2.5). The distinction is that LSTM has a special gating mechanism which regulates access to memory cells (Kalchbrenner, Danihelka & Graves, 2015).

2.2.3 Word Representations

A word can be described as the smallest single isolated element from a sentence that carries meaning (Smrž & You, as cited in Bikel & Zitouni, 2012). In most state-of-the-art NNLM, word embeddings play an important role (Bengio et al., as cited in Kim, 2014). They work tremendously well at capturing words with similar meanings that have a similar representation. Often referred as *distributed representations of words*, word embeddings are the numerical representation of



(a) (b)

Figure 2.6: Architecture of word2vec models of (a) Continuous Bag-of-Words and (b) the Skip-Gram. Adapted from "Two/Too Simple Adaptations of Word2Vec for Syntax Problems", by W. Ling, C. Dyer, A. Black, and I. Trancoso, 2015, in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1299-1304). doi: 10.3115/v1/N15-1142.

words which are mapped in a dense, real-valued vector based on surrounding context (Mandelbaum & Shalev, 2016). Two common algorithms thereof are word2vec and Global Vectors (GloVe) which are explained more fully in the subsections below.

2.2.3.1 Word2Vec

Word2vec[2] is an embedding algorithm which was originally created by Mikolov, Sutskever, Chen, Corrado and Dean (2013). Individual words are positioned in the vector space such that words with common linguistic context are located in close proximity. Given a word w, word2vec can compute a numeric vector and produce a list of words that are similar to w. In this context, this gives cosine similarity values. The word2vec approach to learning the word embeddings comes in two models as illustrated in Figure 2.6. They are:

i. **Continuous Bag-of-Words Model**. In Continuous Bag-of-Words (CBOW)

[2] https://code.google.com/archive/p/word2vec/

representation model (Mikolov et al., 2013), each word is considered a unique token with no relationship to other words. The model discards order information, and works by either summing or averaging the embedding vectors of the corresponding features:

$$CBOW(f_1, ..., f_k) = \frac{1}{k}\sum_{i=1}^{k} v(f_i) \tag{2.2}$$

ii. **The Skip-Gram Model**. In the opposite of the CBOW model, the training objective of the skip-gram model is to predict surrounding context words (Mikolov et al., 2013). Given a sequence of words $w_1, w_2, w_3, ..., w_T$, the skip-gram model maximises the average log probability represented as:

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j} \mid w_t) \tag{2.3}$$

2.2.3.2 Global Vectors

Global Vectors for word representation, or simply GloVe,[3] can be described as an unsupervised learning vector representation for words (Pennington, Socher & Manning, 2014). In contrast to word2vec, which typifies prediction-based models, GloVe is a count-based model. The algorithm works on aggregated global word co-occurrence statistics from a corpus. Considered two words i and j, the relationship of these words can be extracted by computing the ratio of their co-occurrence probabilities with various probe words k, represented as:

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}} \tag{2.4}$$

where, X_{ij} is the number of times word j occurs in the context of word i.

[3]https://nlp.stanford.edu/projects/glove/

The presence of pre-trained word embeddings models like GloVe has been shown to improve performance in many NLP tasks such as query expansion (Diaz, Mitra & Craswell, 2016) in information retrieval, and text classification (Wang et al., 2016) in data mining. The word embeddings technique has also been demonstrated for social media mining; Hayashi and Fujita (2018) utilised GloVe pre-trained word vectors harvested from Twitter to extract features from sentences in sentiment classification. Such tasks and other thematic areas of research on social media will be discussed in the next section.

2.3 Social Media Mining

The phenomenon represented by the buzzwords *social media* seems to have influenced human interaction and communication on an individual and a community level (van Dijck, 2013). Roughly defined as “a convergence between personal communication and public media” (Meikle & Young, as cited in Fuchs, 2013), social media have enabled users to collaborate, exchange content, and disseminate information on social spaces in everyday life.

Promoted by the advances in the Internet of Things (IoT) and mobile technologies, the number of social media users has continued to proliferate over the past decade. Kemp (2018) reported over three billion users were online worldwide in 2018, resulting in nearly 1.5 million new items of data being created per day; and this trend seems more likely to continue with exponentially growth in the future.

The promise of social media has drawn academics’ attention to observing data streamed online. Social media employ a strong appeal to what has been referred to as *big data* which, as opposed to the traditional structured data, is characterised as large, noisy, and dynamic (Barbier & Liu, 2011, p. 332).

Handling such data can be extremely challenging. Using conventional techniques for classic data mining often less effective. This being the case, the importance of interdisciplinary studies became apparent. A new field has emerged—*social media mining*—referring to a process that entails the representation, analysis, and extraction of actionable patterns from social-related data (Zafarani, Abbasi & Liu, 2014).

Many studies show that social media mining techniques have been employed to examine either the social phenomena or the adoption of social media themselves. This can be seen in marketing (Syrdal & Briggs, 2018), digital branding (Al-Sheikh & Hasanat, 2018), or political participation and democratic transition (Kruse, Norris & Flinchum, 2018). In the following sections, we will discuss another fine example of this: a process to determine reputation from the online community, this process being known as *sentiment analysis*.

2.3.1 Opinion Mining and Sentiment Analysis

As defined by Pang and Lee (2008), sentiment analysis is the use of computational linguistics to glean public opinions on a particular product or topic from a vast volume of content generated by users. Also referred to as *opinion mining*, the basis of sentiment analysis is to determine whether a given opinion is positive, neutral (nonpartisan), or negative (B. Liu, 2011).

According to Collomb, Brunie and Costea (2013), in terms of the structure of the text, sentiment analysis can occur at different levels of granularity: *document level* and *sentence level*. At the document level, the classification aims to determine the polarity based on the overall sentiment from a whole review. By contrast, at the sentence level, the classification involves calculating sentiment polarity for each sentence of a review. This process sometimes also entails subjectivity

Table 2.1: Studies of Twitter Sentiment Analysis

(a) Machine learning approaches

Study	Algorithms	Features	Data set
Go et al.	NBC, maximum entropy, SVM.	n-gram, part-of-speech.	STS
Mohammad et al.	SVM	n-gram, part-of-speech, caps, lexicons, punctuation, negation, tweet-based.	SemEval-2013
Bakliwal et al.	NBC, SVM.	n-gram, polarity, emoticons, hashtags, URLs, targets.	STS

(b) Lexicon-based approaches

Study	Algorithms	Features	Data set
Thelwall et al.	SentiStrength	Polarity, emoticons, negations, emphatic lengthening, boosting words.	SS-Tweet
Ortega et al.	Clustering-based WSD, lexicon-based classifier.	WordNet, SentiWordNet.	SemEval-2013
Saif et al.	SentiCircles	SentiWordNet, MPQA, Thelwall-Lexicon.	Obama-McCain Debate, Health Care Reform, STS-Gold.

Note. From "Like it or not: A survey of Twitter sentiment analysis methods" by A. Giachanou and F. Crestani, 2016, *ACM Computing Surveys*, *49*(2), pp. 16-18.

classification of a sentence into objective or subjective.

2.3.1.1 Sentiment Classification Methods

Most existing studies in opinion mining, especially on Twitter, broadly apply two main approaches: *machine-learning* (supervised) and *lexicon-based* (unsupervised) methods. Practically, both approaches have a dependency on underlying opinion words (Hutto & Gilbert, 2015). In the supervised classification, the algorithm relies on a set of examples annotated with true class. This corpus can later be trained to classify correctly all possible inputs. Naive Bayes Classifier (NBC) and Support Vector Machine (SVM) are some commonly used supervised classifiers on Twitter Sentiment Analysis (TSA) (Giachanou & Crestani, 2016; Neethu & Rajasree, 2013). On the other hand, sentiment classification can also be accomplished without the need for training data. Instead, determining the overall opinion score of a given text utilises lists of annotated words. The whole procedure is known as the lexicon-based approach (Kolchyna, Souza, Treleaven & Aste, 2015).

The lexicon-based classifications have been extensively applied to conventional texts. However, compared to supervised learning methods, they have been less fully explored in TSA. This due to the uniqueness of Twitter in which a character limit is imposed on tweets, and so tweets are rife with all sorts of abbreviations and colloquial expressions. Nevertheless, a great many efforts have been made for TSA employing lexicon-based methods, as shown in Table 2.1

2.4 Personality Traits

When it comes to social media, users also share more insight on themselves which, to a certain degree, reflect their personalities. Personality refers to the combination of thought, feeling, motivation and behaviour that characterises a person (Burton, Westen & Kowalski, 2014). According to traits theory, personality occurs through the interplay of psychological processes which are activated in particular situations. From these arise relatively stable characteristics shaping individuals, which are called *traits*. Developed by Allport in the mid-1930s (as cited in Burton et al., 2014, p. 434), the traits approach originated in

Table 2.2: The five-factor model and its facets

Trait	Description	Facets
Openness	Refers to as being emotional, curious, imaginative, and creative	*fantasy, aesthetics, feelings, actions, ideas, values.*
Conscientiousness	Describes as being organised, dependable and motivated.	*competence, order, dutifulness, achievement striving, self-discipline, deliberation.*
Extroversion	A person with the trait has a tendency to be sociable, active, and willing to take risks.	*warmth, gregariousness, assertiveness, activity, excitement seeking, positive emotion.*
Agreeableness	Indicates individuals who are cooperative, helpful, and trusting.	*trust, straightforwardness, altruism, compliance, modesty, tenderness.*
Neuroticism	Defines a continuum from emotional stability to instability.	*anxiety, angry hostility, depression, self-consciousness, impulsivity, vulnerability.*

Note: From "Psychology" by L. Burton, D. Westen, & R. Kowalski. 2014, Hoboken, NJ: Wiley.

psychology, which relies on self-report and empirical investigation through questionnaires. This approach measures individuals' characters in accordance with their personality-related words usage in the English language. However, there is a disagreement over the number of dimensions describing the core of individuals' personalities.

Early research on personality traits conducted by Eysenck in 1947 (as cited in Burton et al., 2014) identified three major psychological types called *supertraits*, namely Extroversion, Neuroticism, and Psychoticism. Later, in 1965, Cattell clustered the personality dimensions into sixteen personality traits (16PF) including emotionally stable, intelligent, cheerful, imaginative, and sensitive. Based upon these prior models, another model has emerged as the most widely accepted measurement of personality today. Developed by Norman in 1963 (as cited in Celli, 2011), the model consists of five major domains as explained in the section below.

2.4.1 The Five-Factor Model

The Five-Factor Model (FFM) of personality dimensions is derived from five high-order traits comprises of Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism, which are commonly abbreviated as OCEAN. Also known as the Big Five, these traits represent an individual's personality over five dimensions which forms by far the most robust determinant of user personality from lexical cues (de Raad, 2000).

The classical approach for assessing personality is to observe individuals behaviour over time within different situations. However, this method can be tedious and time consuming. Psychologists thus implement a pencil-and-paper method that requires participants to answer self-report questionnaires or describe themselves or a person's personality (Burton et al., 2014). Designed to measure the FFM dimensions of individuals, the Big Five Inventory (BFI) is one such questionnaire. This multidimensional 44-item of personality inventory consists of short phrases with a relatively accessible vocabulary (John & Srivastava, 1999). By virtue of the approach, a person has a score for the five personality factors, where each of them represents a composite of more specific traits called *facets* as shown in Table 2.2.

2.4.2 Measuring the Big Five on Social Media

There has been extensive research conducted in an attempt to assess user personality from digital traces, particularly using the FFM. With Twitter and Facebook dominating as the two main platforms, most researchers have explored syntactic and lexical features from social media content as mentioned in Table 2.3.

In the study carried out by Celli (2011), twelve cross-linguistic features were extracted from Twitter based on the list of linguistic features developed

Table 2.3: Studies of personality prediction from social media

Sources	Study	Methods	Evaluation	Result
Twitter	Golbeck et al. (2011)	LIWC, Twitter usage, psycho-linguistic features, sentiment.	MAE	0.1192
	Sumner et al. (2012)	LIWC, Twitter usage.	Accuracy	0.919
	Celli et al. (2013)	Cross-linguistic features.	Co-ocurrence	0.6651
	Lima and de Castro (2014)	Word embeddings, Twitter meta attributes.	Accuracy	0.83
Facebook	Schwartz et al. (2013)	LIWC, open vocabulary, extracted topics.	R	0.42
	F. Liu et al. (2016)	Latent topics from n-grams, word representations.	RMSE	0.479
	Farnadi et al. (2016)	LIWC, social network features.	Precision	0.54
Twitter and Facebook	Carducci et al. (2018)	Word embeddings, SVM.	MSE	0.537

Note. MAE=Mean absolute error, MSE=Mean squared error, RMSE=Root mean squared error, R=Correlation coefficient

by Mairesse, Walker, Mehl and Moore (2007). The author evaluated the co-occurrence of Twitter features with most frequent personality models and obtained an average of 66.51% accuracy. In a similar case, Schwartz et al. (2013) investigated the correlation of language features with continuous or ordinal dependent variables such as gender and age from Facebook users. A text analysis tool called LIWC (pronounced ‘Luke’) was utilised to calculate the percentage of words along with different linguistic categories, e.g., pronouns, verbs, and adverbs (Pennebaker & King, 1999). The study revealed that the use of language is influenced by the preceding factor variables.

Several other studies involved social media platform features including time and content usage, notably from Twitter and Facebook (Farnadi et al., 2016;

Golbeck et al., 2011; Sumner et al., 2012). The results indicated that the nature of each platform in which messages are usually incorporated with informal language and abbreviations tends to affect the prediction effectiveness. As shown in Table 2.3, a model to infer personality from Facebook developed by Farnadi et al. (2016) suffered from low precision, whereas Golbeck et al. (2011) and Sumner et al. (2012), utilising tweets to predict personality, achieved an overall good performance.

In the context of automated prediction systems, various methods have been proposed to identify personality from user generated content. This can be seen in Carducci et al. (2018). Relying on Twitter content, the authors developed a supervised learning-based system called *TwitPersonality* to assess the Big Five model from cross-platform posts. They trained the Facebook status corpus employing SVM, and used them to classify from Twitter. This system obtained significant results with an average of 0.537 MSE. In alignment with those authors, Lima and de Castro (2014) built a multi-label classifier system called PERSOMA, adopting semi-supervised learning techniques with NBC and SVM, which resulted in an approximately 83% accurate prediction. In their study, Twitter's meta-attributes were entailed, however, and rather than a single tweet, the system works with groups of tweets.

2.5 Research Gap

2.5.1 Personality Traits and Public Perceptions

As explained earlier, language features play an important role in an individual's personality development. There is a wide area of research examining the correlation between language use and personality; however, few have observed the sentiment expressed in a sentence.

In the realm of public opinion, an individual's behaviour is closely related to any of the sentiment polarity carried in a sentence, namely positive, negative, and nonpartisan. The early work on the correlation between personality and sentiment undertaken by Golhamer (1950) has shown that a person's orientation to expressing opinions may be accompanied by their characteristics. Psychological research reveals that psycho-linguistics have strong correlations with individuals' self-disclosure, particularly in influencing their choice of words, suggesting that persons with the same trait tend to express their sentiments by using similar words (Stemmler & Wacker, as cited in Lin et al., 2017). Additionally, Schoen (2007) stated that personality traits merit serious attention in sentiment analysis, particularly towards public policy (Gerber, Huber, Doherty & Dowling, 2011; Gravelle, Reifler & Scotto, 2014).

In support of prior studies, Lin et al. (2017) constructed a sentiment classifier using features grouped by different personality traits, and the results show their effectiveness in refining the performance. The authors claimed to be among the first to explore the role of user personality in social media sentiment analysis. In contrast, in computational personality research thus far, the existing models tend to ignore the sentiment information embedded in texts (Carducci et al., 2018); this reveals the need for further empirical investigation. To fill the gap, this study therefore explore the role of opinion polarity in trait inference.

2.5.2 Trait Inference from Short Texts

Compared to documents, short updates such as tweets contain limited context which does not always observe linguistic rules, in contrast to what is expected in a written language (dos Santos & Gatti, 2014). Consequently, traditional techniques may not provide significant results when required to handle such peculiarities. Another issue is a tendency of the twitterverse to use abbreviated words or phrases, idioms, and informal languages which are embedded with emoticons and folksonomies (e.g., social tags and social bookmarking). This makes the task of personality profiling more challenging.

Notwithstanding this, neural networks learning has been found to perform well when dealing with small amounts of training data and able to carry out NLP tasks, despite large corpora not being available (Güngör, 2010; F. Liu et al., 2016). A variety of approaches entailing neural networks learning have been recently proposed to automatically infer users' personalities. Majumder et al. (2017), for instance, adopted a CNN model on document level features. Employing a collection of stream-of-consciousness essays deployed by Pennebaker and King (1999), the model can achieve up to 62.68% accuracy.

Despite the above mentioned studies, little progress has been made on short messages, particularly tweets. This is exemplified in the study carried by F. Liu et al. (2016). Instead of exploiting CNN, the authors developed a recurrent network-based model with LSTM for personality recognition from short texts. Another example body of work by Kalghatgi et al. (2015) entails social behaviours and grammatical features such as the text length and word usage on a multilayer perceptron network model. The authors concluded by claiming to have successfully predicted personality by employing a group of tweets. However, the study did not include detailed evaluations. There is no clear explanation of the data collection used, of how the authors evaluated the model, or of how validity was achieved. Therefore, the present study extends the empirical approach to address research gaps in previous studies, particularly with a focus on this level of granularity.

2.6 Summary

In this chapter, we reviewed prior research relating to personality recognition from social media. In doing so, some fundamental concepts were presented embodying natural language processing, social media mining, and personality traits theory, in particular the broadly accepted dimensions of personality—the Big Five. We explained language modelling, from statistical methods to a more advanced modelling technique using neural networks. We found that two variants thereof have been exploited recently for computational personality detection: the Convolutional Neural Network (Majumder et al., 2017) and an RNN named Long Short-Term Memory (F. Liu et al., 2016). It has also been revealed that the role of sentiment in trait inference merits serious attention. However, in prior studies, this was not fully explored. Moreover, we identified that most researchers tend to focus on inferring traits at the document-level rather than at the sentence-level, specifically tweets. Therefore, in the following chapters, we attempt to define a framework for predicting a person's personality based on the way sentiment was carried in tweets.

Chapter 3

Methodology

This chapter provides the framework for methods and techniques applied in the present study; it consists of four parts, which are outlined as follows: Section 3.1 highlights the research design. We discuss the methods used to construct the models and go into advanced detail on the tools and libraries required. Using social media as research data is not only promising but also presents significant privacy issues. Thus, the choice of data sets used and the processing techniques implied are the subject-matter of Section 3.2. Short message contains limited context which often disregards linguistics rules. Accordingly, for sentiment classification purposes, it is pivotal to determining the approach and features utilised, and is the topic of Section 3.3. We complete the chapter by presenting a unified model for predicting personality traits from opinionated text streamed online in Section 3.4

3.1 Research Design

The expansion in the use of the Internet has facilitated researchers' examination of online society in which social media have come to be adopted. To the social sciences, the Internet offers technological means to address some previously intractable problems of social science methods (Lee, Fielding & Blank, 2017). The same point applies to personality traits inference. Prior study on computational personality recognition from social media can be traced back to the early 2010s (Celli, 2011; Golbeck et al., 2011; Schwartz et al., 2013). These scholars highlighted the potential of social media as method or research instrument for identifying users' personality traits. The approaches varied according to machine learning algorithms, feature sets employed, and platforms used to glean the research data (Farnadi et al., 2016). The current study thus closely followed the path demarcated by these works. It was designed to infer personality traits from opinionated texts streamed online, in particular tweets. We attempted to develop an automated system employing Neural Network Language Model (NNLM), and conducted a case study-based investigation on the correlation of personality traits and opinion polarity. We begin the description of this work with the set up stage of the research. This includes the blueprint for the proposed system, requirements, and data collection. Further details are provided in the following sections.

3.1.1 System Architecture

This section presents the design framework for our system: *Sentiment-based Personality Detection* or SENTIPEDE. In this system, deep learning-based models with neural networks and a single embedding layer are used to forecast personality traits. Each model is made up of a number of parameters that tune the outcomes. The system has three layers working in sequential mode, as explained below. The full description for the modelling design is illustrated in Figure 3.1.

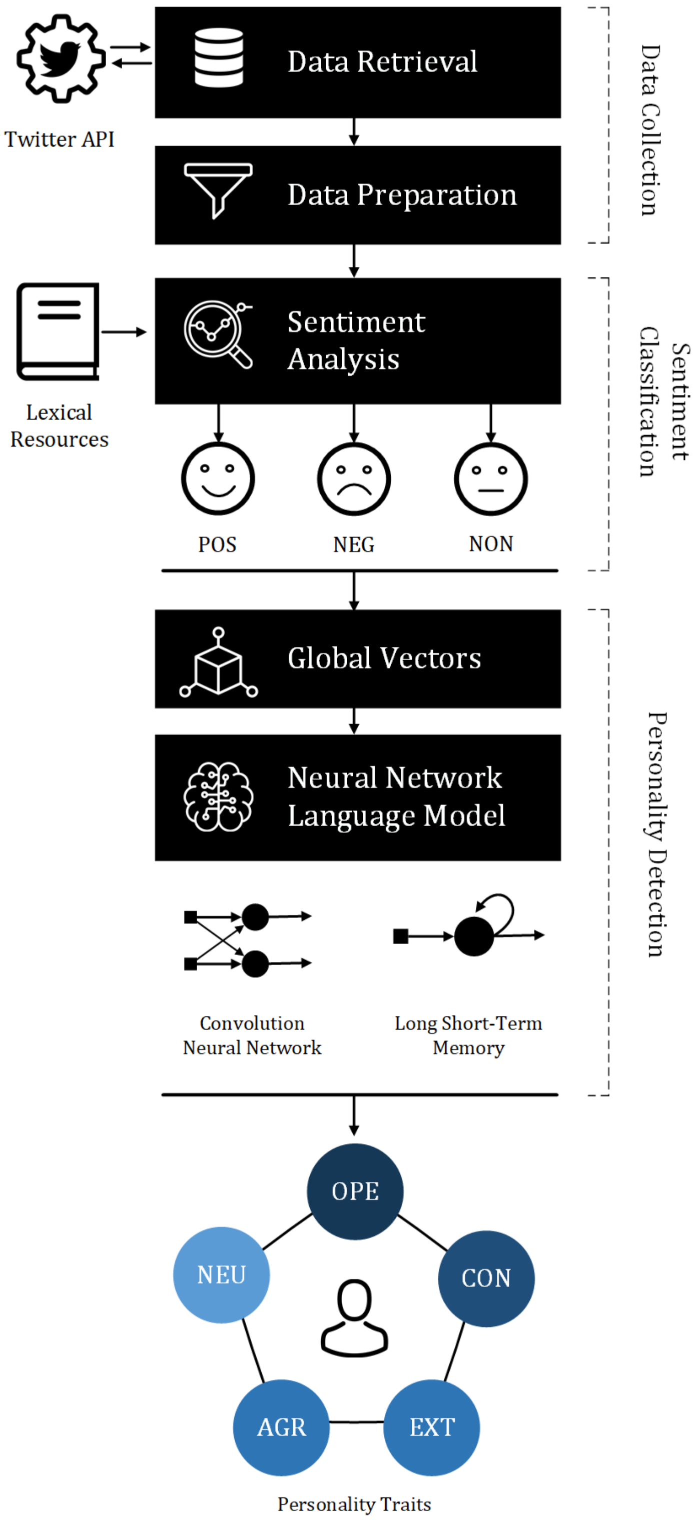


Figure 3.1: Proposed system architecture for Sentiment-based Personality Detection (SENTIPEDE).

Layer 1: Data Collection. In the first layer, we implement Twitter data collection and pre-processing. We use Twitter API to download the tweets, and under the pre-processing phase we remove stop-words and apply text stemming to the original tweets. The steps are demonstrated in Section 3.2.

Layer 2: Twitter Sentiment Classification. Once the collected data is cleaned, it moves to the second layer called *Twitter sentiment classification*. Based on the sentiment analysis, the system determines whether a given texts reflects positive (POS), negative (NEG), or nonpartisan (NON); thus, the output produced by this layer is in the form of three groups of tweets categorised by their polarities. This is further explained in Section 3.3.

Layer 3: Personality Detection. In the third layer, a *predictive model* is implemented. As the processed data has been bundled together in categories, the system then transforms these categories into a word embeddings matrix before feeding them into neural networks and training the networks with several predictors. In developing this layer, we experimented with CNN and LSTM. The system returns the final scores for each personality dimension, i.e., Openness (OPE), Conscientiousness (CON), Extroversion (EXT), Agreeableness (AGR), and Neuroticism (NEU). The processes are described in detail in Section 3.4.

3.1.2 Tools and Libraries

In this work, we utilise tools and libraries that inter-operate with the Python programming language. The tools and libraries used were:

- **NLTK**. The Natural Language Toolkit[1] or, more commonly, NLTK is a suite of libraries and programs for symbolic and statistical natural language

[1]https://www.nltk.org/

processing. The library provides related packages such as stop words and lexicons for English words.

- **Scikit-learn**. Scikit-learn[2] is a machine learning library that features various classification, regression, and clustering algorithms such as SVM.
- **Keras**. This open source library written in Python is used for developing and evaluating deep learning models. Keras[3] supports both CNN and RNN models.
- **Jupyter Notebook**. Jupyter[4] is an open-source web application for creating and sharing documents that contain live code designed to work with Python language. The application can accommodate various tasks including data cleaning and transformation, and machine learning.
- **Anaconda Navigator**. This graphical user interface-based desktop application is a general purposed virtual environment manager. We used Anaconda Navigator[5] for setting up the Python development environment.

3.2 Data Collection

This section explains the scope of the research data. According to Voss, Lvov and Thomson (2017), social media design has implications for the choice of data sets, so it is crucial to identify a suitable source of data. Henceforth, we determine the data sets and collection methods used in this work, the analysis of data and the processing pipeline subsequently employed as described below.

[2]https://scikit-learn.org/
[3]https://keras.io/
[4]https://jupyter.org/
[5]https://anaconda.org/anaconda/anaconda-navigator

3.2.1 Data Selection

The first challenge encountered when working on profile information is to find a relevant, publicly accessible data set, as acquiring such data can be problematic, particularly in terms of privacy. The recent Facebook privacy scandal involving a political consulting and strategic communication firm, Cambridge Analytica,[6] is a clear example. In early 2018, the company had harvested personally identifiable information from 50 million Facebook profiles through a personality quiz application called *thisisyourdigitiallife* (Granville, 2018). However, it was revealed later that Cambridge Analytica exploited the data without authorisation to build a system tailored specifically to deliver personalised political advertisements (Greenfield, 2018). Consequently, this attracted public attention and became a global headline which has led to an ongoing debate surrounding the illicit use of such sensitive data. In order to avoid this type of issue, we thus relied on the community to crowd-source a gold standard data set labelled with the Big Five called *myPersonality*. The collection is part of a project of the same name initiated by Kosinski, Stillwell and Graepel (2013). Harvested from an online personality assessment application that was specifically built for Facebook platform, the myPersonality data has been made publicly available through the project's web site.[7]

Twitter and Facebook shared the same characteristics as they are platforms for users to broadcast ideas and opinions. Thus, the myPersonality corpus met the criteria for data sets used in this study. Additionally, Carducci et al. (2018) trained the same corpus to investigate personality detection from Twitter users. The author applied a transfer learning approach by reusing the trained model to

[6]https://cambridgeanalytica.org/
[7]http://mypersonality.org/wiki/doku.php

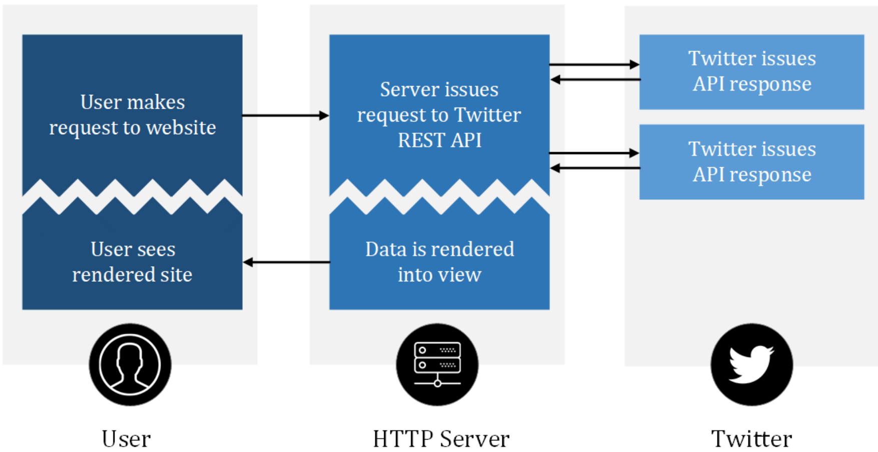


Figure 3.2: Twitter REST API. Reprinted from *Adaptive Crawler for Real-Time Event Monitoring on Social Media (Twitter)*, by A. Tyagi, 2016. doi: 10.13140/RG.2.2.26892.92800.

predict personality traits using tweets as inputs. In correspondence with that work, we based our study on the same sample of the myPersonality data set. Another two sources, the STS corpus and tweets gathered online, were also utilised for training and testing the model respectively. The data retrieval process is explained in detail in the following sections.

3.2.2 Data Retrieval

As indicated previously in Section 2.3, when collecting social media data we are actually dealing with big data and the challenges it entails. Such data was presented in various formats and structures, and sometimes was incompatible as it did not have the format and structure required for this study. The solution for this data collection was found through the use of Application Programming Interface (API).

An API being "a set of procedure definitions and protocols that describe the behaviour of a software component" (Bonzanini, 2016, p. 12). In this study, we employed APIs provided by Twitter[8] for collecting data in real time and historically using Streaming API and Representational State Transfer (REST) API respectively. These APIs enabled us to integrate the application with particular social media functionalities, including accessing the data, possibly behind the authentication layer (see Figure 3.2). However, like any publicly available APIs, Twitter APIs place restrictions and limitations on data access, in terms of historical data collection and/or on the number of data items returned based on the criteria given.

3.2.3 Data Analysis

Data analysis aims to extract information from raw data. In this section, we identify the collected data and discuss insights into data preparation. This includes actions that should be taken in the following stages. The analysis of data used in this study is presented as follows.

3.2.3.1 The Stanford Twitter Sentiment Corpus

The data sets used were collected from the Sentiment140 website,[9] a project of which originated at Stanford University. More commonly known as the Stanford Twitter Sentiment (STS), the data sets mainly contain two corpora which are designed for training and test (STS-Test). The STS is a collection of 1.6 million tweets, while the STS-Test has 498 tweets. Both data sets were automatically labelled with the sentiment polarity of "[0] negative", "[2] neutral" and "[4] positive".

We also utilised the STS-Gold corpus which, unlike the previous data sets (STS and STS-Test), has been human-annotated by Stanford University's research

[8] https://developer.twitter.com/en/docs.html
[9] http://help.sentiment140.com/for-students

Table 3.1: The number of tweets from Stanford Twitter Sentiment data sets

Data set	Positive	Neutral	Negative	Total
STS	800,000	-	800,000	1,600,000
STS-Gold	632	-	1,402	2,034
STS-Test	182	139	177	498

Table 3.2: Statistics about the personality traits of the myPersonality data set

	OPE	CON	EXT	AGR	NEU
Maximum	5	5	5	5	4.75
Minimum	2.25	1.45	1.33	1.65	1.25
Average	4.0786	3.5229	3.2921	3.6003	2.6272
σ	0.5751	0.7402	0.8614	0.6708	0.7768

Notes. OPE=Openness, CON=Conscientiousness, EXT=Extraversion, AGR=Agreeableness, NEU=Neuroticism, σ=Standard deviation.

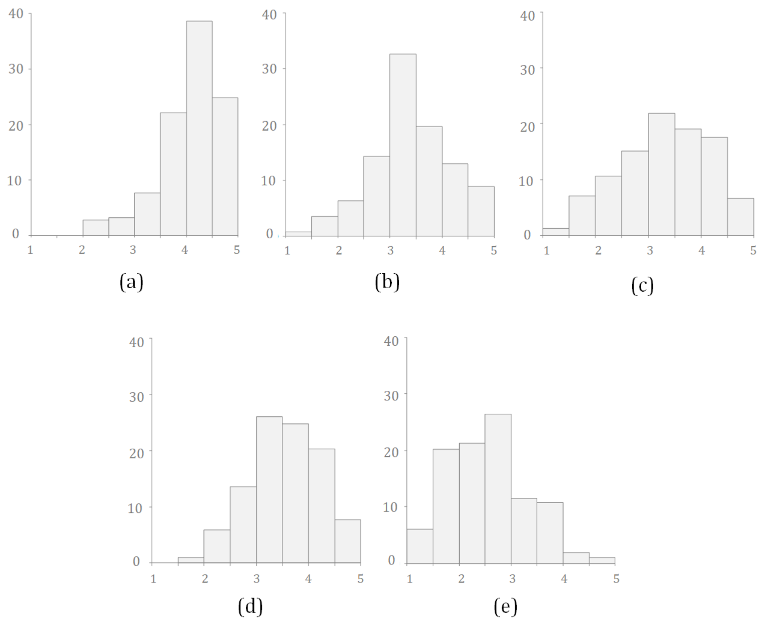


Figure 3.3: A histogram distribution for the five traits of personality: (a) Openness, (b) Conscientiousness, (c) Extraversion, (d) Agreeableness, and (e) Neuroticism.

students. The STS-Gold data set is a collection of 2,034 tweets which are annotated as "[0] negative" and "[4] positive" sentiments. The number of tweets with each sentiment from all three corpora used are shown in Table 3.1.

3.2.3.2 The myPersonality Facebook Status

To predict the traits of the FFM, we used a collection of 9,913 status updates posted by 250 anonymised Facebook users. The myPersonality corpus was tagged with the five personality traits (see Section 2.4) along with social networking features. The copy of this data set was downloaded in February 2018. The statistics and the distributions of the myPersonality corpus are shown in Table 3.2 and Figure 3.3 respectively.

3.2.4 Data Preparation

In this phase, several steps were performed to convert corpora into well-defined training and test data. Data preparation first entailed the *document triage* process of opinionated texts harvested online, followed by *data cleaning* and *text normalisation* for data modelling. The procedure of data preparation is illustrated in Figure 3.4.

3.2.4.1 Document Triage

Document triage is the critical stage in the information seeking process. In this phase, we decide the document's relevance to our information need (Buchanan & Loizides, 2007). This task involved *character encoding identification*, a process to represent the characters into a machine-readable file. Next, *language identification* was employed to determine the writing system of the natural language of the document. This occurs because the computer-based text is merely a sequence of characters represented by digital bits. Finally, *text sectioning* was used to identify and discard undesirable elements such as images, links, and HTML formatting from content within the file. The digital text files, which are encoded using the ASCII character set, require *asciification* encoding or *romanisation* for

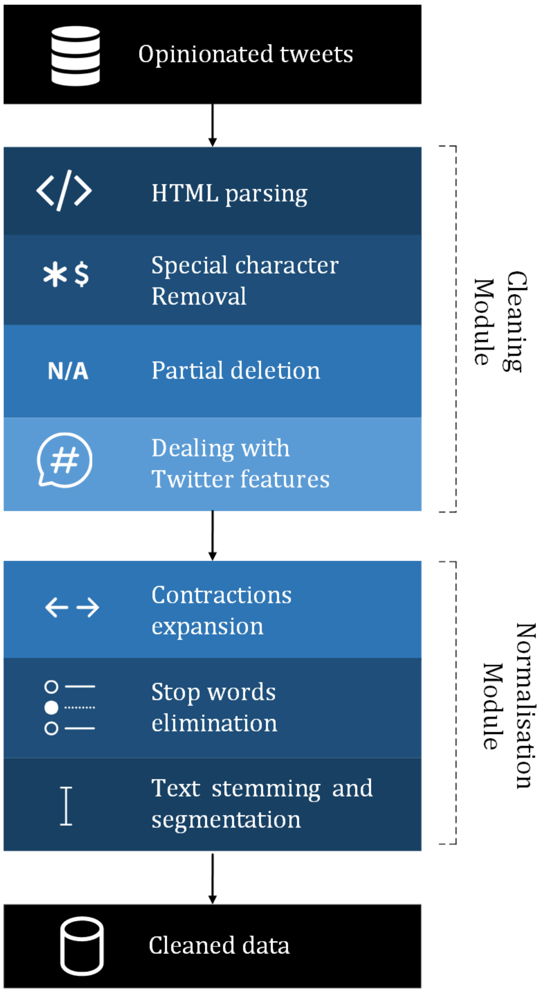


Figure 3.4: Data preparation procedure.

texts commonly using the unicode variable-length standard of UTF-8 encoding. Once the document is set, we move to the cleaning process.

3.2.4.2 Data Cleaning

The data clean-up entailed the following steps:

i. **HTML Parsing**. In some cases, text files formatted in HTML are not fully decoded. HTML tags such as '<p>' and entities like '&amp' are often encountered in a document. To effectively scrape the information from web pages, we applied HTML parsing using a Python library called Beautiful-Soup.[10]

ii. **Special Character Removal and Partial Deletion**. In this step, we performed removal process to all special characters, numeric characters, and empty texts found in the file. We also dropped non-available or missing data in data sets. Additionally, for the myPersonality corpus, we deleted columns containing non-normalised scores for social network features such as ['BROKERAGE'] and ['BETWEENNESS'].

iii. **Dealing with Hashtags, Mentions, and Retweets**. Hashtags, mentions, and retweets like '#Uber', '@uber', and 'RT' respectively may carry certain information which adds value for classification process. Therefore, we treated these popular Twitter features accordingly. For each hashtag in tweets, while we preserved the words, the number signs preceding them were eliminated. However, for mentions and retweets, as in sentiment analysis, the two features can be ignored, and thus we eliminated them.

3.2.4.3 Text Normalisation

Text normalisation entails merging different written forms of a sequence of characters into a canonical normalised form. In detail, the techniques we applied are:

i. **Contractions Expansion**. In this step, we first converted text into lower case. Next, contractions expansion was performed. This process expanded the commonly used English contractions in text, for example, "you're" ⇒ "you are", and "won't" ⇒ "will not".

[10]https://pypi.org/project/beautifulsoup4/

ii. **Stop Words Elimination**. The most frequently used words in text documents are grammatically categorised as articles, prepositions, and pronouns. Some of them, such as ’the’, ’in’, ’has’, and ’he’, are considered to be less informative. These words thus are treated as *stop words*. We filtered them out and did not measure them as keywords. Stop words were removed by utilising the words list for English language provided by a module corpus from NLTK.

iii. **Text Stemming and Segmentation**. We appied stemming using a stemmer[11] provided by NLTK. Subsequently, a Twitter-aware tokeniser[12] is used. This was designed to adapt the tweets’ characteristics while performing tokenisation.

3.3 Twitter Sentiment Classification

Once the document was cleaned of noise and irrelevant information, and the normalisation had been performed, the next step was sentiment classification. In order to assign the sentiment polarity to opinionated texts, we built the sentiment classification of Twitter data by applying lexicon-based approaches. In this stage of the research, three data sets were used for the model evaluation, i.e., STS, STS-Test, and STS-Gold. The pipeline for the lexicon-based Twitter sentiment classification is shown in Figure 3.5.

The Twitter sentiment classification task aimed to automatically estimate the sentiment in a given text as positive (POS), negative (NEG), and nonpartisan (NON). As shown on Figure 3.5, a lexicon-based model invokes a vocabulary of words specifically aligned towards sentiment analysis.

[11]http://www.nltk.org/howto/stem.html
[12]https://www.nltk.org/api/nltk.tokenize.html

Figure 3.5: Pipeline for the lexicon-based Twitter sentiment classification task.

3.3.1 Lexical Resources

In this stage, three lexicons for the English language were used, namely SentiWordNet, AFINN, and VADER. These lexicons are extensively utilised in NLP and semantic analysis, as explained below.

3.3.1.1 The SentiWordNet Lexicon

The SentiWordNet[13] lexicon is based on WordNet[14] synsets, a popular synonym sets used for sentiment analysis and opinion mining. Created by Esuli and Sebastiani (2006), the lexicon assigns three sentiment scores for each WordNet synset: positive and negative polarity scores, and an objectivity score. This lexical resource will return words along with their grammatical category. Some of the most common tags are `NN` for common nouns, `JJ` for adjectives, and `VB` which indicates verbs. To call the lexicon, the line `from nltk.corpus import sentiwordnet` was used.

3.3.1.2 AFINN Lexicon

AFINN is a tab-separated list of English words named after its creator, Finn Årup Nielsen (Nielsen, 2011). The lexical resource contains 2,477 words and phrases which are manually rated on a scale from "[-5] very negative" to "[+5] very positive". The line `from afinn import Afinn` was used to import the library to our model.

[13] https://sentiwordnet.isti.cnr.it/
[14] https://wordnet.princeton.edu/

3.3.1.3 VADER Lexicon

This lexicon is specifically tuned to analyse sentiments in social media. Valence Aware Dictionary and Sentiment Reasoner (VADER) is a rule-based framework developed by Hutto and Gilbert (2015). It contains necessary sentiment scores associated with words, emoticons and slang, with a total of over 9,000 lexical features. Each feature is rated on valence scores of an integer between "[-4] Extremely Negative" and "[4] Extremely Positive", with an allowance for "[0] Neutral (or Neither, N/A)". We used the following code to import the lexical resource: `from nltk.sentiment.vader import SentimentIntensityAnalyzer`

3.4 Personality Detection

The previous section addressed the classification model processing. We trained and tested the models using STS corpus. In this section, we present a predictive model for personality detection. We used the myPersonality corpus that has been automatically annotated using the Twitter sentiment classification built previously in Section 3.3. A unified framework used for predicting personality from short texts incorporating word embeddings is explained as shown in Figure 3.1. The goal was to exploit the more contextual information of short texts as expanded by employing external knowledge from pre-trained word vectors to improve classification performance.

3.4.1 Word Embeddings

In this step, we used the word embeddings technique to transform sparse vector representations of words into a dense, continuous vector space. This process enabled the identification of similarities between words and phrases based on their context. In this study, pre-trained word vectors of tweets provided by GloVe were used. GloVe from Twitter[15] contains two billion tweets with 27 billion tokens and over 1.2 million vocabulary items. It is made available with a vector space ranging from 25 to 200 dimensions.

3.4.2 Training Networks

A CNN is typically a feed-forward neural network, a nonlinear function in which the information flows in the forward direction. Generally, CNN consists of convolution and relevance weight, and pooling layers followed by fully connected layers (Kim, 2014). In this study, we combined it with an LSTM layer. The aim was to take advantage of LSTM in maintaining state by adding the past information to the present state. LSTM has the capability of learning the relationships between elements in an input sequence to overcome the vanishing gradient problem which often occurs when the network is deep enough so that, at some point, the information for learning vanishes.

3.4.3 Model Evaluation

This section describes some of the model evaluation techniques used to calculate the quality of the model predictions. We used performance metrics to measure our models as explained below.

- **Precision (Pre.):** This measures the exactness of the classifier result. Precision can be described as the number of true positives over the total number of positively classified example, represented as:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

[15]http://nlp.stanford.edu/data/glove.twitter.27B.zip

- **Recall (Rec.):** Described as the ratio of the number of positively labelled examples to total examples which are truly positive, recall measures the completeness of the classifier result, formulated as:

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

- **F-Measure (F1):** Also related to the F-score, it is the harmonic mean of precision and recall. F-measure is computed based on the following equation:

$$F - Measure = 2 \times \frac{Prec \times Rec}{Prec + Rec} \tag{3.3}$$

- **Accuracy (Acc.):** Accuracy refers to the overall proportion of correctly classified examples to total number of examples. It is the most common measure of a classification process. The formula for computing accuracy on model predictions is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

3.5 Summary

In the initial sections of this chapter, we introduced the architecture of the proposed system named Sentiment-based Personality Detection (SENTIPEDE). The methods employed, and the packages required in this study were also explained in details. In general, the system has three working layers. The first layer is called *Data Collection*. In this layer, we started with the *retrieval process* and the *analysis of data*. This is followed by the *data preparation* pipeline entailing cleaning techniques such as HTML parsing and special characters removal and, lastly, text normalisation including contractions expansion, stop words elimination, and text stemming and segmentation. Moving to the second layer, *Twitter Sentiment Classification*, the sentiment classification method for short texts was explained. We presented an unified model, combining a type of feed-forward network, Convolutional Neural Network (CNN), and a recurrent network-based model, Long Short-Term Memory (LSTM). Trained employing three lexical resources, a GloVe pre-trained word embedding technique was included in order to expand the knowledge learned from short texts. The whole procedure was described in the *Personality Detection* layer. The evaluation methods taken to calculate the quality of our models were described in the last section. The implementation of this comprehensive strategy is further discussed in the following chapters.

Chapter 4

System Development

The previous chapter dealt with the research design and model construction. In this chapter, we explain thoroughly the development of the proposed smart system for forecasting personality traits. The chapter is divided into four parts. The first part, Section 4.1, focuses on defining the system and the development model used. The rest of the chapter is devoted to the development process which is outlined as follows: Section 4.2 describes technical aspects covering requirements for software and hardware components. We define the design process in Section 4.3. This includes interface design and descriptions of the main tasks. The last part, Section 4.4, highlights the deployment phase, from pipeline integration to releasing the application into production.

4.1 System Definition

The concept of a smart system for predicting a person's personality based on the way tweets are written has been emphasised by recent initiatives. It has emerged as a response to the perceived problem that classification is a complex process in which data must undergo in a smart system (Silvis-Cividjian,

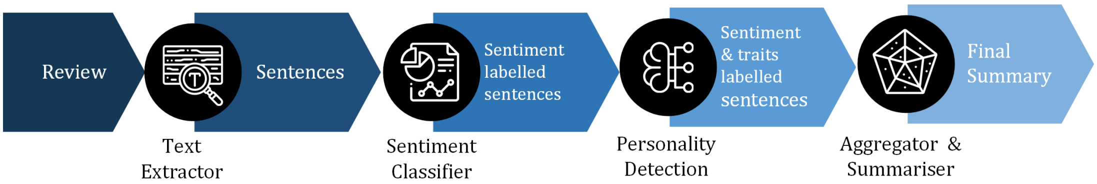


Figure 4.1: System overview: Predicting a user's personality from opinionated texts.

2017). PERSOMA (Lima & de Castro, 2014), which stands for Personality Prediction in Social Media Data, is a fine example. The system has been able to predict specific personality traits present in groups of tweets based on the Big Five model. More recently, Carducci et al. (2018) have developed an application called TwitPersonality to compute personality traits by only relying on what an individual tweets about publicly. The aforementioned examples illustrate the advantage of employing an automated system to perform a complex task. Those concepts, as well as details of other relevant aspects, are further adjusted in the development of the proposed system in the current study.

A *system* is generally described as the combination of interacting elements, organised to achieve one or more stated purposes (Silvis-Cividjian, 2017, p. 129). Sentiment-based Personality Detection (SENTIPEDE) is a web-based system which allows users to input a string, or a file containing opinionated texts, while providing the tools for automated personality prediction. The system seamlessly enables functions to be made available pervasively via the Internet. In other terms, it is characterised as ubiquitous or *pervasive computing* (Silvis-Cividjian, 2017). As illustrated in Figure 4.1, the proposed system contains multi-functional modules that can perform data extraction, sentiment classification and traits detection to present a personality assessment. It was designed to react upon input data, and adapt the output based on external input parameters. This ability is considered to be computationally intelligent—or *smart*. Hence, a

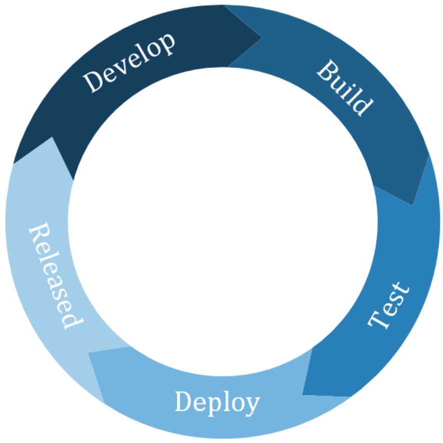


Figure 4.2: Continuous delivery. A software engineering approach which aims at building, testing, and releasing software within short cycles.

system that depends on such computational intelligence can be described as a smart information system (Hopfgartner, 2015). The basic stages involved in the development of the proposed system are explained further in the following sections.

4.1.1 Software Development Model

The software development process follows the same steps as in *systems engineering*, which consists of *specification*, *design*, *implementation*, *testing*, and *maintenance*. The most commonly used software development model, *Waterfall*, follows the same steps on a time line. Consequently, the development process can take months to a year to complete. In the *Agile* model, however, the process is represented as an incremental and iterative approach. In this manner, the intermediate product is exposed to user feedback more often and evolved through several versions (Silvis-Cividjian, 2017). As a result, the development becomes more efficient. Related to Agile, another model that relies on the repetition of a very short development cycle is Test-Driven Development (TDD). This approach allows coding, testing, and design runs to be tightly interwoven.

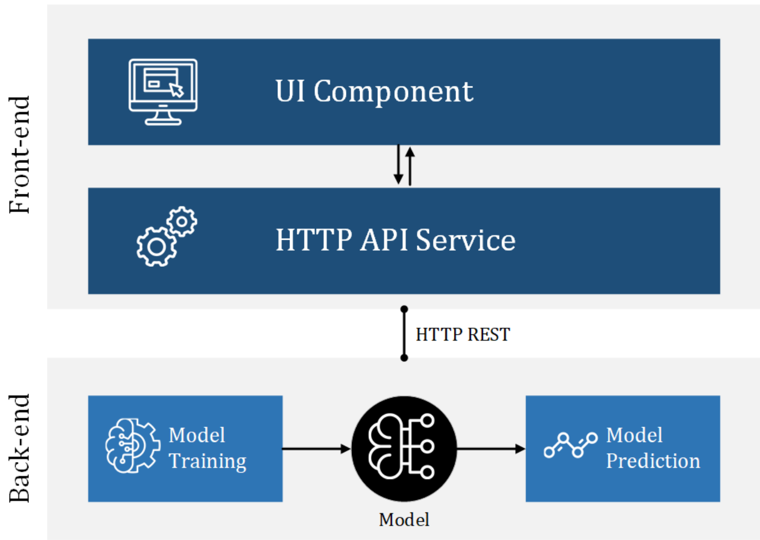


Figure 4.3: Descriptive diagram of deployment of a machine learning model in a web application. Adapted from *Deploying a simple machine learning model in a modern web application* by D. Elsner, 2018. Retrieved from https://medium.com/@dvelsner. Copyright 2018 by Daniel Elsner.

While software developers have benefited from Agile development methods, Continuous Delivery (CD) has emerged as a top priority for Agile environments. CD is "a series of practices designed to ensure that code can be rapidly and safely deployed to production" (Daya et al., 2015, p. 61). This approach aims at delivering software that meets requirements through rigorous automated testing, as illustrated in Figure 4.2. Although there is no one broadly accepted model suite for all software development projects, in this current work, we followed CD for system development.

4.1.2 Software Architecture

Most available platforms used for machine learning are focused on functionalities for developing and tuning models. Less attention is paid to presenting the trained models as an end-user product. In the present work, we attempted to deliver an

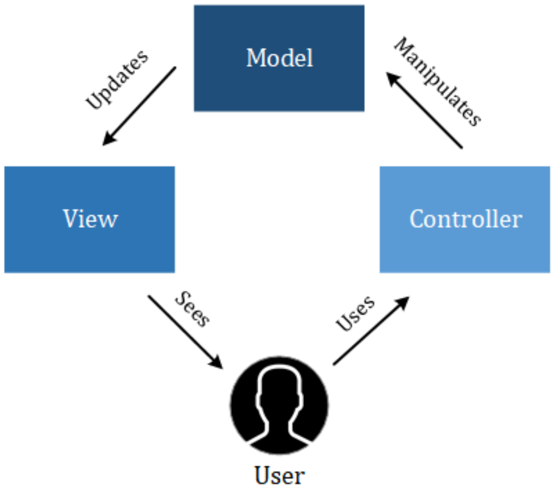


Figure 4.4: The model-view-controller component diagram. Adapted from "An Approach of a Framework to Create Web Applications" by D. Sanchez Rodriguez, O. Mendez, and H. Florez, 2018, in *Computational Science and Its applications – ICCSA 2018* (pp. 341–352). Cham, Switzerland: Springer International Publishing.

interactive web application embedded with models discussed in Chapter 3. We developed our system based upon the architecture diagram proposed by Elsner (2018), which can be seen in Figure 4.3.

The diagram exhibits the deployment of machine learning modelling into a web application. From the figure, the terms *front-end* and *back-end* refer to the *separation of concerns* (SoC). In this fashion, a front-end developer of the UI works on the presentation layer, while the back-end developer works directly in the data access layer of the physical infrastructure (Sanchez Rodriguez, Mendez & Florez, 2018). This modular approach is thus adopted in an agile environment to create a web application.

Throughout the system, we used the *model-view-controller* (MVC) pattern extensively. Corresponding to the SoC framework, this pattern defines the architectural model that separates data from the UI (Sanchez Rodriguez et al., 2018). In practical terms, *model* represents the domain of the software and contains application data, while *view* is the visual representation of the model obtained by managing the user interface. *Controller*, on the other hand, is responsible for receiving user requests processing them and deciding on the actions to be performed. The component diagram of the MVC pattern is illustrated in Figure 4.4.

4.2 System Specification

In a pervasive computing system, software and hardware work collectively to enable the desired functionality (Silvis-Cividjian, 2017, p.129). It is necessary to provide a clear description of the both software and hardware components, technologies, and equipment required. Therefore, the main objective of this section is to present those aspects, which are outlined below.

4.2.1 Prerequisites

The set of tools and packages required to create the software product are described below:

- **Flask**. Flask[1] is a micro web framework written in Python. It allows the development of an API or a web application that responds to the request.
- **Bitbucket**. Bitbucket[2] is a web-based repository hosting service which provides remote code storage and control of the software version.
- **Docker Hub**. Docker Hub[3] is a cloud-based repository service to manage a *container image*—an unchangeable and static file comprised of system

[1] http://flask.pocoo.org/
[2] https://bitbucket.org/product
[3] https://hub.docker.com/

libraries, tools, and other settings that a software program requires to run on a containerisation platform.

- **Puppet Pipeline**. Puppet Pipelines[4] is an automating infrastructure and software management engine. It simplifies continuous software delivery using an agentless, task-based approach.

- **Amazon Elastic Compute Cloud (EC2)**. This web service provides secure, sizeable compute capacity in the cloud. Designed for web-scale cloud computing development, EC2[5] allows developers to control the resources run on Amazon's proven computing environments.

4.3 Design Process

In this phase, we identify the specific designs for the system. We define interface design and provide more details of the workflow process. The design process took in overlapping stages, as explained below.

4.3.1 Interface Design

In designing a UI, we adopted the Responsive Web Design (RWD) approach. In RWD, both development and design respond to the user's behaviour and environment regardless of the screen size, platform, or orientation (Mohorovičić, 2013). In this manner, a web site is developed in a way that means it can be adapted to almost all devices. In order to achieve RWD, an open source package for front-end development called *bootstrap*[6] was used. It supports a

[4]https://puppet.com/products/puppet-pipelines
[5]https://aws.amazon.com/ec2/
[6]https://getbootstrap.com/

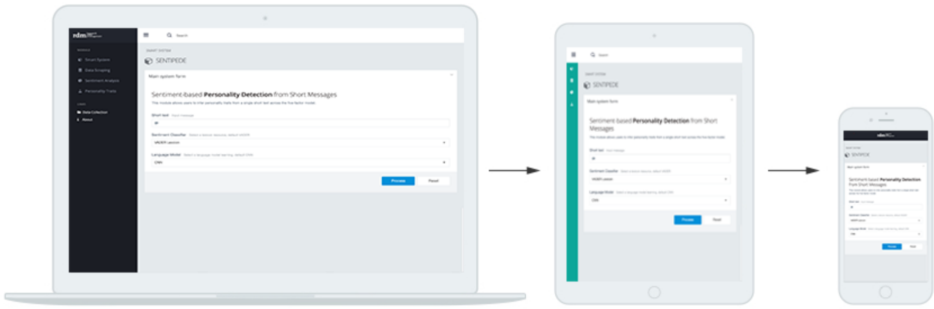


Figure 4.5: An adoption of responsive web design: SENTIPEDE web site loaded on several devices.

media queries module in delivering a tailored style sheet to different devices. Figure 4.5 demonstrates the adoption of RWD in our system.

4.3.2 Component Design

The SENTIPEDE was designed to allow the user to set parameters through a user interface. In response, the trained models compute and present users with the predicted probabilities. There are four main functionalities included in this system. Isolated in *modules*, they are: (1) *Main Module*, (2) *Twitter Data Scraper Module*, (3) *Sentiment Classifier Module*, and (4) *Personality Detection Module*. Each module consists of one or more components and works independently at the same time on the same flow of information, as shown on Figure 4.6.

4.3.2.1 Main Module

The main page of the web interface shows the inputs form for the sentence-level sentiment-based personality detection task. The module provides a text input to be filled with string. Users can also choose a classification method and a language model available from drop-down lists. The screenshot of the main page can be seen in Figure 4.7.

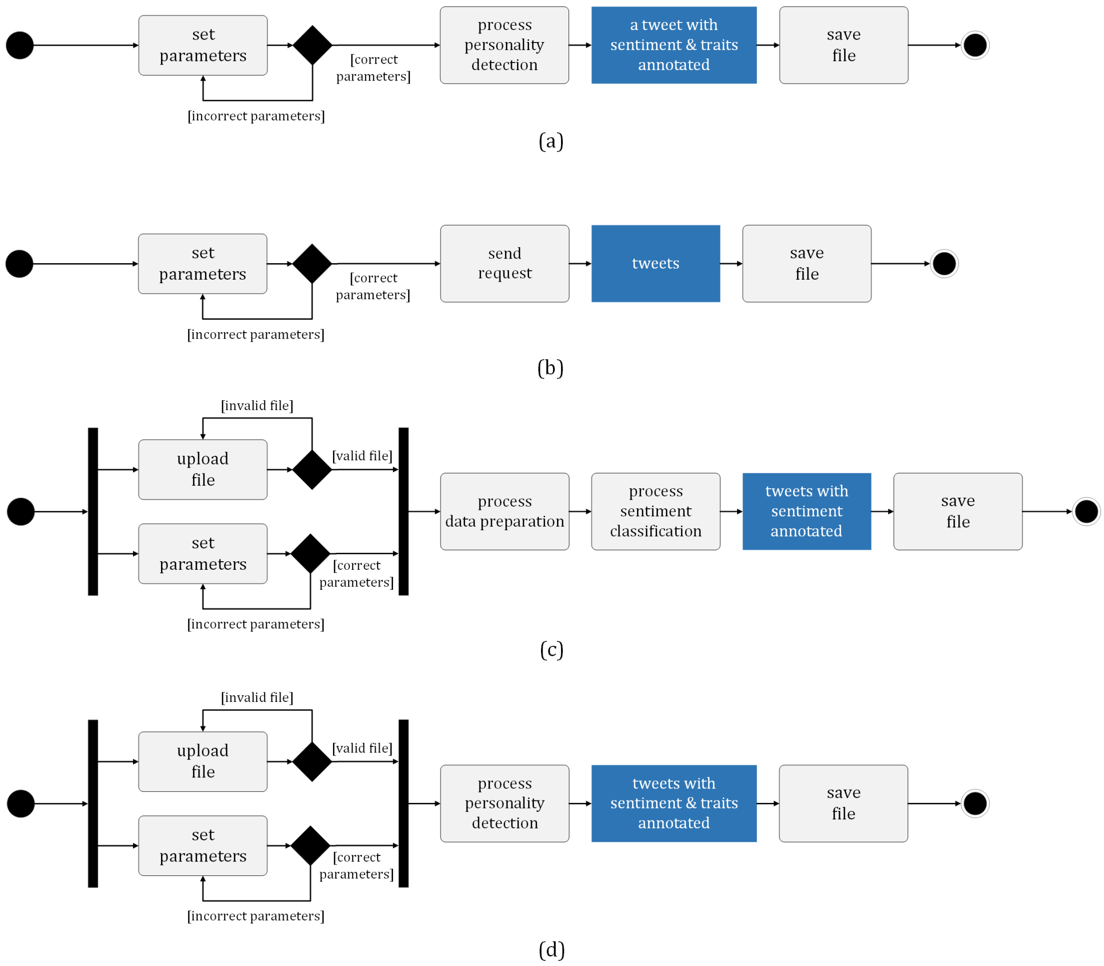


Figure 4.6: Activity diagrams representing the flow of each function the system offered: (a) Sentence-level sentiment-based personality detection, (b) Twitter data scraper, (c) Sentiment analysis, and (d) Personality detection.

The module processes the prediction task based on the parameter values, as described in Table 4.1. First, it cleans the inputted string, and passes it to the sentiment classifier, which annotating the string with a sentiment polarity. The personality traits detection is performed once the string is labelled. This returns the scores for each trait, as shown in Figure 4.8.

Table 4.1: Main system input argument: Different fields, their default settings and variable types

Field	Description	Options	Default	Type
shortText	Input message	*not available*	*none*	string
lexicon	Lexical resources	'VADER'/'AFINN'/'SentiWordNet'	'VADER'	string
languageModel	Language learning model	'CNN'/'LSTM'/'CNN+LSTM'/ 'GloVe+CNN'/'GloVe+LSTM'/ 'GloVe+CNN+LSTM'	'CNN'	string

Table 4.2: Data scraping input argument: Different fields, their default settings and variable types

Field	Description	Options	Default	Type
userName	Input username	*not available*	*none*	string
query	Input hashtag or mention	*not available*	*none*	string
dateSince	Fetching date starts (*yyyy-mm-dd*)	*not available*	*none*	date
dateUntil	Fetching date ends (*yyyy-mm-dd*)	*not available*	*current date*	date
maxTweets	Maximum number of tweets	*not available*	100	integer

Table 4.3: Sentiment analysis input argument: Different fields, their default settings and variable types

Field	Description	Options	Default	Type
file	Upload a CSV file	*not available*	*none*	file
lexicon	Lexical resources	'VADER'/'AFINN'/'SentiWordNet'	'VADER'	string

Table 4.4: Personality detection input argument: Different fields, their default settings and variable types

Field	Description	Options	Default	Type
file	Upload a CSV file	*not available*	*none*	file
lexicon	Lexical resources	'VADER'/'AFINN'/'SentiWordNet'	'VADER'	string
languageModel	Language learning model	'CNN'/'LSTM'/'CNN+LSTM'/ 'GloVe+CNN'/'GloVe+LSTM'/ 'GloVe+CNN+LSTM'	'CNN'	string

4.3.2.2 Twitter Data Scraper Module

The aim of this module is to extract tweets related to a given query, historical data and users' specific timelines. The tweet is gathered based on the username, hashtag or mention, fetching dates, and the maximum number of tweets.

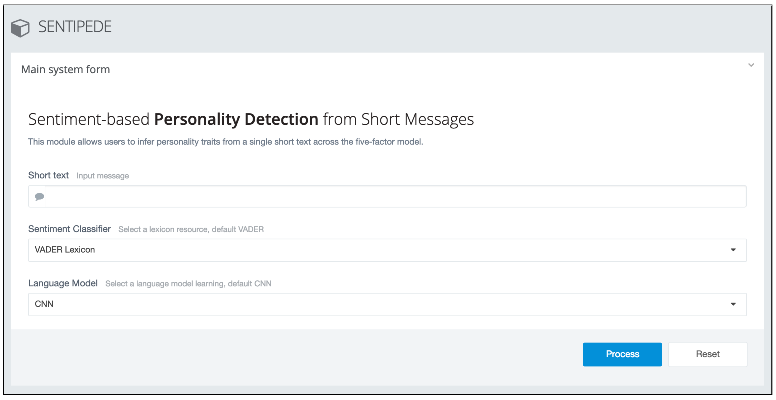


Figure 4.7: A screenshot of the main page. The module allows users to infer personality from a sentence.

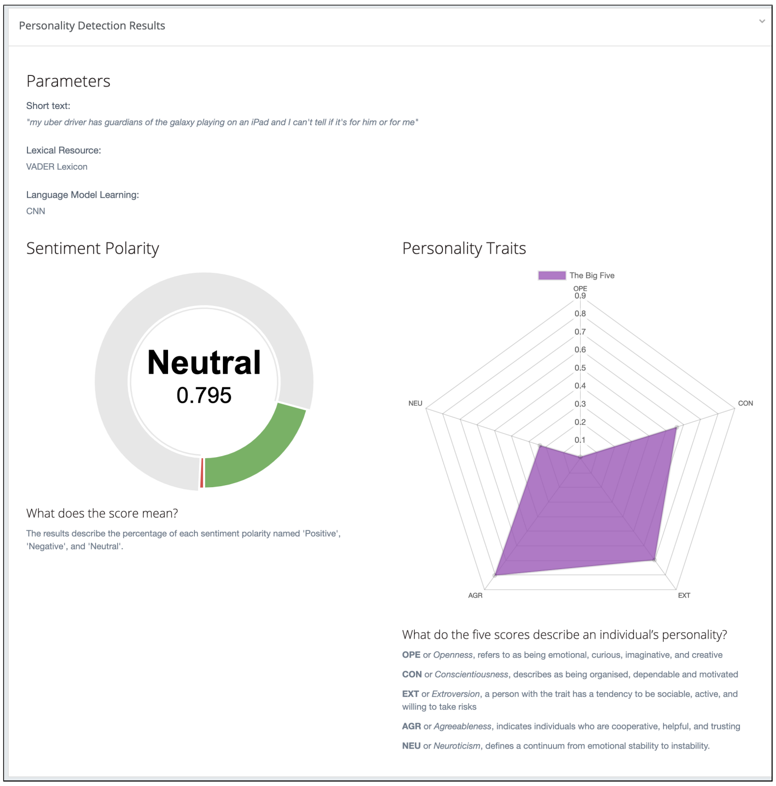


Figure 4.8: Screenshot shows the results of the personality detection from short text.

Table 4.2 describes the input argument details.

If a user clicks on the *'Process'* button which can be seen in Figure 4.9, the request is sent to Twitter through the HTTP Server. Following this process, Twitter issues a response by rendering tweets into the web, which enabling user

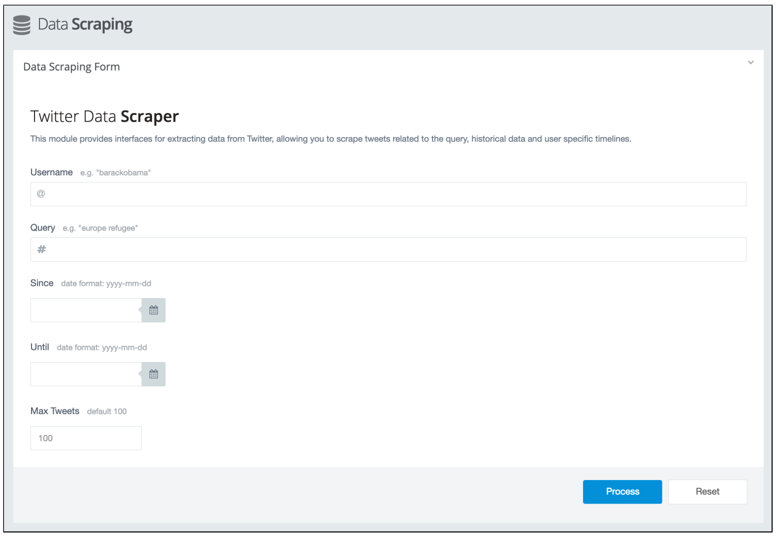


Figure 4.9: Data scraping page: The screenshot shows the parameters for tuning the request. Users might assign specific queries or define the fetching dates.

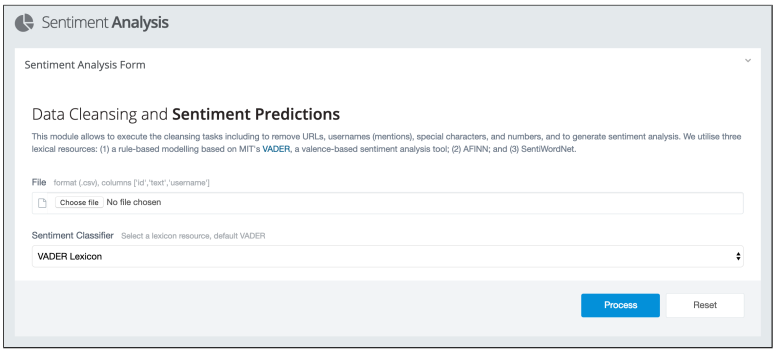


Figure 4.10: Sentiment analysis page: The system allows users to upload files and provides options for the lexical resources to be employed in the classification process.

to download or save a file in comma-separated values (CSV) format.

4.3.2.3 Sentiment Analysis Module

In this module, a sentiment analysis is performed. The given tweets are processed based on the selected lexicon-based classifier. The description of each parameter is presented in Table 4.3.

The sentiment analysis page is shown in Figure 4.10. From the figure, once the '*Process*' button is clicked, the system performs the data cleaning process,

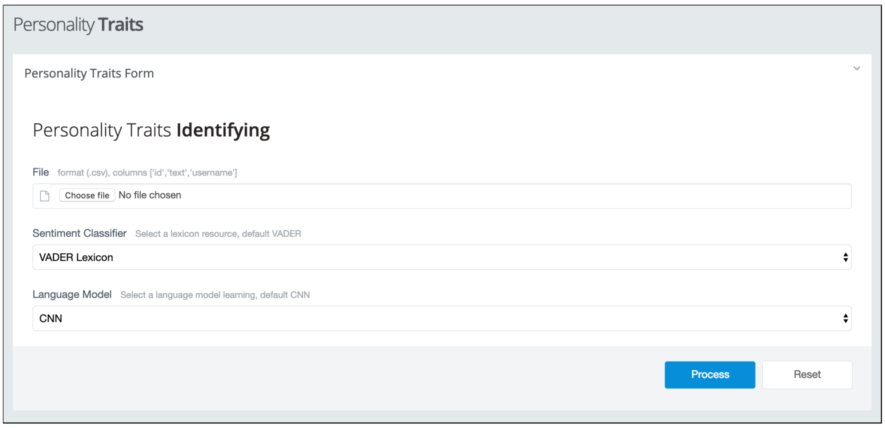


Figure 4.11: The screenshot of the personality detection page shows the function predicting the user personality matching the user file input.

and then applies the sentiment classification task. In this step, the classification model is called in accordance to the type of lexical resource assigned. When the process is completed, users can download or save the output file.

4.3.2.4 Personality Detection Module

The objective of this module is to run personality detection tasks based on the given input arguments, which are described in detail in Table 4.4. Once it is started, the module calls the Twitter sentiment classification model which is responsible for data cleaning and pre-processing, and classifying the given tweets. The process is completed when the module applies the prediction models and assigns the tweets with personality trait scores following the selected language model learning. The web page of the personality detection module is shown in Figure 4.11.

4.4 System Implementation

In the previous sections, we have explored the initial stages for developing the software. This section describes the implementation phase. It involves

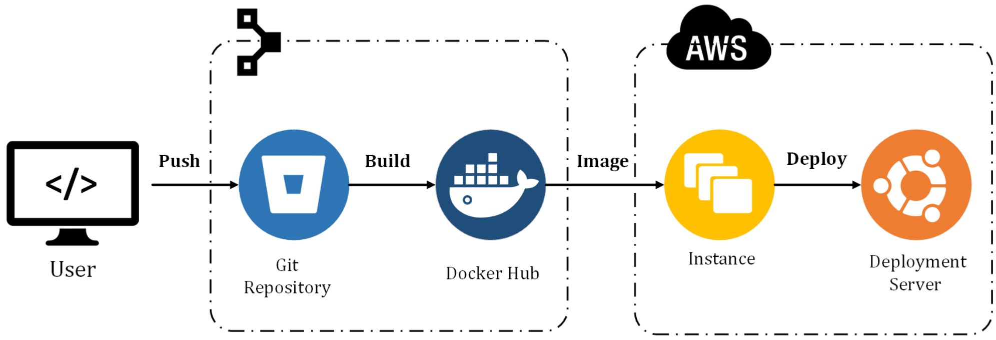


Figure 4.12: Software deployment pipeline adopting the Continuous Delivery approach wherein users push codes to build and deploy an application into production.

constructing the system elements created during the architectural design and the result of system analysis. These system elements are then integrated to deliver a complete system that meets the requirements. In this work, we adopted the CD approach to ensure a rapid pipeline from development to test and production, as can be seen in Figure 4.12.

4.4.1 Building the Application

The initial phase is to create a git repository and push the source code. Git is a distributed version-control system for tracking changes in source code during software development. In the first step, we used the Flask framework to create a web-based system and deploy our machine learning models into the web service. The next step is where we started to build and deploy a docker image. Described as a read-only template to establish an application, a docker image is built by creating a set of instructions called a *dockerfile*. The file includes information such as the base Operating System (OS), programming language used, and the packages required to compile an application.

```
sentipede
├── model/
├── controller/
├── templates/
│       └── index.html
├── static/
│       ├── img/
│       ├── css/
│       └── js/
├── Dockerfile
├── requirements.txt
└── app.py
```

Figure 4.13: A directory tree structure of SENTIPEDE adopting an MVC model separating views from models and controllers.

We wrote our software's code in a modular form based on the MVC framework, so that each component has its own sub-directory, as illustrated in Figure 4.13. We then ran some software testing separately, before the final source code was pushed into a code storage area called a *repository*. Through this repository we can control the software versioning remotely from a local machine. Once the application was tested and operational, we moved to the deployment process. This is explained in the following sections.

4.4.2 Server Configuration

In this phase, we performed an initial server setup and created a Linux-based Puppet agent. We used EC2 to build a virtual server in the cloud known as an *instance*. Within EC2, we specified the Amazon Machine Image (AMI), a template that contains the configuration required to launch the instance (Wittig, Wittig & Whaley, 2018). The first step was to choose an OS. An EBS storage volumes-type and general purpose instance with Ubuntu OS was chosen. Next, a network firewall was added to control both inbound and outbound traffic. This includes access rules to the HTTP and SSH ports, which were configured in the

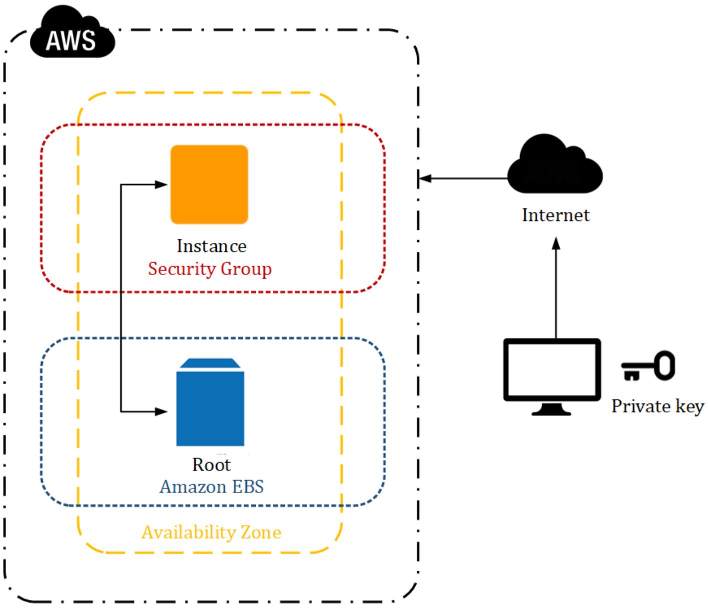


Figure 4.14: An Amazon EBS-backed instance. From *Amazon Elastic Compute Cloud: User Guide for Linux Instances*. Retrieved from https://docs.aws.amazon.com/ec2/index.html. Copyright 2019 by Amazon Web Services, Inc.

Security Group. Finally, the virtual machine's keys were paired to log into the instance. The graphic representation of the Amazon EC2 instance is shown in Figure 4.14.

Once the instance is launched, it is automatically booted to the server selection OS. The next step was to create an agent. Puppet agent is the application that manages configurations on both the server and the local host, so that it facilitates the continuous deployment process. The following syntax was used to install the Puppet agent: `wget -qO- https://pipelines.puppet.com/download/client`

4.4.3 Deployment

This phase aims to deploy a releasable built application into production. In this phase, the process is manually guided by Puppet Pipelines. It began by connecting the application to the source control, which is the repository created

Figure 4.15: A screenshot of the deployment staging on Puppet Pipelines. This entailed four steps: pushing the source code, building an application, creating an image, and deploying the release version into production.

in the previous section. This grants an administrator access to auto-build commits and pull request by adding a *webhook*—a HTTP push API. The second step entailed selecting a docker image to build a production-ready application. The final version of the application was released after running a series of tests on the application. This then was made live on the production environment of an existing server, which is running Ubuntu, and installed with an Puppet agent. The deployment staging on Puppet Pipelines is shown in Figure 4.15.

4.5 Summary

In this chapter, we demonstrated the development stages of our system. In the initial stage, we defined the system architecture and specification including the prerequisites required to build the proposed system. This was followed by the designing stage. In this phase, we specified the user interfaces and described the workflow design and parameters of each functionality. In the implementation stage, we adopted the CD approach to deploy the system into production. It first entailed pushing the source code into a git repository. In the second step, we started to build our application which involved initiating a docker image that contains the configurations of the base OS, the programming language and packages required. We also created an Amazon EC2 instance that runs Ubuntu Server and was installed with a Puppet agent. Finally, we deployed a release version of our system into production. This was done by attaching Puppet Pipelines to enable automated build and push changes to the server. The adoption of our system is evaluated in the following chapter.

Chapter 5

Results and Evaluation

Following on from the completion of the system development, this chapter is dedicated to providing the experimental results obtained and reporting the case study findings achieved in adopting the proposed system. The chapter has three parts. It begins with the configurations for lexical resources and parameters for Neural Network Language Modelling, which is the topic of Section 5.1. Next, we evaluate the model's performance on both tasks, sentiment classification and trait inference, as explained in Section 5.2. The last part will focus on a case study-based investigation. The case study set out to investigate the interrelation of user personality and perceptions, and is presented in Section 5.3.

5.1 Experimental Setup

The experiments were performed on a computer running macOS Mojave with specifications as follows: 2.9GHz Intel Core i5 with 8GB RAM memory and 512GB flash storage. All learning models were written in Python 3.6.0 with Jupyter Notebook on a virtual environment created using Anaconda Navigator. Also, multiple spreadsheet files were prepared to document the results for each model. In addition to that setup, various different configurations and settings were also applied, which are further described in the following subsections.

5.1.1 Configurations for Lexicon-based Sentiment Analysis

As the sentiment classification task relies heavily on the lexicon, setting standardised thresholds for classifying sentences is crucial. In this phase, we configured a classification threshold for each lexical resource included in this work to determine sentences as either positive (POS), neutral (NON), or negative (NEG), as follows:

- **AFINN**. In AFINN, as the return values are categorised from 'very positive' to 'very negative', to produce three groups of sentiment polarity (i.e., POS, NEG, and NON), we defined the threshold values to [1] and [-1]. In this regulation, sentences scoring higher than [1] were set as POS, and those that scored lower than [-1] were automatically classified as NEG. Sentences excluded from the preceding conditions were categorised as NON.
- **VADER**. To score the polarity using VADER,[1] we set the compound scores threshold for POS sentiment to be greater than [0.05]; for NON sentiment, scores were between [-0.05] and [0.05]; and for NEG sentiment score were less than [-0.05].
- **SentiWordNet**. A distinct treatment was applied to SentiWordNet. Since the lexicon classifies words into positive, negative, and objective, we thus set thresholds to normalised scores. NON sentences have a score between [0.01] and [-0.01]. Sentences that have a score greater than [0.01] were set as POS, while sentences scoring less than [-0.01] were set as NEG.

[1] https://github.com/cjhutto/vaderSentiment

5.1.2 Model Tuning

Fine tuning a predictive model is an important step as it determines the accuracy of the predicted results. In this phase, we applied an approach that encompasses model tuning entailing data partitioning. We split the original data set into distinct sets which were used to create the model and for periodic evaluation of accuracy respectively. This process was crucial to prevent the occurrence of *over-fitting* or *under-fitting*. We allocated the data with an 80-20 split. To provide optimal coverage of each class in the data set, we also performed shuffling to both training and test data.

5.1.3 Defining and Compiling Networks

In this stage, hyper-parameters for the neural networks models were set. Once the network had been defined, it was ready to be compiled. Figure 5.1 illustrates the unified model of GloVe+CNN+LSTM for personality detection from short texts.

i. **Word Vector Initialisation**. To start this process, word vector initialisation was performed. As mentioned previously in Section 2.2, in an NNLM, the use of a dense distributed representation for each word is the key to the method. The current work utilised GloVe **pre-trained word vectors** for Twitter with a dimensionality of 200. Sahu and Anand (2015) revealed that 200 dimension distributed word representations perform better for NLP tasks entailing GloVe model. Only the top 20,000 most commonly occurring words in the data sets were used and the sequences were truncated to a maximum length of 1,000 words. The texts were selected randomly for training and the remaining texts were used for testing.

ii. **Neural Network Layers**. We used a simple **convolutional layer** consisting of 64 trainable filters that are convolved across the input matrix. Afterwards, outputs of the convolutional layer are sub-sampled by a **max-pooling layer**. So our next layer is an **LSTM layer** with 100 memory units.

iii. **Activation Functions**. We later experimented with dropout and an activation before concatenating to a fully connected layer. The goal of **dropout** is to randomly drop nodes along with their connections from the neural network during training. This can prevent nodes from co-adapting, a process by which two or more nodes behave as if they are a single node (Hahn & Choi, 2018). In general, softmax activation is used for multi-class classification. Although it can also be used for binary classification, in this stage, we used sigmoid funtion. A **sigmoid activation** is a logistic function that normalises the dimensional vectors of arbitrary real values to a probability distribution over predicted output classes that range from 0 to 1.

iv. **Dense Layer**. Finally at the end we have a dense layer with one node and a sigmoid activation as the output. As we are going to predict probabilities of each class, we used **binary cross-entropy** for the loss function. The optimiser is the standard one (**adam**) and the metrics are also the standard accuracy metric. We ran our test of every itinerary for 10 epochs.

5.2 Model Performance

This section presents the results of model validation. All models were trained on a cleaned training set. We used the performance metric to measure the effectiveness of classification models. Performance evaluations were conducted

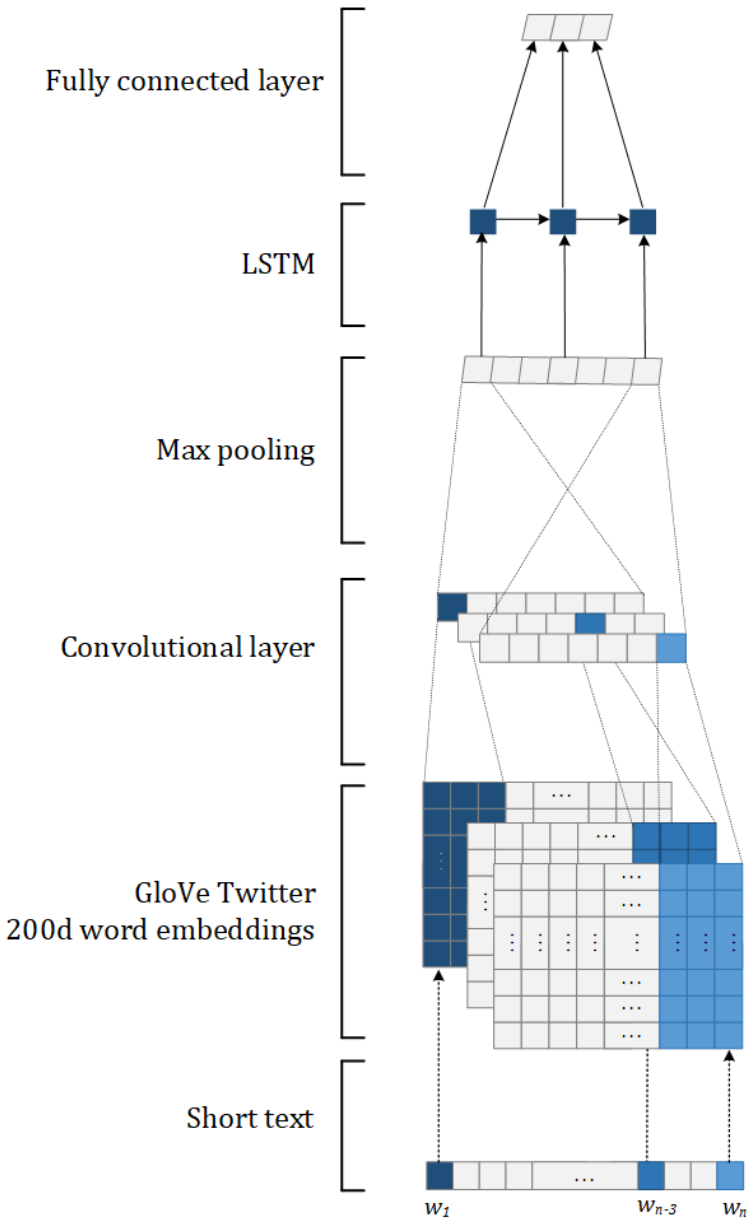


Figure 5.1: Overview framework of the proposed model: a unified model of GloVe+CNN+LSTM for personality detection from short texts.

for the two main tasks: the sentiment classification task and the personality detection task, which are explained in the following subsections.

Table 5.1: Comparison of English lexicon results for the sentiment task. Precision and recall were calculated for positive, negative, and neutral sentences

	Data sets		
Measure	STS	STS-Gold	STS-Test
SentiWordNet			
F-Score	63.59	67.23	57.88
Precision	65.28	75.92	58.84
Recall	64.24	66.08	58.03
VADER			
F-Score	**65.23**	**81.93**	68.87
Precision	**67.36**	**81.83**	70.20
Recall	**65.92**	**82.10**	68.88
AFINN			
F-Score	63.46	78.80	**72.34**
Precision	63.94	80.17	**73.71**
Recall	63.64	78.27	**72.29**

Note. **Bold** highlights best performance.

5.2.1 Sentiment Classification Task

In order to evaluate the classification results, we computed the *Precision*, *Recall* and *F1* measurement. We compared the performance metrics of each lexicon to all STS data sets, as provided in Table 5.1. The results vary significantly for each classifier. As seen in the table, the VADER lexicon provided high precision and recall values for STS and STS-Gold data sets, while AFINN achieved best for the STS-Test data set. In contrast, SentiWordNet performed worse than other classifiers across all three lexical resources. However, the precision achieved in classifying STS-Gold was slightly better than those for STS and STS-Test. Overall, in this evaluation, the VADER lexicon achieved the best performance.

After this evaluation, we performed sentiment classification on a set of 9,913 messages from the myPersonality corpus. We first applied the data cleaning process, this resulting in 9,847 cleaned data. Employing all three lexicon classifiers,

Table 5.2: Results for sentiment classification of the myPersonality data set.

	Lexical resource		
Category	SentiWordNet	VADER	AFINN
POS	3,882	4,243	3,892
NEG	2,923	3,350	2,123
NON	3,042	2,254	3,832

Note.POS=Positive,NON=Nonpartisan,NEG=Negative.

we obtained the results for sentiment classification of myPersonality provided in Table 5.2. As can be seen from the table, in general, each sentiment labelled data set contains a nearly equal number of positive (POS), negative (NEG), and nonpartisan (NON). The final data sets were then divided into training and test sets to be used for the personality detection task which is described in the next section.

5.2.2 Personality Detection Task

The personality detection task predicts the personality score for each trait, namely Openness (OPE), Conscientiousness (CON), Extraversion (EXT), Agreeableness (AGR), and Neuroticism (NEU), from the final data sets. Once we finished training the model we tested its accuracy. The experimental results of sentiment-based personality detection are described in the following paragraphs.

The first experiment employed the VADER lexicon. In most of five personality traits, the proposed methods out performed the baseline models. The unified model (GloVe+CNN+LSTM) provided the highest accuracy (61.13%) for predicting personality traits from the NON category. The model also showed significant results in predicting the CON trait in all categories. Adding GloVe to CNN, has improved the prediction accuracy to 59.11% for the NEG category. An exception was made for LSTM. Although GloVe+LSTM obtained higher scores for the OPE

Table 5.3: Validation accuracy for Sentiment-based Personality Detection with VADER lexicon and its variants

Method	OPE	CON	EXT	AGR	NEU	avgOCEAN
POS						
CNN	66.39	54.49	57.67	53.89	58.84	58.26
LSTM	65.57	56.96	**58.84**	**57.43**	57.43	**59.25**
CNN+LSTM	66.15	57.08	57.43	55.90	57.55	58.82
GloVe+CNN	**70.52**	57.19	55.07	52.00	57.55	58.47
GloVe+LSTM	68.75	55.90	54.36	56.13	**60.97**	59.22
GloVe+CNN+LSTM	68.28	**58.25**	54.85	53.42	58.84	58.73
NEG						
CNN	68.89	56.00	55.78	50.00	**60.22**	58.18
LSTM	60.89	54.00	**59.78**	52.00	55.33	56.40
CNN+LSTM	65.56	55.11	54.44	**55.33**	58.22	57.73
GloVe+CNN	**75.56**	54.89	55.78	51.78	57.56	**59.11**
GloVe+LSTM	68.67	56.44	56.67	51.11	58.44	58.27
GloVe+CNN+LSTM	70.67	**56.44**	54.67	50.89	58.22	58.18
NON						
CNN	68.06	56.27	57.01	57.46	58.96	59.55
LSTM	68.36	54.33	58.51	**57.91**	59.40	59.70
CNN+LSTM	70.15	55.97	57.46	57.31	**62.39**	60.66
GloVe+CNN	70.15	57.61	51.64	53.88	57.31	58.12
GloVe+LSTM	**74.63**	57.15	56.57	54.93	58.06	60.27
GloVe+CNN+LSTM	72.24	**57.91**	**58.51**	54.77	62.24	**61.13**

Note. POS=Positive, NEG=Negative, NON=Nonpartisan, OPE=Openness, CON=Conscientiousness, EXT=Extraversion, AGR=Agreeableness, NEU=Neuroticism, avgOCEAN=average accuracy. **Bold** highlights best performance.

trait in the NON category, the LSTM alone performed better than other models for two traits: the EXT trait in both POS and NEG categories, and the AGR trait in the POS and NON categories. In fact, the baseline model of LSTM achieved best in the POS group with 59.25% accuracy. The results of prediction accuracy obtained with different configurations are shown in Table 5.3.

Next, we employed the AFINN lexicon in the second experiment. As shown in Table 5.4, applying word embeddings in both CNN and LSTM can improve prediction performance. This can be seen in the NEG and NON categories, where GloVe+LSTM showed a good performance with 58.49% and 60.05% accuracy

Table 5.4: Validation accuracy for Sentiment-based Personality Detection with AFINN lexicon and its variants

Method	OPE	CON	EXT	AGR	NEU	avgOCEAN
POS						
CNN	65.94	55.91	**55.01**	55.27	60.15	**59.73**
LSTM	67.10	56.04	51.29	55.01	58.57	58.46
CNN+LSTM	64.91	**59.00**	54.63	56.56	60.15	57.60
GloVe+CNN	**68.64**	51.29	53.08	57.07	**63.88**	58.79
GloVe+LSTM	66.71	54.76	51.80	**57.07**	62.73	58.61
GloVe+CNN+LSTM	59.90	55.78	50.13	52.96	56.81	55.12
NEG						
CNN	63.21	**57.55**	**56.60**	53.30	54.72	57.08
LSTM	68.16	54.25	52.37	50.47	59.20	56.89
CNN+LSTM	61.56	54.48	53.07	**56.60**	**59.67**	57.08
GloVe+CNN	68.63	53.77	55.66	52.12	58.25	57.69
GloVe+LSTM	**72.41**	55.42	53.30	53.54	57.78	**58.49**
GloVe+CNN+LSTM	68.16	53.07	52.59	50.88	57.55	56.45
NON						
CNN	66.97	52.74	54.31	55.09	56.53	57.13
LSTM	65.67	56.27	55.35	**58.50**	56.92	58.54
CNN+LSTM	68.02	58.09	54.31	57.44	59.01	59.37
GloVe+CNN	70.76	54.05	**57.18**	53.00	59.27	58.85
GloVe+LSTM	71.15	**58.09**	56.40	54.96	**59.66**	**60.05**
GloVe+CNN+LSTM	**72.06**	54.83	55.35	50.65	54.96	57.57

Note. POS=Positive, NEG=Negative, NON=Nonpartisan, OPE=Openness, CON=Conscientiousness, EXT=Extraversion, AGR=Agreeableness, NEU=Neuroticism, avgOCEAN=average accuracy. **Bold** highlights best performance.

respectively. By contrast, CNN alone outperformed the proposed models and provided 59.73% accuracy for predicting traits from the POS category.

In the third experiment, we utilised the SentiWordNet lexicon. The validation accuracy results are described in Table 5.5. From the table, a promising performance has been demonstrated by our unified model. GloVe+CNN+LSTM obtained the highest average accuracy with 60.70% in predicting traits from the NON category. Combining GloVe with CNN also showed good accuracy for the POS category (59.51%). The baseline model of CNN exhibited generally good performance in the NEG category with 58.51% accuracy, which is slightly higher

Table 5.5: Validation accuracy for Sentiment-based Personality Detection with SentiWordNet and its variants

Method	OPE	CON	EXT	AGR	NEU	avgOCEAN
POS						
CNN	66.75	54.77	**57.60**	57.99	59.66	59.35
LSTM	63.66	**57.09**	56.83	54.38	57.99	57.99
CNN+LSTM	65.72	55.28	54.90	53.09	56.57	57.11
GloVe+CNN	**71.39**	54.25	57.35	53.87	60.70	**59.51**
GloVe+LSTM	70.23	53.48	53.22	58.24	**62.11**	59.46
GloVe+CNN+LSTM	68.56	56.31	51.93	53.73	59.15	57.94
NEG						
CNN	63.70	**59.32**	57.53	52.57	**59.42**	**58.51**
LSTM	66.95	51.20	56.00	52.91	57.88	56.99
CNN+LSTM	65.75	54.28	54.79	53.42	58.90	57.43
GloVe+CNN	**73.12**	51.89	**58.05**	54.45	53.42	58.19
GloVe+LSTM	67.98	57.19	56.69	54.11	55.31	58.26
GloVe+CNN+LSTM	61.82	53.77	55.48	**55.48**	53.60	56.03
NON						
CNN	65.79	58.89	53.95	53.45	61.18	58.65
LSTM	69.08	55.43	56.42	51.48	59.54	58.39
CNN+LSTM	69.08	57.73	**56.91**	**56.74**	60.53	60.19
GloVe+CNN	70.56	**59.70**	56.74	50.82	62.17	60.00
GloVe+LSTM	**75.82**	57.73	53.29	56.09	58.39	60.27
GloVe+CNN+LSTM	70.72	59.38	55.76	53.78	**63.82**	**60.70**

Note. POS=Positive, NEG=Negative, NON=Nonpartisan, OPE=Openness, CON=Conscientiousness, EXT=Extraversion, AGR=Agreeableness, NEU=Neuroticism, avgOCEAN=average accuracy. **Bold** highlights best performance.

than other proposed methods.

To summarise, the results indicate that using GloVe improves the accuracy of the prediction of traits from short texts. The proficiency of the VADER lexicon in classifying sentiment from tweets also helped to increase the accuracy of prediction results. Our combined model of GloVe, CNN, and LSTM out performed the baseline model's performance for all three sentiment groups, although with different configurations for different traits.

5.3 Case-based Investigation

A case study can be described as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context" (Yin, as cited in Oates, 2006, p. 142). In this section, we describe a case-based investigation into the relationship between personality traits and opinion polarity which was conducted employing the proposed system. We opted for the ride-sharing company Uber as the subject of this study. The topic was selected on the basis of the degree of attention received, which provided us with enough variability in the concepts we wanted to study. More specifically, this investigation intended to answer the following questions: (a) *What do the results tell us about the trend in the public opinion of the brand?*, (b) *What are the characteristics of users responding to the topic?*, and (c) *Were there any correlations between users' personalities and their perceptions?*.

5.3.1 The Uber Case

The sharing economy has rapidly emerged as a viable alternative and, inevitably, is shifting the face of the asset-lending market. Through a convergence of ideas and technologies, it has provided new value to economic agents who were previously had limited access to the market or were even excluded from it (Kasprowicz, 2016). This on-demand business model is enabled over a shared marketplace, collaborative platform, or peer-to-peer application. However, the emergence of the sharing economy not only benefits the marginal market participants, but also is disrupting traditional businesses. Uber's disruption of the taxi industry is a case in point.

Founded as UberCab by Travis Kalanick and Garrett Camp,[2] Uber's penetration in the transportation sector began in 2009. This new entrant offers an arguably more affordable, better user experience than public transit. By utilising the company's mobile application, passengers can hail a ride from private vehicle owners. With over 75 million passengers in 65 countries worldwide, Uber was reported to have reached net revenue of 2.8 billion USD in 2018, bigger than its competitors such as Lyft[3] and Grab[4] (Iqbal, 2019). However, while Uber was defending its market dominance, the long-established taxi industry was struggling. The sharing economy dramatically damaged their conventional business model. Taxi and rental car companies have become antiquated. The incumbents were compelled to adopt the collaborative economy platform (Kasprowicz, 2016). This disruptive force, in turn, leverages tension which often leads to public demonstrations and roadblocks, sometimes involving violence. France, Spain, Indonesia, and Brazil are some of many countries that have taken a rather hostile stand against this archetype of service (Palling, 2016).

Nevertheless, the public perception of sharing economy-based companies has changed considerably in the past few years. Uber's self-inflicted controversies has attracted the attention of social groups across the globe as streamed on social media, particularly via Twitter. While, many patronised the collaborative platform as reported in several European countries (Csaba & Reiner, 2016), the controversies surrounding the company throughout the years come at a price: public loyalty. This was clearly illustrated in 2017 when customers were urged to completely eliminate the service. As reported by Cresci (2017), social tags like *#BoycottUber* and *#DeleteUber* topped the 2017's trending topic in the U.S as public reaction to the company's surge pricing during a taxi strike. A similar case

[2]http://uberestimate.com/timeline/
[3]https://www.lyft.com/
[4]https://www.grab.com/

happened in Australia with Uber reportedly increasing its fares after Sydney's hostage crisis (Vinik, 2014). The calls to boycott the brand continued recently in the Gulf region, following the disappearance of a journalist from Saudi Arabia—a country which is listed as one of the Uber's major investors (Lomas, 2018). Together, these reports signify that Uber, as a globally renowned company, has attracted considerable attention in society, especially through social media where news spreads rapidly. However, thus far, no study has been done on the effect of user personality on public perceptions relating to the brand.

5.3.2 Data Acquisition

In this section, we explain how we used tweets that explicitly refer to the Uber brand as research data. The process commenced with the acquisition of data utilising the Twitter Data Scraper module in SENTIPEDE. This was performed by crawling tweets filtered by hashtag and mention with queries of *#uber* and *@uber*. We set the fetching dates from January to December 2018. Once the data were collected, we performed a data preparation process involving the elimination of URL links, numeric and special characters, mentions, and retweet identifiers. The final version of the data set was formed after applying tokenisation and stop word removal to the original corpus.

Figure 5.2 depicts the monthly volume of tweets gathered. A total of 120,975 tweets in the English language were collected containing tweet IDs, tweets, dates, mentions and permalinks. From the figure, it can be seen that the highest volume of tweets collected was recorded in March 2018, with 13,450 tweets collected, while the lowest was reported in July in the same year, with 5,676 tweets. On average, there were over 10,000 tweets mentioning or relating to Uber posted per month in 2018.

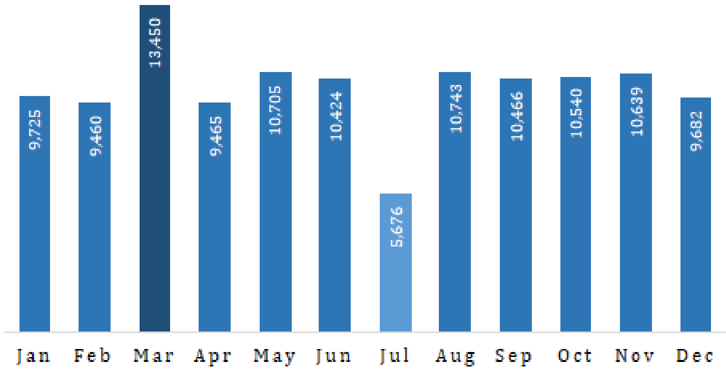


Figure 5.2: Monthly volume of tweets relating to Uber in 2018.

5.3.3 Results and Findings

On completion of data acquisition and preparation, two processes were carried out: (1) sentiment analysis to determine public perceptions of Uber; and, (2) personality detection to infer the traits of users who tweeted about the company. The configurations and results of these processes are explained in the following subsections.

5.3.3.1 Uber Sentiment on Social Media

In this phase, sentiment classification categorised the data into three groups, namely positive (POS), negative (NEG), and neutral or nonpartisan (NON). It was revealed previously in Section 5.2.1 that, compared to other lexical resources, the VADER lexicon has performed best in Twitter sentiment classification. Therefore, we used VADER as the lexicon-based classifier. Table 5.6 shows the Uber sentiment classification results.

As described in the table, at the start of 2018, the public perspective on Uber was positive but restrained. Almost half of tweets (45.75%) expressed

Table 5.6: Results for sentiment classification of tweets relating to Uber in 2018.

	POS	NEG	NON	Total
January	4,450	2,876	2,399	9,725
February	4,300	2,916	2,244	9,460
March	5,393	5,084	2,973	13,450
April	4,390	2,797	2,278	9,465
May	4,831	3,284	2,590	10,705
June	4,736	3,447	2,241	10,424
July	2,637	1,736	1,303	5,676
August	4,984	3,189	2,570	10,743
September	4,991	3,075	2,400	10,466
October	5,194	2,976	2,370	10,540
November	5,007	3,192	2,440	10,639
December	4,537	2,990	2,155	9,682
Total	55,450	37,562	27,963	120,975
Mean	4,620	3,130	2,330	10,081

Note. POS=Positive, NEG=Negative, NON=Nonpartisan.

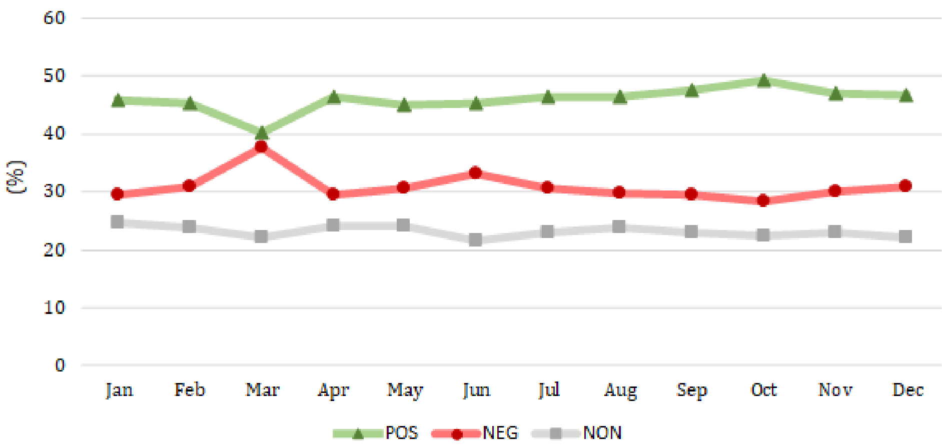


Figure 5.3: 2018 year in review: Sentiment towards Uber, analysed from tweets fetched between January and December.

positive sentiment towards the brand, and only 29.57% expressed a negative reaction, while around a quarter of tweets (24.67%) were categorised as neutral. However, in March, negative views reached their highest point with 38%, yet this trend changed quickly in the following months. Overall, through the year,

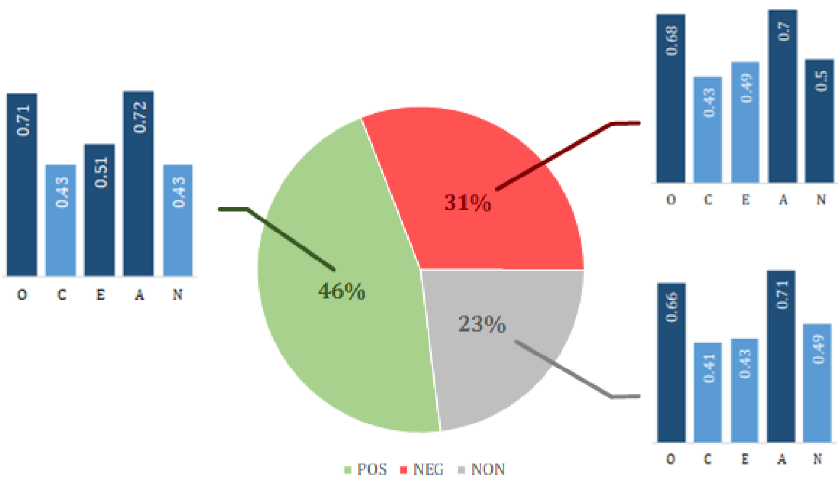


Figure 5.4: Sentiment towards Uber accompanied by the characteristics of each group of users. The results were analysed by SENTIPEDE.

the public grew more positive about Uber. A graphical representation of Uber sentiment month by month through 2018 can be seen in Figure 5.3.

5.3.3.2 Uber Users' Profile

In order to examine the relationship between personality and sentiment, 100 personal Twitter accounts (anonymised) were selected randomly from each group, POS, NEG and NON. We picked 200 tweets from each user, in accordance with the study of linguistic measure variability conducted by Haber (2015). Following this, the personality detection process was then applied to each profile. We utilised the personality detection module of the proposed system. To obtain the prediction scores across the five traits, the unified model (GloVe+CNN+LSTM) was selected. SENTIPEDE then transformed the sentence from each given tweet into the corresponding word integers and then transformed it into GloVe's sparse word vectors. Next, the system fed the vectorial word representations into the model according to the sentiment polarity they carried. Each model then estimated the predicted probabilities for each trait, namely Openness (OPE), Conscientiousness (CON), Extraversion (EXT), Agreeableness (AGR), and Neuroticism (NEU). We ran aggregate tweets per profile by calculating the average score of each trait. Figure 5.4 visualises these results.

In the figure, the OCEAN scores are presented in groups of sentiments. Our investigation shows that although the OPE trait is generally consistently high across the groups, persons who have expressed positive reactions (46%) are also characterised by having a high score on AGR and slightly heightened score on EXT. Additionally, an individual belonging to this group has a tendency to score low in both CON and NEU traits. However, OPE, AGR, and NEU traits are more dominant for users who expressed a negative view (around one-third). They are more likely to score low in CON and EXT. In contrast, those who stay neutral, which is almost a quarter (23%) of individuals, score high in OPE and AGR, with the three other traits (CON, EXT, and NEU) less dominant.

5.4 Summary

In this chapter, we reported the sentiment classification and personality detection task results. Three lexical resources were used in this experiment. Sentiments were extracted according to threshold scores set as appropriate for each lexicon. In addition, data splitting was performed to divide the original data set into training and test sets. We built the models on the training set and configured different hyper-parameters for each model employed for predicting personality traits. Finally, we evaluated the performance of each model for different configurations. This chapter ended with the presentation and analysis of the case study data. In the next chapter, we turn our attention to discussing the preceding experimental results and findings.

Chapter 6

Discussion

This chapter discusses the significant findings drawn from our study concerning the evaluation of the SENTIPEDE system performance. We commence this discussion chapter by first reviewing our research objectives. As stated earlier in this thesis, the present work set out to achieve two main goals: first, to build neural networks language models to predict personality from short messages that incorporate sentiment information in the text; and, secondly, to better understand the existing correlation of personality and public perceptions. In pursuit of those research goals, we have developed a smart system, conducted a case study-based investigation, and presented the results of these steps sequentially. In this chapter, we shed light on the challenges that were encountered and explain the new insights that emerged from the study. Finally, in the last part, some limitations of the study are considered.

6.1 Overcoming Limited Data Set Availability

Personality computing has emerged in response to the presence of digital footprints. Taking into account the convergence of pervasive and mobile technologies, which have contributed to the rapid growth in the numbers of Internet users, in the past few years personality computing researchers have focused mainly on traits inference from social media. In following that direction, the current study also employed the same type of data. Despite the fact that social media data is publicly accessible, obtaining a gold standard samples is not trivial. Like any other classification task of machine learning, a high quality data set is important for model development and testing; however, in regard to our study, there was a lack of available data sets that are specifically annotated with personality traits.

In the initial phases of this research, the aforementioned issue was addressed. We applied a data division technique to prevent an over-fitting from happening when a relatively small data set was used. We also performed transfer learning, a technique whereby information from one data set is used to inform a model on another. This technique has been reported to be effective in overcoming data scarcity while preserving the contextual differences in the underlying measurements (Hutchinson et al., 2017). Accordingly, the experimental results have shown that our system successfully trained the models while still enabling it to predict the personality traits from tweets. This evaluation is further discussed in the next section.

6.2 The Effectiveness of Proposed System

Based on the experimental results obtained, we can state that our proposed model has performed significantly better than the majority of baseline models for all five traits, although, different settings for different traits were implied. Our experiments showed that combining GloVe, CNN and LSTM into one unified framework which was jointly trained increased the overall perceived performance. The unified model achieved significant results up to 72.24% and 63.82%

Table 6.1: Comparison of accuracy for studies in personality recognition employing Neural Network Language Models

	OPE	CON	EXT	AGR	NEU	avgOCEAN
Published state-of-the-art						
Majumder et al. (2017)	62.68	57.30	58.09	56.71	59.38	58.832
Baseline models						
CNN	68.89	59.32	57.67	**57.99**	61.18	61.010
LSTM	69.08	57.09	**59.78**	57.43	59.54	60.584
Proposed model						
GloVe+CNN+LSTM	**72.24**	**59.38**	58.51	55.48	**63.82**	**61.862**

Note. OPE=Openness, CON=Conscientiousness, EXT=Extraversion, AGR=Agreeableness, NEU=Neuroticism, avgOCEAN=average accuracy. **Bold** highlights best performance.

accuracy for Openness (OPE) and Neuroticism (NEU) traits respectively. A good performance was also shown in predicting Conscientiousness (CON), Extraversion (EXT), Agreeableness (AGR) traits. For these traits we obtained 59.38%, 58.51%, and 55.48% accuracy, respectively. The results outperformed the current state-of-the-art model (Majumder et al., 2017), which strengthens the argument presented in this paper. This can be seen in Table 6.1.

Surprisingly, in the negative (NEG) category, the proposed model performed slightly worse than the baseline models (see Table 5.5). This was strongly influenced by an insufficient sample size, as the data set used comprised only 9,913 sentences with over 146,000 words. Nevertheless, we found that using the GloVe word embeddings has improved prediction performance. GloVe gives additional knowledge by capturing semantic similarity between words from the given short texts. This is showed in the positive (POS) category, where the unified model achieved an overall 58.73% accuracy, as can be seen in Table 5.3. A relative improvement after training with 10 epochs was also been observed in both NEG and nonpartisan (NON) sentiment groups as the model achieved 58.18% and 61.13% accuracy respectively. In general, then, we conclude that the smart system demonstrated satisfactory performance.

6.3 Amount of Text Required to Infer Traits

The previous section has shown that personality traits can be inferred with reasonable accuracy from self-authored texts from social media. This is inline with prior studies (Carducci et al., 2018; Celli et al., 2013; Farnadi et al., 2016) which revealed that the results obtained in this manner exhibited the same quality as the conventional method, i.e., a pencil-and-paper-based test. However, while the results are promising, the representativeness is still questionable.

Earlier in this thesis, we have mentioned that researchers often require a vast amount of data to train machine learning models in assessing human personality, for example, a collection of stream-of-consciousness essays (Majumder et al., 2017), or approximately 100,000 sample of messages posted online (Schwartz et al., 2013; Yarkoni, 2010). Applying this level of practice was considered unrealistic since, on average, Twitter users have only 22 tweets on their timeline (Burger et al. as cited in Arnoux et al., 2017). In that regard, therefore, the question is also raised as to *How much text is required to know a person?*.

It is reasonable to raise the issue of the minimum amount of text required for a representative result, as theoretically a high degree of granularity might allow researchers to get more information. In the light of this, Haber (2015) suggested that a total sample of 200 contiguous tweets is sufficient to achieve a stable measure in inferring trait. Additionally, Arnoux et al. (2017) revealed that using the same number of tweets to predict a user's personality has obtained significant results. Moreover, the authors highlighted that the performance using text as short as 25 tweets is on a par with the state-of-the-art model with 200 tweets. Nevertheless, our proposed system was designed to accommodate this difference.

In the current study, we attempted to infer personality from short texts. Such level of granularity comes down to a single tweet. The approach achieved an overall 61.862% accuracy, which outperformed that obtained with document-level model (Majumder et al., 2017). Furthermore, the flexibility of our system in processing different amount of texts was demonstrated in our case study-based investigation. Taking that example, we have used a sample of 200 contiguous tweets from an anonymised Twitter profile, and the system has been shown to be able to process each tweet and prediction the personality traits. The results were summarised and later aggregated to determine the user personality traits along with the sentiments conveyed. The following section will explore the evidence in more detail.

6.4 User Personality and Perceptions Correlation

In the introduction to this chapter, we noted that one of the main objectives of our research was to investigate the role of opinion polarity in trait inference based upon social media phenomena. Thus, to achieve this, we conducted a case-based investigation involving over 6,000 tweets from 300 Twitter profiles (anonymised) who expressed their sentiments about Uber, a ride-sharing service. In this study, we aimed not merely to process individual tweets, but also to understand the users who tweeted and the features they carried, which is sentiment polarity.

We found that Openness (OPE), Extraversion (EXT), and Agreeableness (AGR) scored high in the individuals who tweeted a positive review about Uber. In fact, the OPE score, which corresponds to receptivity to new ideas and approaches (Gross, 1996), was found to be higher in this group than others. This finding supports the evidence that a person with high OPE tends to express a positive perception towards the company. The fact that Uber is a sharing economy company and is categorised as new business model platform, also explains why users who belong to this group were high scorers in EXT. A person with the EXT trait, which was only found to be high in this group, has a tendency to be sociable, active, and willing to take risks.

In contrast to OPE trait, we observed that persons in all groups scored low in Conscientiousness (CON)–a trait that indicates an individual to be organised, dependable and motivated. We also identified that a person who stays neutral about Uber is accompanied by a high-level of the AGR trait. High scorers in this trait tend to obey rules and adopt the conventions of society (Gross, 1996). Although, it might be related, we cannot determine whether AGR sufficiently dominates users to the extent that it cause them to express their neutrality, as the trait also found to be continuously high in the other two groups. On the contrary, while Neuroticism (NEU) scored low in persons who showed positive and neutral feelings, the trait was revealed to be slightly higher in the group of users with negative views of Uber. The NEU trait indicates an emotional instability, thus, a high scorer is characterised as being moody and experiencing feelings such as anxiety, worry, fear, or anger (Gross, 1996); these are more likely related to negative sentiment. In summary, the results of this study support the previous findings in that those individuals with the same personality traits tend to make similar sentiment expressions (Lin et al., 2017; Stemmler & Wacker, 2010).

6.5 Limitations

Social media data is often publicly available; however, there were several aspects of the data that needed to be considered while doing this research. These included data control and privacy. Although these issues are much debated in the literature, the present study was designed with an awareness of those concerns. For example, while a personality test like the Big Five Inventory (BFI) is a fairly common approach for examining user personality, the current study does not imply nor present a comprehensive view of this assessment. Instead, our study relies on a crowd-sourced data set which was labelled with the five personality traits; although the trait scores might be seemingly different to the BFI scoring as we converted each score into a binary class for simplification purposes.

Furthermore, this study used data-driven machine learning techniques to extract activation patterns from training data. Hence, we acknowledge the importance of the representativeness of the data, which may account for the potentially biased results. Moreover, we have not analysed the results within a social theory framework. Such an approach could result in somewhat different interpretations. Therefore, the contribution of the current study in personality psychology might be limited.

The system has been tested under the macOS operating system. It was written in Python and deployed to a web service engine for use in demonstration. However, no usability tests have performed to evaluate the product by testing it on users. The software was assessed merely on the technical aspects in meeting the requirements. Finally, although we attempted to examine all available configurations and parameters, this was not always possible due to time constraints. Therefore, these further investigations will be performed in the future implementation of the software.

6.6 Summary

In this chapter, we highlighted the potential of and challenges faced during the research endeavour reported in this thesis. Several issues were raised in connection with the relevant findings reported. The first was the data quality issue. To overcome this problem, some techniques were considered such as employing data splitting and transfer learning. Next, we discussed the effectiveness of our system based on the experimental results and compared them with the current state-of-the-art. While we found that our proposed system performs well, a question was raised regarding the sufficient amount of text needed to infer traits. Some related studies thus were referred to support our analysis in this matter. Finally, the correlation between individual personality and perceptions was discussed. We noted that personality traits correspond to individual's perceptions. The chapter concluded by reflecting on the limitations of this study.

Chapter 7

Conclusions and Further Research

The conventional approach to measuring personality requires participants to answer a series of questions to evaluate their behaviours and preferences. This assessment process is tedious and labour-intensive. On the other hand, social media provides a vast amount of openly accessible social-related data that can be employed to infer a user's personality. In the real usage scenario, however, where on average users only have around twenty tweets on their timelines, this seems impractical. Also, while predicting user personality traits through text features on Twitter is promising, the character limit imposed on tweets makes the use of standard linguistic methods challenging and inefficient.

To address this problem, we developed a deep learning-based smart system for trait inference employing a Neural Network Language Model (NNLM). The system was designed to forecast a person's personality traits based on the way that person tweets. In addition, we also explored the sentiment information at the sentence level, building upon the assumption that personality traits correlate to users' sentiments. To capture that information, we ran lexicon-based sentiment classifiers utilising the Stanford Twitter Sentiment (STS) data sets. This was followed by grouping the outputs into three main categories, namely positive, negative, and nonpartisan. Lastly, in order to detect personality traits, a collection of 9,913 Facebook status updates which were labelled with sentiment polarity and the Big Five personality scores were used in training.

However, there were two main obstacles encountered in this approach. The first was data training quality, and the second was preserving information from short texts. To cope with that, a unified language model was defined combining Convolutional Neural Network (CNN) and the advantage of Long Short-Term Memory (LSTM) in maintaining information by adding past information to the present state. We applied the Global Vectors (GloVe) word embedding technique to add external knowledge by identifying similarities between words. Finally, we applied transfer learning by reusing the previously trained model to forecast traits using Twitter post as inputs. The result demonstrates the feasibility of inferring traits with reasonable accuracy from opinionated texts streamed online.

7.1 Achievements

The following achievements were reported in this thesis:

- **Sentiment-based Personality Detection**. A web-based smart system was developed based on an empirical neural network language modelling methodology and related deep-learning approaches. Sentiment-based Personality Detection, or SENTIPEDE, encompasses several functionalities in fulfilling the main objective: data acquisition, sentiment classification, and personality detection. We practised a Continuous Delivery approach—in which the system was reliably built, tested, and deployed—to deliver the proposed system to production. SENTIPEDE can be accessed through http://sentipede.dsrg.ac.nz

- **Case-based investigation on user personality and perceptions**. Motivated to investigate the existing correlation between user personality and perception, we conducted a case study-based investigation which employed the proposed system. The experiment revealed that personality traits correspond to the way persons express their perceptions towards a topic. This experiment has also proven the SENTIPEDE's applicability through the case study implementation. Based on our investigation, Openness and Agreeableness traits were observed to be consistently high in all group of users who tweeted positive, negative, and neutral reviews. In contrast, Conscientiousness scores were found to be low. We noted that high scorers in Extraversion tend to express positive review, while low scorers in Extraversion stayed neutral. And lastly, those having high Neuroticism scores are more likely to express negative views toward the brand.

- **An evaluation of the methods with different neural network architectures**. This work included some deep learning approaches under the umbrella of NNLM. We trained and applied configurations and some variations of the network architecture. An experimental study showed that the proposed model (GloVe+CNN+LSTM) outperformed the majority of baseline models. Our model achieved overall a good accuracy across the Big Five personality dimensions.

7.2 Further Work

In this thesis, we have introduced a smart system utilising some well-known deep learning techniques to achieve state-of-the-art computational personality recognition. The use of social media data makes this task somewhat easier; however, personality detection is a time-intensive and complex process. By adopting a software development model, our system was developed, tested and found to perform well, but not without room for enhancement. Therefore, we make some recommendations for future research as follows:

- We seek to expand our training data to better evaluate accuracy for various network architectures. Subsequently, we plan to involve participants to take personality assessments and use their Twitter posts as sample. Such an approach has been adopted by, for example, Carducci et al. (2018) and Qiu, Lin, Ramsay and Yang (2012), and thus will give us a point of comparison for our predictive models.
- We also aim to run a usability test on our web-based smart system. We expect users to take part in a survey while carrying out each task. The evaluation will provide us with valuable insights on user experience. The results of this assessment will allow us to improve the system's functionality.
- Finally, we mean to explore brand personality on social media using the proposed system. This future work is expected to set the stage for larger research projects such as investigate the relationship between brands' and customers' personalities.

References

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G. & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(10), 1533–1545. doi: 10.1109/TASLP.2014.2339736

Allen, J. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA: Benjamin-Cummings Publishing.

Allison, B., Guthrie, D. & Guthrie, L. (2006). Another look at the data sparsity problem. In P. Sojka, I. Kopeček & K. Pala (Eds.), *Text, speech and dialogue* (pp. 327–334). Berlin, Germany: Springer.

Al-Sheikh, E. S. & Hasanat, M. H. A. (2018). Social media mining for assessing brand popularity. *International Journal of Data Warehousing and Mining (IJDWM)*, *14*(1), 40–59. doi: 10.4018/IJDWM.2018010103

Arnoux, P., Xu, A., Boyette, N., Mahmud, J., Akkiraju, R. & Sinha, V. (2017). 25 tweets to know you: A new model to predict personality with social media. *Computing Research Repository (CoRR)*, *abs/1704.05513*.

Barbier, G. & Liu, H. (2011). Data mining in social media. In C. C. Agrawal (Ed.), *Social network data analytics* (pp. 327–352). New York, NY: Springer.

Bermingham, A. & Smeaton, A. (2011). On using twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)* (pp. 2–10). Chiang Mai, Thailand: Asian Federation of Natural Language Processing. Retrieved from `https://www.aclweb.org/anthology/W11-3702`

Bikel, D. & Zitouni, I. (2012). *Multilingual natural language processing applications: From theory to practice*. Upper Saddle River, NJ: Pearson Education.

Bonzanini, M. (2016). *Mastering social media mining with Python*. Birmingham, UK: Packt Publishing.

Buchanan, G. & Loizides, F. (2007). Investigating document triage on paper and electronic media. In L. Kovács, N. Fuhr & C. Meghini (Eds.), *Research and advanced technology for digital libraries* (pp. 416–427). Berlin, Germany: Springer.

Burton, L., Westen, D. & Kowalski, R. (2014). *Psychology*. Hoboken, NJ: Wiley.

Carducci, G., Rizzo, G., Monti, D., Palumbo, E. & Morisio, M. (2018, May). Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning. *Information*, *9*(5), 1–20. doi: 10

.3390/info9050127

Celli, F. (2011, September). Mining user personality in Twitter. *Language, Interaction and Computation Laboratory (CLIC)*. Retrieved from http//clic.cimec.unitn.it

Celli, F., Pianesi, F., Stillwell, D. & Kosinski, M. (2013). Workshop on computational personality recognition: Shared task. In *Seventh international aaai conference on weblogs and social media.* Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewPaper/6190

Chamorro-Premuzic, T. & Furnham, A. (2003). Personality traits and academic examination performance. *European Journal of Personality*, *17*(3), 237–250. doi: 10.1002/per.473

Cireşan, D. C., Meier, U., Gambardella, L. M. & Schmidhuber, J. (2011). Convolutional neural network committees for handwritten character classification. In *International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1135–1139). doi: 10.1109/ICDAR.2011.229

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Computing Research Repository (CoRR)*, *abs/1103.0398*.

Collomb, A., Brunie, L. & Costea, C. (2013). *A study and comparison of sentiment analysis methods for reputation evaluation.*

Conneau, A., Schwenk, H., Barrault, L. & LeCun, Y. (2016). Very deep convolutional networks for natural language processing. *Computing Research Repository (CoRR)*, *abs/1606.01781*.

Cresci, E. (2017, January 30). #DeleteUber: How social media turned on Uber. *The Guardian*. Retrieved March 1, 2019, from https://www.theguardian.com/technology/2017/jan/30/deleteuber-how-social-media-turned-on-uber

Csaba, R. & Reiner, R. (2016, October). The sharing economy: Economic frame or forerunner of another political cleavage? *4Liberty Review*, 18–33. Retrieved from http://4liberty.eu/4liberty-eu-review-5-is-already-available-online-and-for-download/

Darliansyah, A., Wandabwa, H. M., Naeem, M. A., Mirza, F. & Pears, R. (2019). Long-term trends in public sentiment in Indian demonetisation policy. In I. S. Bajwa, F. Kamareddine & A. Costa (Eds.), *International Conference on Intelligent Technologies and Applications (INTAP)* (pp. 65–75). doi: 10.1007/978-981-13-6052-7_6

Dave, K., Lawrence, S. & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International Conference on World Wide Web* (pp. 519–528). doi: 10.1145/775152.775226

Daya, S., Duy, N. V., Eati, K., Ferreira, C. M., Glozic, D., Gucer, V., . . . Vennam, R. (2015). *Microservices from theory to practice: Creating applications in IBM bluemix using the microservices approach*. Poughkeepsie, NY: IBM Redbooks. Retrieved from http://www.redbooks.ibm.com/abstracts/

sg248275.html

de Raad, B. (2000). *The big five personality factors: The psycholexical approach to personality*. Kirkland, WA: Hogrefe & Huber Publishers.

Diaz, F., Mitra, B. & Craswell, N. (2016). Query expansion with locally-trained word embeddings. *Computing Research Repository (CoRR), abs/1605.07891*.

dos Santos, C. & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the Twenty-fifth International Conference on Computational Linguistics (COLING): Technical Papers* (pp. 69–78). Retrieved from https://www.aclweb.org/anthology/C14-1008

Elsner, D. (2018, February). *Deploying a simple machine learning model in a modern web application.* Retrieved June 1, 2018, from https://medium.com/@dvelsner/

Esuli, A. & Sebastiani, F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC)* (pp. 417–422). doi: 10.1.1.61.7217

Farnadi, G., Sitaraman, G., Sushmita, S., Celli, F., Kosinski, M., Stillwell, D., ... De Cock, M. (2016). Computational personality recognition in social media. *User Modeling and User-Adapted Interaction*, *26*(2), 109–142. doi: 10.1007/s11257-016-9171-0

Fuchs, C. (2013). *Social media: A critical introduction*. London, UK: SAGE.

Gerber, A. S., Huber, G. A., Doherty, D. & Dowling, C. M. (2011). The big five personality traits in the political arena. *Annual Review of Political Science*, *14*(1), 265–287.

Giachanou, A. & Crestani, F. (2016). Like it or not: A survey of Twitter sentiment analysis methods. *ACM Computing Surveys*, *49*(2), 1–41. doi: 10.1145/2938640

Golbeck, J., Robles, C., Edmondson, M. & Turner, K. (2011, October). Predicting personality from Twitter. In *IEEE Third International Conference on Privacy, Security, Risk and Trust and IEEE Third International Conference on Social Computing* (pp. 149–156). Boston, MA: IEEE.

Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American Psychologist*, *48*(1), 26–34. doi: 10.1037/0003-066X.48.1.26

Golhamer, H. (1950). Public opinion and personality. *American Journal of Sociology*, *55*(4), 346–354. Retrieved from https://www.jstor.org/stable/2772295

Granville, K. (2018, March 19). Facebook and Cambridge Analytica: What you need to know as fallout widens. *The New York Times*. Retrieved March 1, 2019, from https://www.nytimes.com/2018/03/19/technology/facebook-cambridge-analytica-explained.html

Gravelle, T. B., Reifler, J. & Scotto, T. J. (2014). *Personality traits and foreign policy attitudes: A comparative examination of the United States and* *the United Kingdom.* Paper presented at the Annual Conference of the International Studies Association, Toronto, Canada.

Greenfield, P. (2018, March 26). The Cambridge Analytica files: The story so far. *The Guardian*. Retrieved March 1, 2019, from https://www.theguardian.com/news/2018/mar/26/the-cambridge-analytica-files-the-story-so-far

Gross, R. D. (1996). *Psychology: The science of mind and behaviour.* London, UK: Hodder & Stoughton.

Güngör, T. (2010). Part-of-speech tagging. In N. Indurkhya & F. J. Damerau (Eds.), *Handbook of natural language processing* (2nd ed., pp. 205–235). Boca Raton, FL: Chapman and Hall/CRC.

Haber, E. M. (2015). On the stability of online language features: How much text do you need to know a person? *Computing Research Repository (CoRR)*, *abs/1504.06391*.

Hahn, S. & Choi, H. (2018). Gradient acceleration in activation functions. *Computing Research Repository (CoRR)*, *abs/1806.09783*.

Hayashi, T. & Fujita, H. (2018). Sentence-level sentiment analysis using feature vectors from word embeddings. In H. Fujita & E. Herrera-Viedma (Eds.), *New trends in intelligent software methodologies, tools and techniques.* Amsterdam, Netherlands: IOS Press.

Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Hochreiter, S. & Schmidhuber, J. (1997, December). Long short-term memory. *Neural Computation*, *9*, 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hopfgartner, F. (2015). *Smart information systems: Computational intelligence for real-life applications*. Cham, Switzerland: Springer International Publishing.

Hutchinson, M. L., Antono, E., Gibbons, B. M., Paradiso, S., Ling, J. & Meredig, B. (2017). Overcoming data scarcity with transfer learning. *Computing Research Repository (CoRR)*, *abs/1711.05099*.

Hutto, C. & Gilbert, E. (2015, January). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media* (pp. 216–225). Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109

Iqbal, M. (2019). *Uber revenue and usage statistics 2018.* Retrieved March 1, 2019, from http://www.businessofapps.com/data/uber-statistics/

John, O. P. & Srivastava, S. (1999). The big-five trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research* (Vol. 2, pp. 102–138). New York, NY: Guilford Press.

Joyce, B. & Deng, J. (2017, November). Sentiment analysis of tweets for the 2016 US presidential election. In *IEEE MIT Undergraduate Research Technology Conference (URTC)* (pp. 1–4). doi: 10.1109/URTC.2017.8284176

Judge, T., Thoresen, C., Bono, J. & Patton, G. (2001, January). The job satisfaction-job performance relationship: A qualitative and quantitative review. *Psychological Bulletin*, *127*, 376–407. doi: 10.1037//0033-2909.127.3.376

Jurafsky, D. & Martin, J. H. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Kalchbrenner, N., Danihelka, I. & Graves, A. (2015). Grid long short-term memory. *Computing Research Repository (CoRR)*, *abs/1507.01526*.

Kalghatgi, M. P., Ramannavar, M. & Sidnal, N. S. (2015). A neural network approach to personality prediction based on the big-five model. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, *2*(8), 56–63. Retrieved from http://www.ijirae.com/volume-2-issue-08

Kasprowicz, T. (2016, October). The emergence and development of a sharing economy. *4Liberty Review*, 4–17. Retrieved from http://4liberty.eu/4liberty-eu-review-5-is-already-available-online-and-for-download/

Kemp, S. (2018). *Global digital report 2018.* Retrieved July 1, 2018, from https://digitalreport.wearesocial.com

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751). doi: 10.3115/v1/D14-1181

Kirchhoff, K. (2012). Language modeling. In D. Bikel & I. Zitouni (Eds.), *Multilingual natural language processing applications* (pp. 169–198). Upper Saddle River, NJ: IBM Press.

Kolchyna, O., Souza, T. T. P., Treleaven, P. C. & Aste, T. (2015). Twitter sentiment analysis. *Computing Research Repository (CoRR)*, *abs/1507.00955*.

Kosinski, M., Stillwell, D. & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. In *Proceedings of the National Academy of Sciences of the United States of America* (Vol. 110, pp. 5802–5805). doi: 10.1073/pnas.1218772110

Kruse, L. M., Norris, D. R. & Flinchum, J. R. (2018). Social media as a public sphere? Politics on social media. *Sociological Quarterly*, *59*(1), 62–84. doi: 10.1080/00380253.2017.1383143

Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, *8*(1), 98–113. doi: 10.1109/72.554195

LeCun, Y. & Bengio, Y. (1998). Convolutional networks for images, speech, and time series. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 255–258). Cambridge, MA: MIT Press.

Lee, R. M., Fielding, N. & Blank, G. (2017). The Internet as a research medium. In N. Fielding, R. M. Lee & G. Blank (Eds.), *The SAGE handbook of online research methods.* London, UK: SAGE Publications Inc.

Lima, A. C. E. & de Castro, L. N. (2014). A multi-label, semi-supervised classification approach applied to personality prediction in social media. *Neural Networks*, *58*, 122–130. doi: 10.1016/j.neunet.2014.05.020

Lin, J., Mao, W. & Zeng, D. D. (2017). Personality-based refinement for sentiment classification in microblog. *Knowledge-Based System*, *132*, 204–214. doi: 10.1016/j.knosys.2017.06.031

Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents, and usage data*. Berlin, Germany: Springer.

Liu, F., Perez, J. & Nowson, S. (2016). A language-independent and compositional model for personality trait recognition from short texts. *Computing Research Repository (CoRR)*, *abs/1610.04345*.

Lomas, N. (2018, October 15). Saudi ally calls for Uber boycott over response to Khashoggi's vanishing. *TechCrunch*. Retrieved March 1, 2019, from `https://techcrunch.com/2018/10/15/saudi-ally-calls-for-uber-boycott-over-response-to-khashoggis-vanishing/`

Mairesse, F., Walker, M. A., Mehl, M. R. & Moore, R. K. (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, *30*(1), 457–500. Retrieved from `http://dl.acm.org/citation.cfm?id=1622637.1622649`

Majumder, N., Poria, S., Gelbukh, A. & Cambria, E. (2017). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, *32*(2), 74–79. doi: 10.1109/MIS.2017.23

Mandelbaum, A. & Shalev, A. (2016). Word embeddings and their use in sentence classification tasks. *Computing Research Repository (CoRR)*, *abs/1610.08229*.

Manning, C. & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

Matthews, P. H. (2007). *The concise Oxford dictionary of linguistics* (2nd ed.). New York, NY: Oxford University Press.

Mayfield, J. & McNamee, P. (2003, July). Single n-gram stemming. In *Proceedings of the Twenty-sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 415–416). doi: 10.1145/860435.860528

McCrae, R. R. & John, O. P. (1992, July). An introduction to the five-factor model and its applications. *Journal of Personality*, *60*, 175–215. doi: 10.1111/j.1467-6494.1992.tb00970.x

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Computing Research Repository (CoRR)*, *abs/1310.4546*.

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill.

Mohorovičić, S. (2013). Implementing responsive web design for enhanced web presence. In *The Thirty-sixth International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1206–1210). Retrieved from `https://ieeexplore.ieee.org/document/6596440`

Nam, H., Joshi, Y. V. & Kannan, P. K. (2017). Harvesting brand information from social tags. *Journal of Marketing*, *81*(4), 88–108. doi: 10.1509/jm.16.0044

Neethu, M. S. & Rajasree, R. (2013, July). Sentiment analysis in Twitter using machine learning techniques. In *The Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (pp. 1–5). doi: 10.1109/ICCCNT.2013.6726818

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *Computing Research Repository (CoRR)*, *abs/1103.2903*.

Oates, B. (2006). *Researching information systems and computing*. London, UK: SAGE Publications.

Palling, K. (2016, October). Ride-sharing as the ultimate sustainable alternative to a traditional economy. *4Liberty Review*, 18–33. Retrieved from http://4liberty.eu/4liberty-eu-review-5-is-already-available-online-and-for-download/

Palmer, D. D. (2010). Text preprocessing. In N. Indurkhya & F. J. Damerau (Eds.), *Handbook of natural language processing* (2nd ed., pp. 9–30). Boca Raton, FL: Chapman and Hall/CRC.

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*(1-2), 1–135. doi: 10.1561/1500000011

Parkhi, O. M., Vedaldi, A. & Zisserman, A. (2015, January). Deep face recognition. *British Machine Vision Conference (BMVC)*, *1*, 1–12. doi: 10.5244/C.29.41

Pennebaker, J. & King, L. (1999, December). Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, *77*(6), 1296–1312. doi: 10.1037/0022-3514.77.6.1296

Pennington, J., Socher, R. & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Retrieved from https://www.aclweb.org/anthology/D14-1162

Qiu, L., Lin, H., Ramsay, J. & Yang, F. (2012, December). You are what you tweet: Personality expression and perception on Twitter. *Journal of Research in Personality*, *46*, 710–718. doi: 10.1016/j.jrp.2012.08.008

Sahu, S. & Anand, A. (2015). Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of BioNLP* (pp. 158–163).

Sanchez Rodriguez, D., Mendez, O. & Florez, H. (2018, July). An approach of a framework to create web applications. In O. Gervasi et al. (Eds.), *Computational science and its applications – ICCSA 2018* (pp. 341–352). doi: 10.1007/978-3-319-95171-3_27

Schoen, H. (2007). Personality traits and foreign policy attitudes in German public opinion. *Journal of Conflict Resolution*, *51*(3), 408-430. doi: 10.1177/0022002707300180

Schrammel, J., Hochleitner, C. & Tscheligi, M. (2009, September). Personality traits, usage patterns and information disclosure in online communities. In *Proceedings of the British Computer Society Conference on Human-Computer Interaction (BCS-HCI)* (pp. 169–174). doi: 10.1145/1671011.1671031

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., . . . Ungar, L. H. (2013, September). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLOS ONE*, *8*(9), 1–16. doi: 10.1371/journal.pone.0073791

Shi, D. (2017). A study on neural network language modeling. *Computing Research Repository (CoRR)*, *abs/1708.07252*.

Silvis-Cividjian, N. (2017). *Pervasive computing: Engineering smart systems*. Cham, Switzerland: Springer International Publishing.

Smith, K. A. (1999). *Introduction to neural networks and data mining for business applications*. Melbourne, Australia: Eruditions Publishing.

Steele, G. L., Jr. (1999). Growing a language. *Higher-Order & Symbolic Computation*, *12*(3), 221–236. doi: 10.1023/A:1010085415024

Stemmler, G. & Wacker, J. (2010, October). Personality, emotion, and individual differences in physiological responses. *Biological Psychology*, *84*, 541–551. doi: 10.1016/j.biopsycho.2009.09.012

Sumner, C., Byers, A., Boochever, R. & Park, G. J. (2012, December). Predicting dark triad personality traits from Twitter usage and a linguistic analysis of tweets. In *Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA)* (Vol. 2, pp. 386–393). doi: 10.1109/ICMLA.2012.218

Syrdal, H. A. & Briggs, E. (2018). Engagement with social media content: A qualitative exploration. *Journal of Marketing Theory and Practice*, *26*(1-2), 4–22. doi: 10.1080/10696679.2017.1389243

Tausczik, Y. R. & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54. doi: 10.1177/0261927X09351676

Tripathy, A., Agrawal, A. & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126. doi: 10.1016/j.eswa.2016.03.028

van Dijck, J. (2013). *The culture of connectivity: A critical history of social media*. New York, NY: Oxford University Press.

Vinik, D. (2014). Uber's prices surged in Sydney during the hostage crisis, and everyone is furious. *The New Republic*. Retrieved March 1, 2019, from `https://newrepublic.com/article/120564/during-terrorist-attack-sydney-uber-imposing-surge-pricing`

Voss, A., Lvov, I. & Thomson, S. D. (2017). Data storage, curation and preservation. In L. Sloan & A. Quan-Haase (Eds.), *The SAGE handbook of social media research methods* (pp. 161–176). London, UK: SAGE Publications.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. & Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, *174*, 806–814. doi: 10.1016/j.neucom.2015.09.096

Wittig, M., Wittig, A. & Whaley, B. (2018). *Amazon web services in action*. Shelter Island, NY: Manning.

Yang, H. C., Lin, C. S., Huang, Z. R. & Tsai, T. H. (2017). Text mining on player personality for game recommendation. In *Proceedings of the Fourth Multidisciplinary International Social Networks Conference* (pp. 1–6). doi: 10.1145/3092090.3092132

Yarkoni, T. (2010, June). Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of Research in Personality*, *44*, 363–373. doi: 10.1016/j.jrp.2010.04.001

Yi, J., Nasukawa, T., Bunescu, R. & Niblack, W. (2003, November). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *IEEE Third International Conference on Data Mining* (pp. 427–434). Melbourne, FL. doi: 10.1109/ICDM.2003.1250949

Zafarani, R., Abbasi, M. A. & Liu, H. (2014). *Social media mining: An introduction*. New York, NY: Cambridge University Press.

Appendix A

Example Code Snippet

Preparing the Embedding Layer

```
import numpy as np
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.models import model_from_json
from keras.layers import Dense, Flatten, Conv1D, MaxPooling1D,
    LSTM, Dropout, Activation
from keras.layers.embeddings import Embedding

EMBEDDING_FILE = 'glove.twitter.27B.200d.txt'
EMBEDDING_DIM = 200
MAX_NB_WORDS = 20000
MAX_SEQUENCE_LENGTH = 1000
FILENAME = 'mypersonality.csv'
NAMES = ['TEXT','TRAIT']

df = pd.read_csv(FILENAME, names = NAMES,encoding = 'utf-8')

embeddings_index = {}
f = open(EMBEDDING_FILE)
count = 0
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype = 'float32')
    embeddings_index[word] = coefs
f.close()

tokenizer = Tokenizer(num_words = MAX_NB_WORDS)
tokenizer.fit_on_texts(df['TEXT'])
word_index = tokenizer.word_index
```

```
sequences = tokenizer.texts_to_sequences(df['TEXT'])
data = pad_sequences(sequences, maxlen = MAX_SEQUENCE_LENGTH)

nb_words = min(MAX_NB_WORDS, len(word_index)) + 1
embedding_matrix = np.zeros((nb_words, EMBEDDING_DIM))

for word, index in tokenizer.word_index.items():
    if index > nb_words - 1:
        break
    else:
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
            embedding_matrix[index] = embedding_vector
```

Building the Convolution Neural Network

```
VALIDATION_SPLIT = 0.2

indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])

x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]

model = Sequential()
model.add(Embedding(len(word_index) + 1, EMBEDDING_DIM,
    input_length=MAX_SEQUENCE_LENGTH, weights=[embedding_matrix],
    trainable=False))
model.add(Dropout(0.2))
model.add(Conv1D(64, 5, activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
    metrics=['accuracy'])
```

Adding the LSTM Layer

```
model = Sequential()
model.add(Embedding(nb_words, EMBEDDING_DIM, input_length =
    MAX_SEQUENCE_LENGTH, weights = [embedding_matrix], trainable=
    False))
model.add(Dropout(0.2))
model.add(Conv1D(64, 5, activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
```

```
model.compile(loss='binary_crossentropy', optimizer='adam',
    metrics=['accuracy'])
model.summary()

model.fit(x_train, y_train, validation_data=(x_val, y_val),
    epochs=10, verbose=2)
score = model.evaluate(x_val, y_val, verbose=0)

model_json = model.to_json()
```

Appendix B

Sample Data

MyPersonality Corpus

Status	cEXT	cNEU	cAGR	cCON	cOPN
likes the sound of thunder.	n	y	n	n	y
is so sleepy it's not even funny that's she can't get to sleep.	n	y	n	n	y
Back from vacation and tired	y	y	n	y	y
just watched an episode of locked up abroad... dont take your freedom for granted people... its a blessing...	y	y	n	y	y
"I find it absolutely appalling that anyone could believe that political affiliation would be used to determine who receives medical treatment. That's what race and socioeconomic standing are for."	n	y	n	n	y
In other news, the Steelers 4th quarter is the most depressing thing of all time.	n	y	n	n	y
Four day camping trip!!! Holy Crap! this is going to be fun...	n	n	y	y	n
things just got really complicated	n	n	y	y	y
is trying to figure out the tornado winds outside right now. This is LA people. Let us not forget.	n	n	n	n	y
is really just trying to get through this last hard week of work before winter vacation.	n	n	n	n	y
just went to a bar with my mom and a bunch of other old people in Alaska. wtf?	n	y	n	y	n
now that i've been released from the clutches of retail it's time for some friends and family. merry xmas eve peeps	n	y	n	n	y

Tweets with Traits and Sentiment Labelled

Tweets	OPN	CON	EXT	AGR	NEU	Polarity
best I ever had Uber. Hope it was good for you too	0.962	0.906	0.978	0.092	0.006	Positive
Our Uber driver told me I was too cute to be a 3rd wheel and my confidence is so boosted wow	1	0.952	0.981	0.939	0	Positive
Me saying goodbye to my super nice uber driver: I hope you have a wonderful life and I hope all of your dreams come true!!!!!! Him: Okay this is your apartment	0.754	0.174	0.945	0.999	0.003	Positive
Why do I feel like that Uber driver cared about me more than my friends lol	0.983	0.897	0.998	0.948	0	Positive
@uber just passed a green levy (clean air fee) onto the customer putting a CC levy on ech ride would be disaster for them	0.985	0.0053	0.979	0.020	0.578	Negative
you ever be so bored in a uber and have nothing to do on your phone that you just watch your own trip?	0.504	0.718	0.145	0.993	0.944	Negative
UBER Where your safety is apparently meaningless drivers in stolen vehicles and drivers rated at 4.9 who don't even possess a driving license. What happened to the strict temporary license	0.006	0.361	0.973	0.844	0.938	Negative
I am so disheartened. I had a little extra $ so I got myself a new phone but I guess I just dont deserve anything nice. really wish @uber @uber_support do something	0.14	0.292	0.981	0.84	0.989	Negative
Bankruptcy beckons Uber's self-driving car unit was burning $20 million a month	0.389	0.449	0.18	0.023	0.01	Neutral
A licensed car hired by a driver that uses an app to pick up passengers. #uber #spain	0.759	0.528	0.206	0.061	0.06	Neutral
@username Connecting people. That's what the Uber brand is all about	0.864	0.46	0.115	0.708	0.501	Neutral
Working on the Uber app in London. Seen working at Kings Cross and Waterloo	0.913	0.887	0.14	0.333	0.019	Neutral