Exploring embedding vectors for emotion detection

Mohammed Alshahrani

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

October 2020

Abstract

Textual data nowadays is being generated in vast volumes. With the proliferation of social media and the prevalence of smartphones, short texts have become a prevalent form of information such as news headlines, tweets and text advertisements. Given the huge volume of short texts available, effective and efficient models to detect the emotions from short texts become highly desirable and in some cases fundamental to a range of applications that require emotion understanding of textual content, such as human computer interaction, marketing, e-learning and health.

Emotion detection from text has been an important task in Natural Language Processing (NLP) for many years. Many approaches have been based on the emotional words or lexicons in order to detect emotions. While the word embedding vectors like Word2Vec have been successfully employed in many NLP approaches, the word mover's distance (WMD) is a method introduced recently to calculate the distance between two documents based on the embedded words. This thesis is investigating the ability to detect or classify emotions in sentences using word vectorization and distance measures. Our results confirm the novelty of using Word2Vec and WMD in predicting the emotions in short text.

We propose a new methodology based on identifying "idealised" vectors that capture the essence of an emotion; we define these vectors as having the minimal distance (using some metric function) between a vector and the embeddings of the text that contains the relevant emotion (e.g. a tweet, a sentence). We look for these vectors through searching the space of word embeddings using the covariance matrix adaptation evolution strategy (CMA-ES). Our method produces state of the art results,

surpassing classic supervised learning methods.

Dedication

This thesis is dedicated to

My Parents, My siblings, My wife and My children

for their love, encouragement and support during this past 4 years

Acknowledgements

My grateful and sincere thanks goes to my supervisors Professor Maria Fasli and Dr Spyros Samothrakis for their help, guidance, motivation and usual support during this period of study. I really appreciate their help, guidance, valuable comments, feedback and time throughout the time of this study.

My hearty thanks filled with gratitude to my parents for their love, kindness, care and support. Also my deepest thanks to my siblings for their support and encouragement. To all of you my parents, brothers and sisters I am so sorry for missing many occasions and being away when you needed me.

I am also grateful to all my colleagues and friends. I apologise, I'm not going to mention anyone in particular. There have been many of you over the years, and you have all been great colleagues and good friends.

I would like to take the opportunity to express my gratefully thanks to my lovely wife (Zainab) who have been besides me all the time.

My deepest emotions are for my children Ali and Hazim, I know that I was busy with my study in the last few years. I have now finished my PhD thesis, and I will spend more time with you.

Contents

1 Introduction **14**

1.1 Overview 14

1.2 Understanding Emotions 15

1.3 Research Aims and Objectives 16

1.4 Contributions 18

1.5 Thesis structure 19

1.6 Publications 21

2 Literature review **22**

2.1 Introduction 22

2.2 Formal and Informal Text 23

2.3 Sentiment Analysis 23

2.4 Emotion Frameworks 25

2.4.1 Discrete Frameworks 25

2.4.2 Dimensional Framework 26

2.5 Emotion Detection 28

2.5.1 Unsupervised Learning 28

2.5.2 Supervised Learning 33

2.6 multi-classification 40

2.7 Emotion Related Lexicons 41

2.8 Word Embedding 42

2.8.1 Continuous Bag-of-Words Model 43

2.8.2 Continuous Skip-gram Model . . . 43
2.8.3 Word2Vec Embedding . . . 44
2.8.4 GloVe Embedding . . . 44
2.8.5 ELMo . . . 45
2.8.6 BERT . . . 46
2.8.7 XLNet . . . 46
2.9 Word Mover's Distance . . . 46
2.10 Evolution Strategies . . . 48
2.10.1 Brief history . . . 49
2.10.2 (1+1)-ES . . . 50
2.10.3 CMA-ES . . . 51
2.10.4 NES . . . 52
2.10.5 Separable NES (SNES) . . . 53
2.11 Summary . . . 54

3 Emotion detection using WMD 56

3.1 Introduction . . . 56
3.2 Dataset . . . 58
3.3 WMD-ED approach . . . 59
3.4 Experiments . . . 59
3.4.1 Word Net Affect experiment . . . 59
3.4.2 WMD-ED experiment 1 . . . 61
3.4.3 WMD-ED experiment 2 . . . 61
3.4.4 WMD-ED Experiments 3a and 3b . . . 62
3.4.5 Bootstrapping . . . 64
3.5 results . . . 64
3.5.1 Word Net Affect experiment . . . 64
3.5.2 WMD-ED Experiment 1 . . . 65
3.5.3 WMD-ED Experiment 2 . . . 66
3.5.4 WMD-ED Experiments 3a and 3b . . . 70

3.5.5 bootstrapping . . . 77
3.5.6 Surrounding seed words . . . 77
3.6 Discussion . . . 79
3.7 Summary . . . 80

4 ES for emotion detection 82
4.1 Introduction . . . 82
4.2 Datasets . . . 84
4.3 Benchmark . . . 87
4.4 Methodologies . . . 88
4.4.1 WMD-ED . . . 88
4.4.2 CMA-ES . . . 89
4.5 CMA-ES Experiments . . . 89
4.5.1 WMD-ED Experiment 1 . . . 90
4.5.2 WMD-ED Experiment 2 . . . 90
4.6 ES Search Exp . . . 91
4.6.1 SNES . . . 91
4.6.2 CMA-ES . . . 93
4.7 Results . . . 95
4.7.1 WMD-ED Experiment 1 . . . 95
4.7.2 WMD-ED Experiment 2 . . . 95
4.7.3 ES Search Exp . . . 96
4.8 Different Datasets . . . 109
4.8.1 ISEAR Dataset . . . 109
4.8.2 EmoContext Dataset . . . 111
4.9 Different embedding . . . 112
4.9.1 Exploring domain specificity using GloVe . . . 112
4.9.2 Emotion Embedding . . . 114
4.10 Discussion . . . 115
4.11 Summary . . . 116

5 Conclusions **119**

5.1 Summary and Contributions 119

5.2 Future Work 121

List of Figures

2.1 Country and Capital Vectors in Word2Vec Projected by two dimensional PCA (Image taken from [66] page 5). 45
2.2 Illustration of WMD between documents while (top) with equal BOW (Bottom) with different count of words (Image taken from [57] page 3). 47

3.1 An example of WMD distances calculated directly for the six different emotions - the algorithm relates the sentence to “fear” - the word with the shortest distance from the sentence. 60
3.2 PCA of WMD-ED Experiment 2 seed words) 68
3.3 WMD-ED Experiment 3a for anger - with basic word “anger” included in the seed words 71
3.4 WMD-ED Experiment 3a for disgust - with basic word “disgust” included in the seed words 71
3.5 WMD-ED Experiment 3a for fear - with basic word “fear” included in the seed words 72
3.6 WMD-ED Experiment 3a for joy - with basic word “joy” included in the seed words 72
3.7 WMD-ED Experiment 3a for sadness - with basic word “sadness” included in the seed words 73
3.8 WMD-ED Experiment 3a for surprise - with basic word “surprise” included in the seed words 73
3.9 WMD-ED Experiment 3b - basic word “anger” is deliberately not included in the seed words 74

3.10 WMD-ED Experiment 3b - basic word "disgust" is not deliberately included in the seed words 74
3.11 WMD-ED Experiment 3b - basic word "fear" is not deliberately included in the seed words 75
3.12 WMD-ED Experiment 3b - basic word "joy" is not deliberately included in the seed words 75
3.13 WMD-ED Experiment 3b - basic word "sadness" is deliberately not included in the seed words 76
3.14 WMD-ED Experiment 3b - basic word, "surprise" is deliberately not included in the seed words 76
3.15 The python code used to obtain the most similar words 78
3.16 The PCA of the surrounding seed words while separated 79

4.1 Word-cloud of TEC corpus including annotation (hash-tag) words like joy, sadness and surprise. 86
4.2 Word-cloud of TEC corpus after removing the annotation words, like joy and sadness (so that corpus can be used for training and testing). . 87
4.3 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for anger 97
4.4 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for disgust 98
4.5 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for fear 98
4.6 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for joy 99
4.7 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for sadness 99
4.8 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for surprise 100

List of Tables

2.1 Basic emotions of discrete frameworks 27

3.1 The emotion seed words that achieved the best results 63

3.2 Performance of the Word Net Affect experiments 65

3.3 Overall average results 65

3.4 WMD-ED Experiment 1 results - including correct and incorrect classification 66

3.5 WMD-ED Experiment 2 overall F1 average results 67

3.6 WMD-ED Experiment 2 F1 score results for each system for every emotion category 68

3.7 WMD-ED Experiment 2 F1 score results compared with recent results from [16] 69

3.8 Results obtained using the bootstrapping 77

3.9 The emotion seed words which achieved the best results 77

3.10 Results obtained using the surrounding seed words 79

4.1 The emotion "idealised" words that achieved the best results for Experiment 1 91

4.2 The emotion "idealised" words that achieved the best results for Experiment 2 91

4.3 Experiments results 95

4.4 Most similar words to each vector from cross-validation for each emotion and most similar word to the mean 101

4.11 The F1 score of the results yielded by calculating the WMD distance between the mean vectors and the tweets 102
4.5 The most similar words to the emotions anger and not-anger, the latter being all other emotions 103
4.6 The most similar words to the emotions Disgust and not-Disgust, the latter being all other emotions 104
4.7 The most similar words to the emotions Fear and not-Fear, the latter being all other emotions 105
4.8 The most similar words to the emotion Joy and not-Joy, the latter being all other emotions 106
4.9 The most similar words to the emotion Sadness and not-Sadness which is the position of other emotions 107
4.10 The most similar words to the emotions Surprise and not-Surprise, the latter being all other emotions 108
4.12 ISEAR experiments results 110
4.13 EmoContext experiments results 112
4.14 The emotion seed words that achieved the best results in the course of the GloVe experiment 113
4.15 GloVe experiment's results 113
4.16 The emotion seed words that provided the best results in the Emotion-Word2Vec experiment. 115
4.18 Emotion Wor2Vec Experiment's results 115
4.17 Emotion embedding - most similar words for each category 116

Chapter 1

Introduction

1.1 Overview

Written text is a powerful tool which is not only used to communicate content in terms of data and information but it can carry emotional expression as well. It is clear that textual data such as stories, poems and novels carry with them a certain emotional content. However, the generated textual data, nowadays, is tremendous especially through the use of new technologies such as social media and the prevalence of smartphones. Of course, people use social media to post and express their emotions and feelings instantly via smartphones. Since emotions are involved in this huge volume of textual data, wherefore being able to automatically detect and identify emotions has become highly desirable and in some cases a requirement [112].

Emotions may be articulated by written text which is known as text based emotion. Nowadays, writings take many forms of social media posts, tweets, micro-blogs, news articles, etc. The content of these posts can be useful resource for emotion detection. Different types of dataset such as new headlines, tweets and micro-blogs are conceptually different and they might convey or contain emotions in different ways. For example, news headlines are written by experts to attract the reader. While, in tweets emotional state may be included as a hashtag or emoticons. Not to mention the fact that dataset like news headlines are written in a formal way while datasets like tweets are written in informal way which makes it more complicated.

Most works that have been presented in the literature and are described in this thesis can be considered to have used supervised learning. In the course of our research work, only a few studies that have proposed an unsupervised method for detecting emotions expressed in short informal text (tweets) such as [28]; this represents the motivation for at least part of the present study. Furthermore, in the other part of the study we used semi supervised learning while not relying on specific emotion keywords or lexicon. This present the second motivation by trying to generate and optimise vectors that capture the essence of emotion which can be used as an emotion representative, instead of the real emotional word vectors, in the process of emotion detection distance measurements. To the best of our knowledge we are the first to optimise artificial vectors which can be used to represent emotion category in emotion detection [4]. This publication [4] was published in July 2019. While, a relevant work [8] was published later in October 2019.

In this thesis we are interested in developing and extending emotion detection methods for short text such as news headlines and social media messages. We want to focus on the methods that deal with the word embeddings. Instead of dealing with text, word embeddings transform words into vectorial representations, thus allowing distance calculations. Thus, we seek to adapt the distance calculation methods to classify the emotion type based on the calculated distances between the embeddings of the sentences and emotional word vectors. We will also explore whether the use of optimization methods would be effective to generate vectors which can represent emotion types optimally in order to be used for emotion detection.

1.2 Understanding Emotions

Psychology and behaviour science have studied the emotions because it is an important aspect of human nature [101]. Computer science researchers are becoming increasingly interested in making emotion in text understandable by Natural Language Processing (NLP). The word vector representation, such as Word2Vec [66], is one of the proposed techniques to approach such problems. It can be argued that word vectorization

with representations for semantic relationships would be useful for improving the effectiveness of NLP applications [67]. Being able to detect the emotional state from the produced text would be very beneficial in many situations. Emotion detection from text systems has many potential applications such as:

- In Human computer interaction: based on the user's emotional state, emotion detection could be used to produce a recommendation or sort of an interaction [111]
- In marketing: customers' reactions to products can be analysed by using emotion detection approaches in order to make changes in a product and create a better relationship with customers [38]
- In e-learning: using emotion detection of the users' state in e-learning applications would help to create more effective tutoring systems [104]
- In health: psychologists can use emotion detection to predict the patients' emotional state in the long term so they can infer whether a patient may be facing depression or stress [21]

Many approaches have been proposed towards detecting emotion categories in textual data. Some of these approaches will be mentioned in the following chapters of this thesis. Most of these approaches relied on a limited number of emotion keywords or a word lexicon and number of other such approaches were focused only on specific domain and tend to be inadequate in terms of being generalised to other domains. However, the problem is still unresolved, while the existence of word vectorization and distance measurements shows a possible solution.

1.3 Research Aims and Objectives

In this thesis, we are interested in developing methods and techniques by which to detect emotions from various types of short text, typically tweets and news headlines. We intend to focus on methods that deal with word embeddings. Hence, the main

aim is to study emotion detection methods from short text by exploiting and understanding the word embeddings. Word embeddings have shown promise in similar domains by capturing the essence of the relationships between the different words. With respect to emotions, in general there are relationships between the emotional words. Thus we want to explore the word embeddings in regards to emotion detection. Guided by the above aim, we look forward to achieving the following core objectives:

- Textual emotion detection by calculating the distance between emotional words and general text using word embedding representation.

As short texts have become a prevalent form of information such as news headlines, tweets and text advertisements, we seek to develop methods that can detect the emotions within the short text by the aid of word embeddings. News headlines can be considered as short texts; which are typically written in emotional language and mainly to intrigue people. Therefore, it was considered as the source of data for part our experiments because its texts' are written in formal language which is compatible with Google's pre-trained Word2Vec embedding. We intend to develop a method for identifying emotions from news headlines, by using the word embedding approach and Word Mover's Distance (WMD) [57]. We want to see if the emotions which are detected using the proposed method can better detect the emotions within text while training data labeling, ontologies and term extractors were not required.

- Search the word vector space for an "idealised" emotional vectors in order to identify emotion in informal text (tweets).

In regards to understand emotions from informal text we have chosen Tweets. As Tweets consist of a limited number of characters. Furthermore, Twitter[1] is a social media platform available and freely accessible to people regardless of their culture, age or education. Moreover, it is one of the most popular microblogging platforms [58] in existence, with almost 500 million tweets being posted on a daily basis [110]. More importantly, many of tweets load emotions as users of Twitter may express and

[1] `https://twitter.com`

post their thoughts and emotions in real-time on a daily or shorter-duration basis. We intend to develop a method for optimising an "idealised" emotional vector using evolutionary strategies in order to be used for emotion detection. we intend to adopt the Euclidean Distance function to calculate the distance between the word embedding of tweets and the "idealised" emotional vector. We want to find out if the proposed method is going to show good results in detecting emotions from tweets by adopting an evolutionary strategy with Word2Vec through distance functions.

1.4 Contributions

The work presented in this thesis has two main contributions. The first makes a novel contribution in the field of emotion detection in literature. While the second contributes in using evolutionary algorithms to find optimal vectors for emotion detection. The main contributions are as follows: we have come across many literature papers that are making contribution in textual emotion detection field. Although we have found out that these approaches are either relying on a limited number of keywords or word lexicon like: [56], [98], [10], [25], [13], [115], [95] and [6] or they are performed only on specific domain and tend to be inadequate to be applied on different domains like: [59], [38], [3] and [30].

The vast availability of both online and offline data can augment our understanding of emotions with studies of scale previously unthinkable. Moreover, a recent methodology has been developed in neural networks that transforms words into their vectorial representation [67]. It is possible to calculate the distance between emotional words and general text using this transformation [57].

Therefore, we focus on identifying and investigating what are the best novel methods of word vectorization and distance measures towards identifying emotions, while at the same time aim at cataloguing the emotional content of documents across different genres. In particular, this work is motivated by Word Mover's Distance (WMD) [57]. Unlike in other approaches, training data labeling, ontologies and term extractors are not required in this approach. Therefore, we want to demonstrate that using word

vectorization (Word2Vec) and WMD would gain a better results in regards to emotion detection. To the best of our knowledge, this work presents the first investigation into using the word vector representation along with WMD to detect emotions in text, as it was published in 2017 [5]. Not to mention that, a relevant work was published after our publication [85]. Emotion detection goes beyond sentiment analysis by detecting a set of emotions, through the expression of texts, for division of text. By considering the fact that, in this approach, data labeling, ontologies and term extractors are not required, our results surpassed the benchmark approaches' results across all emotion categories.

From emotion detection and recognition stem the idea of optimal vector generation that has the best guess of emotions. Generating an optimal vector that can be used effectively in emotion detection is challenging. We focus on identifying and demonstrating the best novel method of the evolutionary algorithms that should be able to optimise the optimal vector for each emotion category. Moreover, to the best of our knowledge evolutionary algorithms have not been adopted to optimise vectors for emotion detection in text before. We propose the new methodology based on identifying "idealised" words that capture the essence of an emotion. We look for these "idealised" vectors through searching the space of word embeddings using covariance matrix adaptation evolution strategy (CMA-ES). Our results which we achieved by using this new methodology for emotion detection surpassed the benchmark's results which produces state of the art results, surpassing classic supervised learning methods.

1.5 Thesis structure

The rest of the thesis is composed of 5 chapters whose structure is as follows:

Chapter 2 summarises the main literature review and related work done on emotion detection. We start by defining sentiment analysis and review some of the related approaches. In the next section, we present the topic emotion frameworks, which are the discrete frameworks and the dimensional frameworks. Next we review many

emotion detection systems including two parts: unsupervised learning and supervised learning. After that, we illustrate the emotion related lexicons. Subsequently we discuss the formal and informal texts and next we explain the word embedding, which is transforming words into vectorial representations. Then, we introduce the word mover's distance. After that, we describe the evolution strategies methods. First we introduce a brief history of evolutionary strategies. This is followed by a presentation of the (1+1)-ES which is the simplest evolution strategy (ES). The next subsection reviews the CMA-ES. After that we introduce the Natural Selection Strategies (NES) and Separable Natural Selection Strategies (SNES) which are another methods of evolution strategy. Next we present a summary of the chapter.

Chapter 4 describes the developed method for emotion detection using WMD. We introduced the dataset. After that, we show how WMD method is used in association with word embedding in order to detect emotion categories. Then we present the experimental work, which starts by showing how we followed the preliminary approach, then present the main approach and how the experiments were carried out. After that, two experiments were carried out to distinguish the best approach whether to use basic emotion seed words deliberately or not. Moreover, we present the findings. Furthermore, a discussion is presented. Finally, we conclude the chapter by a summary.

In Chapter 5, we propose a new methodology based on identifying "idealised" words, which capture the essence of an emotion, by using evolution strategies to see whether it performs well in emotion detection with the association of word embedding and calculating the distance using some metric function. We discuss and analyse the performance of this methodology on tweets dataset. Moreover, we test our method on two other datasets. One of them, called ISEAR, contains formally written texts. The other one, called EmoContext, contains emotional dialogues (three-turn conversation per dialogue). The purpose of this test is to assess the robustness of our method against different domains of texts. Next, we describe different embedding experimental work and results. Furthermore, a discussion is presented. Finally, We summarise the

chapter.

Finally, in Chapter 6, we conclude the thesis with a summary regarding the work presented, the research contributions and the novelty of the achieved results. Furthermore, the planned future work is outlined which shows the research scope and the recommendations that might assist in future work in the same topics.

1.6 Publications

Portions of this thesis were produced in the following publications:

1. Alshahrani, M., Samothrakis, S., Fasli, M. (2017, October). Word mover's distance for affect detection. In 2017 International Conference on the Frontiers and Advances in Data Science (FADS) (pp. 18-23). IEEE. [5]

2. Alshahrani, M., Samothrakis, S., Fasli, M. (2019, July). Identifying idealised vectors for emotion detection using CMA-ES. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 157-158). [4]

3. More work performed as part of this thesis but has not been published yet, it will be prepared for another publication on the IEEE Transactions on Affective Computing.

Chapter 2

Literature review

2.1 Introduction

The topic of emotion detection from texts is a diverse one. Emotion detection or emotion analysis of texts has the mission of identifying emotions from natural language texts such as news headlines, articles, customer reviews, tweets and blog posts. Emotion detection from textual data has become a field of interest within Natural Language Processing (NLP) and text analytics over the last few years [25] [1] [95]. Deducing emotions and opinions from textual information is a complicated task because it relies on the information that can be gathered from the text only rather than on any information from external features such as body language [61] or tone of voice. This chapter will present an overview of the background to the research area and a literature review of relevant studies etc., including those focused on some of the existing approaches to detecting emotions in text. This literature review chapter covers the following aspects (in sequence): Formal and Informal Texts; Sentiment Analysis; Emotion Frameworks; Emotion Detection; Emotion Related Lexicons; Word Embedding; and finally, Word Mover's Distance.

2.2 Formal and Informal Text

Inferring emotions from textual data is a challenging task because it depends on the information retrieved from text alone and from no other expressive features of an utterance, like face affect or tone of voice [39]. Textual data can be divided into two types: formal text and informal text [50]. It is generally accepted that recognising the emotions expressed in the latter type of text is more complicated than the same task in relation to the former [79]. This is for a number of reasons that will be discussed in more detail in this section.

Formal writing is commonly used in research papers, official documents and literary arts such as poetry and novels whereas informal text is used in daily conversations, text chat and generally in social media applications such as Twitter and Facebook [49]. However, the nature of informal text makes it more difficult to analyse or to retrieve information from - due to the following [50]: Firstly, informal text may not follow the prescribed grammatical rules and may be less well-structured than formal text; Secondly, in informal text the use of slang words and phrases is typical; Thirdly, abbreviations such as "OMG" referring to "oh my God" are commonly used; Fourthly, messages may contain spelling mistakes; Fifthly, there may also be incorrect repetitions of some letters in some words; Finally, some messages may include emoji, hash-tags, URLs and/or icons – which, of course, are not part of standard language.

2.3 Sentiment Analysis

One of the sub-fields of Natural Language Processing is sentiment analysis. Sentiment analysis is a computational technique that is used to identify the polarity of a text - that is, whether it is positive, negative or neutral [37]. Further, sentiment analysis can be considered as being the most analogous topic to emotion detection in the context of text [71].

According to Bhadane, Dalal, and Doshi [12] there are two major approaches to sentiment classification: machine learning, and lexical. Support Vector Machines

(SVMs) and naïve Bayes are the commonly used machine learning methods employed in sentiment analysis. On the other hand, the lexical approach has many variants - such as the baseline approach, Part of Speech tagging (PoS), WordNet, stop words, the negation method, N-grams and stemming [12].

With regard to differences in data scope, Kang, Yoo, and Han [47] have argued that a lexicon assembled for analyses in one domain, such as movie reviews, will not work sufficiently well in another, e.g., restaurant review sentiment analysis. They stated that the lexicons which are constructed for general or specific domains wouldn't have a clear polarity in some occasions. For instance, the polarity of the word "delicious" vary in the general domain and in the restaurant review. Therefore, they proposed a new type of lexicon based on unigrams and bigrams. Furthermore, to narrow the accuracy gap which is generally exists when making positive and negative classifications, two improved naïve Bayes approaches were created.

A lexicon-based approach to performing text sentiment analysis was proposed by Singh, Bagla, and Kaur [98]. This approach was applied to classifying whether a Facebook [1] post's comment was positive, neutral or negative. In that research, a dataset of about 7,000 positive and negative words was created. It is useful to note, here, that the proportion of negative data encountered was about 70%. The words in the lexicon were labeled with +1 for positive and -1 for negative. After data collection, the pre-processing steps included removing punctuation, articles, pronouns, prepositions,"Wh" words and auxiliary verbs. Then the adjectives, verbs and adverbs were identified and assigned a magnitude of polarity. These magnitudes were 4, 3, 2 and 1 for adjectives, verbs, adverbs and simple words respectively. The sign changes depended on the label of the word. The authors of [98] stated that the reason behind assigning these different magnitudes was that the different types of word differed critically in terms of their influence on the sentiment expressed by a text overall, the adjectives having the most influence. Thus, subsequently, overall comment polarity could be calculated.

[1] `https://www.facebook.com`

Gonçalves et al. [35] made a comparison between the eight most widely used methods in sentiment analysis: emoticons, linguistic inquiry and word count, SentiNet, SentiStrength, SentiWordNet, happiness index, SailAil sentiment analyzer and the PANAS-t (psychometric scale)). They discovered that, across all different literature domains, it was still the case that no best method could be found. Moreover, they concluded that their own combined method achieved better results.

2.4 Emotion Frameworks

Scientific studies and psychological literature have debated the categorization of human emotions since before the 1960s [29]. While Sentiment Analysis (SA) research frameworks focus on classifying documents or sentences as either positive or negative - as does, for instance, The General Inquirer (GI) [100] - this section looks more closely on the theories that are focused on the classification of documents or sentences into more specific emotion categories. Even though it is outside the scope of this work to describe or list all of the various emotion frameworks which have been proposed, nevertheless, it is important to mention the main groups and the most commonly used frameworks. Discrete frameworks and dimensional frameworks are the two most predominant categories in this regard [117]. Hence, the following section 2.4.1 presents some of the most common discrete models relating to emotions, and the subsequent section 2.4.2 discusses some of the most common dimensional models in this area.

2.4.1 Discrete Frameworks

In a discrete framework, distinct categories of emotions are employed. Ekman's is one of the most commonly used basic emotional frameworks that follows the discrete model [24]. The six basic emotions as identified in Ekman's framework are anger, disgust, fear, joy, sadness and surprise - see table 2.1. A further study by Shaver et al. [96] suggested the same six basic emotions but included "love" instead of "disgust". These six emotion categories (as given in table 2.1) are used as the main branches of

a tree structure in [96] where the model specifies that each branch has a further categorization. Plutchik [80] defined eight basic emotions "trust" and "anticipation" were added to Ekman's six categories. A bipolar taxonomy was considered in Plutchik's work: joy versus sadness, trust versus disgust, anger versus fear, and surprise versus anticipation. On the other hand, Izard [46] argued that there are ten basic emotions which included Ekman's emotions except sadness (see table 2.1). Noticeably, there is not much in the way of re-enforcement across these theories, as regards the taxonomy of emotions, but it can be seen from table 2.1 that anger, fear, joy, surprise and sadness are among the most commonly suggested categories. However, using the discrete model is easier and clearer for humans to indicate or define the emotions in a text because its categories are comprehensible and natural to humans. Furthermore, Ekman's model has been used in this thesis to evaluate our approaches for three reasons. First, it was extensively verified by psychologists [2]. Second, it is the most popular used model [81] especially in discrete categorization [63]. Third, as Ghazi, Inkpen, and Szpakowicz [31] stated that Ekman's model [24] is the most commonly adopted emotion model in NLP studies.

2.4.2 Dimensional Framework

Emotions in a dimensional framework are distributed, as the name suggests, across two or more dimensions. The circumplex model is one of the most popular dimensional models of emotion [88]. The authors suggested that emotions can be defined using two dimensions: valence and activation. "Valence" refers to the pleasure dimension, i.e, positive to negative emotions while activation, otherwise known as arousal, represents the intensity of the emotion.

The Pleasure, Arousal and Dominance (PAD) model, is another dimensional framework which uses three dimensions [89]. Pleasure and arousal measure the pleasantness and intensity of an emotion respectively, whereas the dominance dimension represents whether the emotion is dominant or submissive. However, there are three shortcomings for using the dimensional model. First, using dimensional labels are ambiguous

emotion	Ekman [24]	Shaver [96]	Pultchik [80]	Izard [46]
anger	✓	✓	✓	✓
fear	✓	✓	✓	✓
joy	✓	✓	✓	✓
surprise	✓	✓	✓	✓
distress				✓
sadness	✓	✓	✓	
disgust	✓		✓	✓
love		✓		
contempt				✓
trust			✓	
interest				✓
anticipation			✓	
guilt				✓
shame				✓

Table 2.1: Basic emotions of discrete frameworks

to humans [53] especially to lay person such as in self labelling data like tweets hashtags which makes using the dimensional model impossible. Second, it is unnatural and more difficult for humans to identify minimal numbers of dimensions for different emotions in text. Thirdly, some categories of emotions are indistinguishable, for instance, both categories anger and fear overlap in the same quadrant of negative valence and arousal levels [53].

2.5 Emotion Detection

Clearly, emotions can be recognized by facial expression, by tone of voice and by other non-verbal cues, but these are not the only ways emotions can be detected. Classifying the texts, documents or sentences into different emotion categories such as anger, fear, disgust or joy is the task which is most commonly referred to as emotion detection [115, 13]. Emotion detection from texts is more complex than sentiment analysis because its outputs are not limited to just positive, negative and neutral. This section will cover unsupervised learning and supervised learning.

2.5.1 Unsupervised Learning

Unsupervised learning comprises a class of machine learning methods which are aimed at learning from data which has not been annotated or labeled [72]. In other words, it is a method that will find patterns without the need for human guidance or supervision. Thus, the following approaches are considered to be unsupervised because they do not require annotated data for training purposes.

Kim, Valitutti, and Calvo [51] proposed some unsupervised learning techniques for emotion detection, based on both dimensional and categorical models. Three different datasets were used, ISEAR, SemEval-2007 Affective Text, and a collection of children's fairy tales. As regards categorical models, they used three different techniques: Latent Semantic Analysis (LSA), Probabilistic LSA (PLSA), Non-negative Matrix Factorization (NMF)- all using Word Net Affect [103] as a lexicon. In terms of

dimensional models, they used just Affective Norm for English Words (ANEW). Cosine similarity between the sentence vector and an emotion vector was used to assign a sentence to the closest emotion category. Although this paper comprehensively covered unsupervised emotion detection techniques, these techniques were applied only to formal texts; informal texts were not considered.

Another unsupervised emotion detection approach was proposed by Agrawal and An [1]. They started by extracting nouns, verbs, adjectives and adverbs from a sentence as a pre-processing step. Then, emotion vectors for such words were computed based on semantic relations involving the emotion concept. Also, the syntactical dependencies between the words in each sentence were considered – in order to "tune" the emotion vector. After this, the technique generated vectors for each sentence by aggregating the vectors of all the emotionally laden words in it. However, it should be noted here that this paper also looked at formal texts only.

Strapparava and Mihalcea [101] proposed both unsupervised and supervised approaches, and they applied these to the News headlines dataset that was developed for SemEval 2007 [102]. This data was collected from different news websites and Newspapers such as BBC News, Google News and the New York Times. These news titles data were divided into two parts: the development data, which included 250 titles while the rest, the test data, consisted of 1000 titles [101]. Ekman's basic emotions [23] were used for the data annotation; this annotation was done manually by six annotators. Each headline was tagged with at least one emotion category [101]. Utilizing the news data set, Strapparava and Mihalcea [101] implemented five approaches. For the unsupervised method, the Latent Semantic Analysis (LSA) vector space was adopted by three of these approaches. These three approaches adopted a Latent Semantic Analysis (LSA) vector space. The difference between these three approaches is that LSA is based on emotion words only; emotion words with the WordNet synset in order to take account of synonyms; or emotion words with all the labeled WordNet-Affect synsets for the same emotion. For the supervised approaches, first, the Word Net-Affect lexicon was employed to examine the presence of emotionally laden words

in news headlines, and for the second approach, a naïve Bayes classifier was trained and then used [101]. Again, a limitation of these approaches is that they were used only to detect emotions in formal texts.

Strapparava and Mihalcea [101] also described three other systems (SWAT, UA and UPAR7) but these systems were by others, who participated in SemEval. These three approaches can be compared with the previous five methods described above; they all used the same dataset and were subject to the same metrics for evaluation. Each approach is briefly outlined in the following paragraphs.

The UPAR 7 was a rule-based approach developed by Chaumartin [17]. It used three different lexical dictionaries: WordNet, WordNet-Affect and SentiWordNet. De-capitalization was performed on common words as a pre-processing step. Then WordNet-Affect and SentiWordNet were employed in order to detect the emotion represented by each of the headline's words. Finally, for each sentence, its emotional burden was classified using rules based on the examination of the main root of the dependency graph generated by the Stanford Parser as the main subject. Further, in order to obtain the emotional rating for each sentence, the rating of the 'main' word was multiplied by 6.

The UA [54] was another approach developed to classify emotions expressed in news headlines. This system was based on the hypothesis that a relatively frequent co-occurrence of words together with an emotion word is likely to mean that this co-occurrence expresses the same emotion (as the emotion word) very often. First, statistics were gathered from the Yahoo, MyWay, and AllWeb search engines. Second, the Point Mutual Information (PMI) technique was used to compute the emotion scores.

The SWAT system [48] was a supervised approach also developed to classify emotions in news headlines. In this approach, Roget's New Millennium Thesaurus 6 was queried in order to create word emotion maps. Every word in each headline was then scored using such a map. In order to label the headline with a particular emotion, the average of the headline's word scores was taken into consideration.

The first five approaches (as described above [101]) were applied to the same data, as has been mentioned, and were subjected to both coarse-grained and fine-grained evaluations. Moreover, Strapparava and Mihalcea [101] stated that the systems differed in terms of their overall abilities. In addition, a noticeable limitation which can be gleaned from the results is that not one of these eight systems which have been described achieved the best (attained) F1 score for all or even most emotion categories.

Ezhilarasi and Minu [25] introduced an approach to classifying and recognizing emotions automatically using an ontology, relying on WordNet [68], which stores emotion verbs. This approach proceeds in three main steps: text processing, emotion ontology creation, and emotion classification. In order to extract the emotionally laden words in the first step, term extraction was applied to the input text. In the second step, emotions were inferred from the domain knowledge in order to perform concept and relationship identification, in turn, to create the emotion ontology. WordNet was then used for appropriate meaning checking and synset collection. In the last step, the synset of all the emotions encountered was used to create the ontology. Finally, based on the emotion-class hierarchy of the ontology, the sentence was classified into one of the emotion-classes. However, this approach had three main limitations. Firstly, the ontology only included verbs which contained less than 5500 verbs. Secondly, the ontology has to be recreated in order to be used for a different domain. Finally, the ontology is not published for others to re-use and build on.

In order to detect emotions expressed in emails Gupta, Gilbert, and Di Fabbrizio [38] used one of the "boosting" family of algorithms. The algorithm employed was one called BoosTexter which had been proposed by [90] as a classifier. For feature extraction, the authors used eight different feature lists which comprised a set they called "salient features". These salient features included negative opinions, negative emotions and threats: for example, indicators that a customer was willing to take legal action. The authors [38] clarified that using salient features with n-gram resulted in significant improvements to the performance of the classifier. However, their work

only determines whether an email is emotionally laden or not; it does not annotate it in relation to any of the commonly used emotion types. Moreover, due to the following two reasons it would not be appropriate to use this approach as a first step, to filter whether the sentence is emotionally laden and then proceed with identifying the emotion, in another approach. Firstly, it is built specifically, to identify whether an email is emotional, for customer care domain. Secondly, it relied on limited size of training data which produced a limited number of words and phrases that are used as labels for salient features.

A Support Vector Machine (SVM) combined with the machine learning technique, Conditional Random Field (CRF), was used to classify emotions expressed in web blog corpora by Yang, Lin, and Chen [115]. In this approach, emoticons in the training data were considered as emotion indicators. An emotion lexicon is built by forming the features which could be used for emotion classification, sentence by sentence. The emotion category was then assigned to the sentence by the SVM classifier. The given context is considered by CRF to determine the appropriate emotion. Finally, the trained classifier was applied to the entire document. Emotions, in this experiment, were categorized as either positive or negative. The positive emotion category was further divided into happy or joyful while sad and angry were the subcategories for the negative emotion category. However, certain specific criteria were taken into account when applying classifications at the document level. For example, the emotional burden of the final sentence plays an important role in the overall emotion detection for the document. A limitation of this approach is that it relies on its own created emotion lexicon (which had a limited number of key words). Moreover, the final sentence at the document level may not, in all cases, necessarily be emotion representative of the whole text.

Aman and Szpakowicz [6] developed a method, which they tested on a blog dataset, that classified whether a sentence was emotional or non-emotional, using naïve Bayes and SVM. They collected the dataset of blog posts from the web. The annotation was conducted at the sentence level and was based on Ekman's six basic emotions plus

no-emotion and mixed-emotions classes. Each sentence was subject to two judgment processes. The authors themselves produced the first set of annotations, while another three annotators created the second set. The agreement of these two inter-annotator judgments was measured using Cohen's kappa coefficient [19]. Further, the sentences from the blog posts were categorized into just the six basic emotions in the continued study [7]. They then applied corpus-based unigrams, the emotion lexicon features from Roget's Thesaurus and Word Net Affect to the SVM trained classifier. They concluded that the combination of these (above) three features gave the best F1 score results in comparison to their baseline experiment - which relied on assigning each sentence the category to which the largest number of emotionally-laden words in the sentence belonged. A limitation of this approach is that it relies on words from emotion lexicons.

2.5.2 Supervised Learning

Supervised learning is a methodology whereby the learning process is provided with, alongside the input data, an annotation of this and/or the desired output. These data are required for the training phase of the model so that it can then generate outputs which are close enough to the desired outputs. The following approaches are considered to be supervised because they all require an annotated corpus for training purposes. This subsection can be divided into two main categories: lexicon-based methods and learning-based methods. Lexicon-based methods are mainly relying on predefined keywords and lexicons in order to classify emotions in text. While, learning-based methods are based on machine learning trained classifiers in order to detect emotions in text.

Lexicon-based methods

A lexicon-based approach to performing text sentiment analysis was proposed by Singh, Bagla, and Kaur [98]; the specific purpose of this approach was to classify whether a Facebook post's comments were positive, neutral or negative. This ap-

proach was mentioned earlier in 2.3. However, despite the fact that this approach was designed specifically for sentiment analysis purposes within this particular context, it had two main limitations in this regard. First, more than two thirds of the dataset examined in detail consisted of negative words. Second, word abbreviations were ignored by the technique employed.

An emotional lexical choice was defined by Bautista, Gervás, and Dıaz [10]. In this study, the proposed process was described as having two inputs, a term and an emotion, and one word as an output. While, the term here was defined by the vector (word, type) where type can be defined as: noun, adjective, verb or adverb. On the other hand, the emotion represented by a vector of emotional dimensions (activation, evaluation). In addition, it was carried out over four steps. First, a disambiguation process was performed on the term's words – the procedure adopted for this was to select the first synset in Word Net for each word. The second step was to collect the lexical choice that represented the first synset. Thirdly, they assigned an emotional annotation to each of these lexical choices. Finally, depending on the emotion input, one output module was selected as being the one with the shortest vector from the results of calculating the Euclidean distance between the emotion which was input and the candidate words. The output of this process was positive, negative or neutral; there were no further emotion categorisations provided.

Lei et al. [59] described a technique for detecting emotions expressed in on-line news items. This method had the following phases: document selection, the Parts of Speech (PoS) were tagged then the generated lexicon-based emotions were used to estimate the emotions expressed. However, in this paper emotion detection was performed only on formal texts.

Tao [107] introduced the emotion estimation net (ESiN) which estimates the overall emotion expressed by looking at the combination of the emotion content words and emotion functional words (EFWs). Emotion keywords, modifier words and metaphor words are the three categories comprising the emotion functional words. The basic emotion values are provided by the emotion keywords. If the word is considered an

emotion keyword, then weights are assigned to six tags each representing an emotional state. Modifier words are the kind of words which influence (enhance or weaken the intensity) or change (negate) the emotional state expressed by the sentence. Metaphor words are of two types: spontaneous expressions and personal character indications; both of these types have only a latent influence on the emotional state. For both modifier and metaphor words, a coefficient is marked in the lexicon. ESiN followed three main steps in order to detect an emotion expressed in a sentence. Firstly, a PoS tagger was used to tag the input text and then the EFWs were checked and assigned emotional ratings. Secondly, a weight is assigned to the emotion keywords and the links among EFWs were found. Thirdly, a decision was made based on the emotional value and a propagation formula which used both nodes and routes. In this latter step, a word in ESiN is represented as a node that has three features: emotion states, a corresponding weight and semantics. In between these nodes there are routes, which indicate the propagation of the emotion. Such a route is composed of transmission probability, direction and a decreasing propagation coefficient. A limitation is that this experiment operated on a Chinese lexicon not an English one. The same experiment could be run in English but it is still limited to a number of emotional keywords.

A framework for emotion analysis which was focused on assessing the emotions expressed in tweets was proposed by Kumar, Dogra, and Dabas [55]; this employed opinion mining. In the proposed model, unwanted data are removed first as a pre-processing step. After this, opinion words (adjectives, verbs, and adverbs) were extracted and emoticons that had an emotional meaning were replaced with a text translation. It is of note that adverbs (like “not”) were also considered by this system, as their use can change the meaning of a phrase from positive to negative or vice versa. After data tokenization, a PoS tagger was fed with the extracted words. Furthermore, a survey had been conducted on approximately 500 students in order to associate common adjectives with a number from 0 to 5, each number representing one of a list of five main emotions. Moreover, each verb and adverb was given

a weight between -1 and +1, as specified in Kumar and Sebastian [56]. Then, the value vector of the adjectives was found by employing a corpus-based method while a dictionary-based approach was applied to find the semantic orientation of the verbs and adverbs. After the determination of these scores, the average score of the tweet's emotion was calculated using a linear equation. One of the limitations of this method was that the corpus-based component only included about 1000 adjectives. Further, the verbs and adverbs recognised were limited to the most frequently used adverbs and verbs - along with their synonyms and antonyms [56].

Sykora et al. [105] created an "emotive ontology" in order to detect emotions expressed in informal texts. This approach was designed to detect emotions across a range of eight categories: anger, surprise, confusion, shame, disgust, fear, sadness and happiness. During the ontology creation process, many dictionaries were used as resources, such as: Dictionary.com, Word Net, the Oxford English on-line dictionary, the Merriam-Webster on-line dictionary and Thesaurus.com. Furthermore, in order to achieve the specific aim of detecting emotions in informal texts some slang dictionaries were taken into consideration, such as: the Dictionary of Slang, the Leicestershire Slang Page, and the On-line Slang Dictionary.

Learning-based methods

To categorise the emotions expressed by sentences taken from children's fairy tales, Alm, Roth, and Sproat [3] proposed a method using supervised machine learning and the Sparse Network of Winnows (SNoW) learning architecture. A corpus of 185 children's stories manually annotated with eight emotions was used in this study. The authors Alm, Roth, and Sproat [3] proposed a sentence level emotion classifier which used those annotations and employed a supervised machine learning technique based on a linear classifier, SNoW. This classifier required the input of a number of specific features - such as the first sentence in the story, the emotion words from WordNet and thematic story type. Their experiment was conducted on 22 children's stories and the results were found to be better than the naïve Bayes baseline and the results

of the Bag of Word (BOW) approach as reported in [3]. A noticeable limitation of this technique is that while it seemed to work well on children's stories, it would be quite difficult to adapt it to other domains.

The framework presented by Shaheen et al. [95] was used to recognize the expression of Ekman's six emotions [23] within English sentences. Their methodology consisted of two main stages: an off-line stage (training) and a comparison and classification stage. The first stage comprised three steps. The first step was to tag each word of the input text using the Stanford PoS tagger. In the second step, the Stanford dependency parser was applied to extract the input sentence's dependency tree. The third step was non-emotional content removal; such content was deleted from the tree by following a set of rules - two separation rules and three deletion rules. Ignoring the sentence before the word "but" and ignoring the sentence after "as" were the two separation rules (these rules were also applied to words that have the same effect). The three deletion rules were as follows: verbs with no object and connected to WP pronouns (like "who" and "what") or WRB adverbs (like "where" and "when") were removed; "to be" verbs, and node and non-emotional verbs were also removed; and nodes of pronouns that have no other nodes connected to them were removed as well. This completed the off-line stage. The trees resulting from all the sentences used as training data in the first stage represented the Emotion Recognition Rules (ERRs) which were to be used in the next stage. This, the comparison and classification stage, consisted of comparing the annotated ERRs (which each exemplified a sentence in the training set) with the ERRs of the input sentences by applying one of the three classifiers: k-Nearest Neighbours (KNN), Point Mutual Information (PMI), and Point Mutual Information with Information Retrieval (PMI-IR). This method was tested on two quite different training sets. The first experiment used the blog posts dataset from [6]. Their system achieved better results (F-score 84%) than the naïve Bayes baseline (which recognized emotions based on the presence of emotion words). Moreover, this system's results were better, across all emotion categories, than those of EmoHeart [73] with respect to the same dataset. The second experiment used a

dataset of tweets [112], which were labeled with an emotion derived from the emotion hash tags which were present. The ERR system yielded an average F-score of 84% over all the emotions except disgust (because this emotion was not covered in the dataset). The ERR system results outperformed EmoHeart [73] on this, latter, test data as well. A drawback of this approach is that it relies on annotations (limited to key words).

A multi class SVM emotion classifier for the tweets domain was presented in Balabantaray, Mohammad, and Sharma [9]. First, the collected data (a corpus of more than 8000 tweets) were manually labeled by five different annotators. The tweets were annotated in relation to seven emotional classes: neutral plus Ekman's basic emotional classes [23] (anger, fear, sadness, disgust, joy and surprise). Then, an SVM was used for feature classification. The features presented to this were derived from the Word-net Affect emotion lexicon, the Word-net Affect emotion PoS, the Word Net Affect emotion lexicon with left/right context, adjectives, PoS, Unigrams, Bigrams, Personal-pronouns, PoS-Bigrams, Emoticons and a Dependency-Parsing Feature. In the process of this experimental work, leave-one-out cross-validation was also used.

Similarly, Roberts et al. [86] created a corpus which consisted of 7000 tweets. These tweets were manually annotated using Ekman's [23] emotion categories and the addition of "love". For each of these seven emotion classes, a binary SVM was used to determine whether a particular tweet expressed that particular emotion. This approach applied many sub-systems such as WordNet hypernyms, WordNet synsets, trigrams, unigrams, bigrams, and the noting of question marks and indicators of exclamation. The experiment was performed using a 10 fold cross validation.

Annotating tweets with the appropriate emotions manually is, nevertheless, very challenging and time consuming. Instead, emotional hash-tags actually contained in tweets have been used by some researchers to construct, in effect, self-labeled data – as in [69, 20, 112]. Wang et al. [112] surmised that emotional hash tag data are more accurate than manually labeled data because a post with such a hash tag has, effectively, been labeled by its author rather than by somebody else. Hasan, Agu,

and Rundensteiner [44] also argued that the conventional method of annotation is time-consuming, labour-intensive and tedious in comparison to annotation based on self-labelling.

Wang et al. [112] collected approximately 2.5 million tweets which were then labeled automatically using the emotional hash-tags which were present. In order to verify the quality of the hash-tag annotation, 400 tweets were randomly selected and then manually labeled. Comparing the two, the manually labeled tweets and the hash-tag annotated tweets after some heuristic filtering, [112] stated that an acceptable level of consistency was demonstrated. They then explored the utilization of various different features such as differing emotion lexicons, PoS, n-grams and the processing of adjectives. For this research, two machine learning systems were used: Multinomial naïve Bayes and LIBLINEAR.

The efficiency of using hash-tags for emotion self-labelling of tweets has also been validated by Hasan, Agu, and Rundensteiner [44]. Some heuristic rules were applied on the collected tweets and this resulted in their being 134,000 labeled tweets once this processing had been completed. In this study as well, a supervised classifier was trained to detect emotion. Specifically, they ran SVM and K-Nearest Neighbours (KNN) systems on the training data using features such as unigrams, emoticons, punctuation and a list of negation words.

Binali, Wu, and Potdar [13] introduced a hybrid approach to detecting emotions in text. This combines a keyword and a learning-based method. The keyword method used relied on the existence of emotion words (keywords) which could be found in an emotions lexicon. The authors [13] employed this keyword-based system as a pre-processing step which included tokenization, Part of Speech tagging (POS) and the use of gazetteer semantic information to categorise each emotion expressed in each sentence into one of the six basic emotion classes. Then, a trained SVM classifier was employed in the learning-based step; this used the pre-processing features from the first step to classify the input text according to the six classes of emotion. The final step was categorising each emotion as either positive or negative. They concluded that

96.43% of the test data were accurately classified. The limitation of this approach is that the final result was a categorisation of the input text only as either positive or negative.

Discussion

Many studies relied on annotated datasets for emotion detection such as Balabantaray, Mohammad, and Sharma [9] and Roberts et al. [86]. Whether manual or automatic annotation is used, annotation is always keyword-based since it relies on the presence of the emotionally charged words from a pre-determined emotion lexicon. However, such an approach has a number of limitations: the emotion lexicon is limited in extent, and, furthermore, some words are ambiguous or related to more than one emotion category. Moreover, in regard to manual annotation, this is time consuming due to the large quantity of training data required and other difficulties resulting from the ambiguity which arises when different people are asked to perform the annotation – based on their differing judgements.

All the studies that have been described in this subsection can be considered to have used supervised learning. In the course of our research work, we have not yet come across any study that has proposed an unsupervised method for detecting emotions expressed in short informal text (tweets); this represents the motivation for at least part of the present study.

2.6 multi-classification

Most multi-class classification approaches are based on binary classification methods. For instance, Sentiment Analysis is considered to be one verses one binary classification. moreover, Emotion detection approaches in many cases are based on The one verses rest strategies which work by reducing the problem into multiple binary classification task. This single classification strategy is followed in this study. However, the multi-label classification will be mentioned here briefly as it is beyond the scope of this study. A sample of document or a sentence is assumed to have a single emotion while

in many cases the text can be multi-emotional which means it can contain several emotions at the same time. Probabilistic topic modeling is a suite of algorithms that aim to annotate textual documents with thematic information by using probability theory. probabilistic latent semantic analysis (PLSA) and Latent Dirichlet allocation (LDA) perhaps the most common topic models [64].

Luyckx et al. [63] proposed study on multi-label classification of emotional texts. They focued on a dataset of notes which were written by people who have committed suicide. The task is to predict labels of a note among 15 possible emotions, such as love, hopelessness, thankfulness, pride, etc.

2.7 Emotion Related Lexicons

It is common practice in emotion analysis to rely on emotion lexicons, this is especially so for studies that work on both learning-based and lexically-based methods (which, as has been said, are the ones studied here). As a lexicon is defined to be a list of words of a specific language [60], emotion lexicons are dictionaries of words that are labeled or annotated in terms of one or more of categories of emotions. This section highlights some of the predominant lexicons used for emotion analysis.

Affective Norms for English Words (ANEW) is one of the most popular emotion lexicons [14]. The ANEW lexicon consists of 1,034 English words that are related to emotions. These words have been labeled in relation to three dimensions by a group of students. The emotional dimensions used are valance, arousal and dominance, representing unpleasant and pleasant; calm and excited; and dominated and in control respectively.

WordNet-Affect is an emotion lexicon developed by Strapparava, Valitutti, et al. [103]. The authors built this lexicon based on the WordNet [68] lexicon by annotating the synsets with a corresponding emotion. WordNet-Affect employs six emotion categories which correspond to Ekman's [24] basic emotions. They started their annotation process by manually labelling the initial set of affective seed words and then they expanded this set by appending all these words' correlated verbs, nouns, adverbs

and adjectives [103].

Another emotion lexicon is the EmoLex which is commonly known as the NRC (referring to National Research Council in Canada) lexicon, the word-emotion association lexicon [70]. The authors developed EmoLex by using Amazon Mechanical Turk crowd-sourcing[2]. EmoLex contains more than 14,000 words which have been annotated manually with one or more of Plutchik' [80] basic eight emotions in addition to the two sentiments positive and negative (making ten classifications in all). One major difference between this lexicon and WordNet-Affect is that, in this lexicon, each word can be associated with multiple emotion categories whereas each word of WordNet-Affect belongs to only one emotion category.

Another dictionary that is related to emotions is the Linguistic Inquiry and Word Count (LIWC) lexicon [76]. This lexicon provides a set of words and terms that are sorted into one or more categories, including emotional categories such as sadness, anger and anxiety. LIWC was first introduced in 1993. Since 1993, LIWC has been expanded and updated a great deal. The second and third updates were in 2001 and 2007 respectively. While LIWC 2007 contained 4500 words and word-terms, LIWC was recently expanded in 2015 to include 6400 words and term-words [76]. Clearly, there is always a requirement to improve any particular emotion related lexicon, even a very large one such as this one - which yet is still limited in terms of the number of words that have been included.

2.8 Word Embedding

Deducing emotions, opinions, facts and so on from contextual information is a complicated task because it relies on the information which can be gathered from the text only. However, word vector representations are one of the techniques proposed for solving the problems associated with this task. Word embeddings operate by transforming natural language text into vectors of real numbers that capture the essence of the relationships between the different words. This section will introduce some

[2] `https://www.mturk.com`

of the methods that are related to word vectors or which use word vectors, as such, including those which will be used in our experimental approaches.

The words "happy" and "happier" are similar or, in other words, they have a strong relationship. In the same sense, which word is similar to sad? In this context, the target word must be in a similar relationship to "sad" that "happier" is to "happy". This can be solved by applying primary algebra and calculating vector X = vector ("happier") - vector ("happy") + vector ("sad"). Now the cosine distance can be measured to find the closest word (vector) to X [67]. However, with respect to emotions there are relationships both between each emotion word and obviously between the emotion itself and its associated words. Mikolov et al. [67] recommended that word vectorization, given the existence of semantic relationships, might be useful for improving NLP applications.

2.8.1 Continuous Bag-of-Words Model

Bag-of-words (BOW) architecture is similar to a neural network model but the network itself here only has input, projection and output layers. Only one projection layer is included, which all words can be projected to; then the average will be taken for all the vectors. Continuous Bag of Words (CBOW) is a modified BOW model which employs a continues distributed representation based on the context to predict the target word [67].

2.8.2 Continuous Skip-gram Model

The skip-gram architecture is also similar in structure to a neural network model and has the same layer architecture as CBOW. However, in fact it is the inversion of CBOW because it works by employing the current word (as input) to predict a range of words surrounding it [67]. The skip-gram model tends to be faster in regard to training than other neural embeddings [26].

2.8.3 Word2Vec Embedding

Word2Vec is an open source word embedding predictive model which can be trained by the approach described in [66]. It contains 3 million embeddings related to English language words and phrases from Google News [57]. It is an adaptation of both models: continuous bag of words and skip-gram [67]. It 'neighbours' the vectors of similar words in the same vector space [57]. Word2Vec does not need data annotation or labelling in order to create meaningful representations. Word2Vec was designed to learn the features and relationships between words without any human supervision. For example, it can predict that the pair of words 'Spain' and 'Madrid' have a strong relationship (see Figure 2.1). So for example, in the same sense, which word is similar to France? the target word must be in a similar relationship to the word France as Madrid is to Spain. This problem can be solved by employing primary algebra and calculating vector X = vector ('Madrid") - vector ('Spain') + vector ('France'). Now the cosine distance can be measured to find the closest word vector to X – which, of course, is Paris in this case [66]. However, in relation to emotions, there are relationships between each emotion word and obviously between the emotion category and its associated words.

2.8.4 GloVe Embedding

Global Vectors (GloVe) [77] is a count-based word embedding model which was developed a year after Word2Vec. It is an open source model that was produced by Stanford[3]. GloVe has been trained on several corpora, including a corpus of two billion tweets which employs a vocabulary of approximately 1.2 million words. Another corpora that was used to train the model consisted of 6 billion words extracted from a combination of both Wikipedia2014[4] and English Gigaword5[5]. Furthermore, GloVe, like Word2Vec is clearly an unsupervised model focused on obtaining word embedding. Moreover, word vectors from GloVe capture many linguistic regularities such as

[3] https://nlp.stanford.edu/projects/glove/
[4] https://dumps.wikimedia.org/enwiki/20140102/
[5] https://catalog.ldc.upenn.edu/LDC2011T07

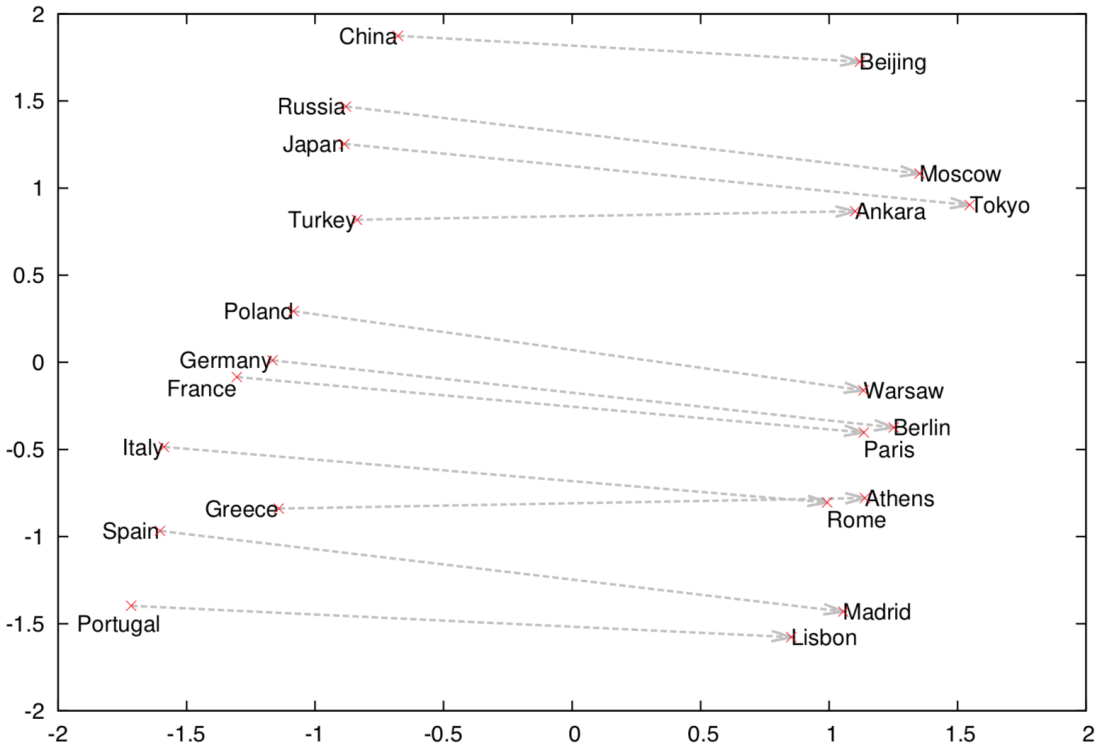

Figure 2.1: Country and Capital Vectors in Word2Vec Projected by two dimensional PCA (Image taken from [66] page 5).

male and female as well as city and zip code.

2.8.5 ELMo

Embeddings from Language Models (ELMo) [78] was introduced as a new term in embedding in 2018. The authors created ELMo to address two main shortcomings of previously proposed word emedding such as Word2Vec. The two main challenges authors claimed ELMo addresses are that ELMo is able to model both complex characteristics and polysemy. Therefore, a word with two or more different meanings or which appears in a number of different contexts can be represented by a number of different vectors. This contextualized word embedding uses the bi-directional Long Short Term Memory (LSTM) architecture [78]. The pre-trained version of ELMo is trained on the 1 billion word benchmark from [18] – a very large dataset, of course.

2.8.6 BERT

The Bidirectional Encoder Representations from Transformer (BERT) system is another open source pre-trained word embedding model; this was published by Google in late 2018 [22]. BERT is a context sensitive embedding, like ELMo, but it uses transformer architecture instead of LSTM. BERT utilises the Mask language model in combination with a transformer to train the system on an entire sentence in parallel. The BooksCorpus [118] and the English Wikipedia are the two corpora which have been used to train BERT; these contain 800M words and 2,500M words respectively [22].

2.8.7 XLNet

Yang et al. [116] proposed XLNet in 2019; thus, this is a recent, state-of-the-art word embedding system. XLNet is a generalized autoregressive model which utilises permutation language modelling instead of relying on masked language modelling. Furthermore, in order to improve the results and to process longer sentences, XLNet uses Transformer-XL. A noteworthy feature of XLNet is that it captures the dependency between pairs of words like (New York) while the previous state-of-the-art system, BERT, neglected such a dependencies. XLNet is pre-trained on various corpora which altogether come to 33 billion tokens.

2.9 Word Mover's Distance

The Word Mover's Distance (WMD) method [57] is an approach which computes the distance between two text documents. WMD, as originally implemented, employed the word vectorization features of Word2Vec embedding, although in fact, it can employ any other word embedding. Kusner, Sun, Kolkin and Weinberger [57] stated that their WMD method can be considered as a special instance of Earth Mover's Distance [87]. Word Mover's Distance derives the dissimilarity between two documents by taking advantage of the embedded words of each document. This distance

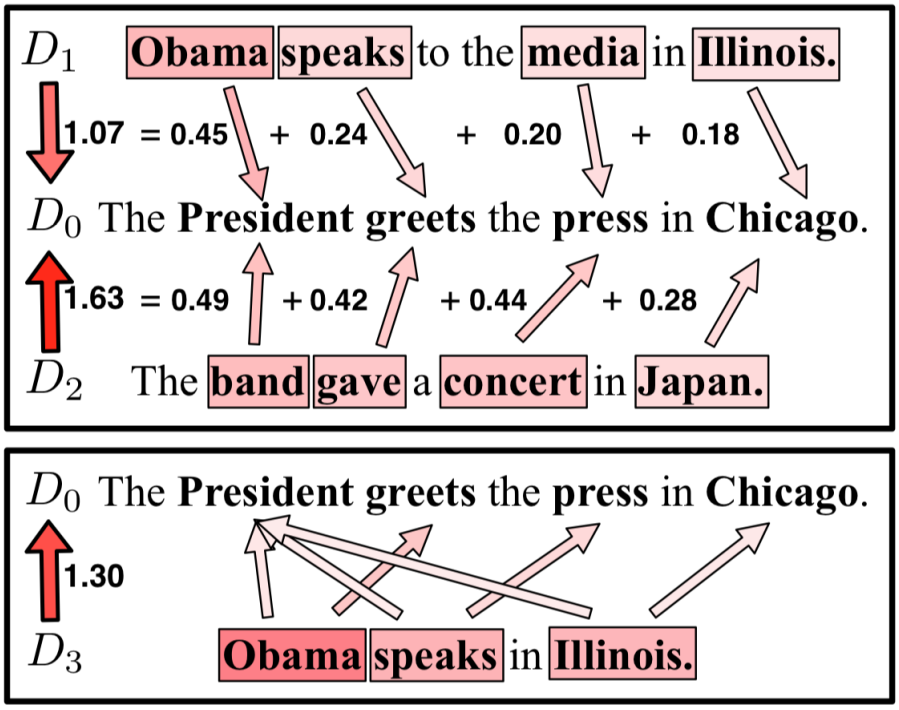

Figure 2.2: Illustration of WMD between documents while (top) with equal BOW (Bottom) with different count of words (Image taken from [57] page 3).

is measured by the sum of the minimal distances that all the word embeddings in the first document need to move to make the two documents match [57] (see Figure 2.2). The smaller the distance yielded by WMD, the more similar are the sentences of the two documents; this can be seen from Figure 2.2: sentence D1 is more similar to D0 than it is to D2.

This assumes that the normalized bag-of-words (BOW) vectors represent the text documents. For instance, if a word a in a document appears c_a times, then the weight $s_a = \frac{c_a}{\sum_{b=1}^{n} c_b}$, where n refers to the words count in the document. A word dissimilarity method which measures a pair of words' semantic similarity is required; the Word2Vec embedding approach is used for this. The distance c between word a and word b is calculated by $c(a, b) = ||X_a - X_b||_2$, where X_a and X_b are the embedding vectors of the words a and b respectively.

Let s and s' be the normalized BOW vectors for two documents. Let $T \in R^{n \times n}$ be the flow matrix, where $T_{ab} \geq 0$ refers to how much it costs to transport from a in s to b in s'. The outgoing flow from a equals s_i and the incoming flow from b equals

s'_j must be ensured to entirely transport s to s'. The WMD, between s and s', is defined by the minimum cumulative distance to move s to s': i.e., the following linear program as described in [57].

$$\min_{T \geq 0} \quad \sum_{a,b=1}^{n} T_{ab} c(a,b)$$

$$subject\ to \sum_{b=1}^{n} T_{ab} = s_a, \forall_a \in \{1, ..., n\}$$

$$\sum_{a=1}^{n} T_{ab} = s'_b, \forall_b \in \{1, ..., n\}.$$

Significantly, WMD disregards word-ordering which makes it suitable in this case for emotion detection from short texts - by computing the distance between sentences and between emotionally laden words. With this in mind, the authors stated that they had achieved remarkable results in terms of sentiment analysis [57]. Another key point to remember is that WMD is able to measure the semantic similarity between two documents even when they don't share words in common. Unlike other methods such as term frequency–inverse document frequency (TF-IDF) and bag-of-words, both of which compute the similarity between two texts based on the appearance of words in common. It should be noted that WMD is considered an unsupervised method.

2.10 Evolution Strategies

Evolution strategies (ES) represents an optimisation method based on the theory of evolution [41]. The use of evolutionary methodologies in computation can help to solve some problems and to find optimal solutions [45] [27]. ESs have some basic elements which they typically work with a population of individuals at a time (individuals are the candidate solutions). Furthermore, they use a selection method biased by fitness function which can be a measure of quality of the individual. Hence, the better the fitness of an individual, the more likely to be selected for the breeding of the next generations. Evolution Strategies represents one of the topics, alongside

Genetic Algorithms (GA) and Evolutionary Programming, in the research area of Evolutionary Algorithms. ES has frequently been employed to solve the problem of continuous black box optimisation. The continuous black box optimisation aims to find the optimal solution x contained within the continuous search domain $\mathbb{R}^d$. In order to find optimal solutions, ES frameworks adopt the following which is consisting of four basic steps:

1. Stochastically generate some individuals to make up the population which is to evolve
2. Based on a fitness function (selection algorithm), select the best individuals to be the parents of the next generation
3. Generate the new offspring via the selected parents (the best individuals selected in step 2)
4. If the desired results have not yet been achieved, then go back to step 2.

ES has been applied to solve a number of different problems relating to real world applications. Application areas where ES has been used include, but are not limited to: Parameter Estimation [45]; Image processing [62]; Car Automation [75]; path planning for mobile manipulators [113]; and Task scheduling [36].

In this thesis, we intend to utilise ES to generate a vector intended to represent a desired emotion category optimally in order to detect emotions from text, in this section, we present a brief history of ES (section 2.10.1) and then a discussion of some related topics. Four main related topics are covered here, as follows: (1+1)-ES (section 2.10.2); the covariance matrix adaptation evolution strategy (section 2.10.3); natural evolution strategies (section 2.10.4); and finally, separable natural evolution strategies (section 2.10.5).

2.10.1 Brief history

The topic of Evolution Strategies was first introduced in the 1960s at the Technical University of Berlin by Rechenberg [83]; Further developments were made in the

1970s by Schwefel [93]. According to Beyer and Schwefel [11] ES has endured as an active research field for many decades, and it remains an active research field at present. Since the 1960s the ES framework has been developed comprehensively and this development includes (1+1)-ES which has been further augmented to form frameworks based on the 1/5th success rule [84]. Many alternative research studies have used forms of step size adaptation ES σ such as self-adaptation [94] and cumulative step-size adaptation (CSA) [74].

In preceding years, researchers have investigated the potential advantages of using full covariance matrices for the process of mutating to the next generation of individuals [43]. The Covariance Matrix Adaptation Evolution Strategy, CMA-ES, [42] is one of the most popular kinds of ES and it has been demonstrated to be successful in many studies: for instance, [27] and [97]. The Natural Evolution Strategies (NES) approach [114] is another class of ESs which was introduced in 2008. Such a strategy maintains a search distribution by utilising the natural gradient present in order to update the distribution's parameters. Moreover, Separable Natural Evolution Strategies SNES [91] represent one of the instantiations of the NES family designed to minimise complexity and to optimize in relation to the high dimensionality problem. Over the past few decades, many annual conferences such as The Genetic and Evolutionary Computation Conference (GECCO) and the Congress on Evolutionary Computation (CEC) have contributed to EA and ES specifically.

2.10.2 (1+1)-ES

The (1+1) Evolution Strategy [83] is arguably the simplest evolution strategy possible. It is commonly known as *two membered ES* as well. From the name (1+1)-ES, it is relatively straightforward to infer that in this method just one parent generates just one offspring per generation. Thus, (1+1)-ES starts with a single population x. Let $x \in \mathbb{R}^d$, and (1+1)-ES apply the following loop:

1. generate a stochastic vector, $m \in \mathbb{R}^d$, as chromosomes;

2. create $x' \in \mathbb{R}^d$ by $x' := x + m$ (one offspring only);

3. using the fitness function, if x' is better that x then it becomes the current parent ($x = x'$) otherwise x will remain the parent for the next generation (greedy selection);

4. If the desired results have not been achieved yet then go back to step 1, otherwise output x.

2.10.3 CMA-ES

The Covariance Matrix Adaptation Evolution Strategy CMA-ES is a stochastic optimization algorithm proposed by Hansen, Müller, and Koumoutsakos [42]. CMA-ES is an iterative evolutionary algorithm based on a set of solutions representing continuous optimization. The CMA-ES is considered to be a Monte Carlo method. In general it iterates over three steps. First, it generates λ new vector solutions from a multivariate Gaussian distribution according to:

$$x_i \sim \mathcal{N}(m_k, \sigma_k^2 C_k) = m_k + \sigma_k \times \mathcal{N}(0, C_k) for\ i = 1, ..., \lambda \tag{2.1}$$

Second, it evaluates the fitness of the sample solutions generated in the first step. Third, it updates the sampling distribution by adapting the mean m, step size σ and covariance matrix C from the best solution of the population. By repeating the above procedure, the CMA-ES moves the sampling distribution towards an optimum solution. The mean vector m always denotes the favourite solution at the current generation, and the step size σ and the covariance matrix C control the length of the step and the distribution shape respectively. The mean m of the distribution is updated via

$$m_{k+1} = \sum_{i=1}^{\mu} w_i \, x_{i:\lambda} = m_k + \sum_{i=1}^{\mu} w_i \, (x_{i:\lambda} - m_k) \tag{2.2}$$

With $\sum_{i=1}^{\mu} w_i = 1$, where the symbol $i : \lambda$ denotes the i-th best individual according to the objective function. The step size σ_k is updated after updating the evolution

path p_σ as :

$$p_\sigma \leftarrow (1-c_\sigma)p_\sigma + \sqrt{1-(1-c_\sigma)^2}\sqrt{\mu_w}\, C_k^{-1/2}\, \frac{m_{k+1}-m_k}{\sigma_k} \tag{2.3}$$

$$\sigma_{k+1} = \sigma_k \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\parallel p_\sigma \parallel}{E \parallel \mathcal{N}(0,I) \parallel} - 1\right)\right) \tag{2.4}$$

Finally, the covariance matrix C is updated by using the update of the evolution path p_c:

$$\begin{gathered} p_c \leftarrow (1-c_c)p_c + \mathbf{1}_{[0,\alpha\sqrt{n}]}(\parallel p_\sigma \parallel)\sqrt{1-(1-c_c)^2} \\ \sqrt{\mu_w}\, \frac{m_{k+1}-m_k}{\sigma_k} \end{gathered} \tag{2.5}$$

$$\begin{gathered} C_{k+1} = (1-c_1-c_\mu+c_s)C_k + c_1 p_c p_c^T + \\ c_\mu \sum_{i=1}^{\mu} w_i \frac{x_{i:\lambda}-m_k}{\sigma_k}\left(\frac{x_{i:\lambda}-m_k}{\sigma_k}\right)^T \end{gathered} \tag{2.6}$$

2.10.4 NES

Natural Evolution Strategies NES represents a recent variant of ES which was introduced by Wierstra et al. [114] in 2008. The NES approach is a class of evolutionary algorithms implementing a real-valued optimization. It maintains a search distribution and it follows the natural gradient to update the distribution's parameters.

For each generation the algorithm produces a population of $n \in \mathbb{N}$ samples $z_i\ \pi(z|\theta), i \in \{1, ..., n\}$, independent and identically distributed from its search distribution, which is parameterized by θ with the goal of maximizing the fitness function $f : \mathbb{R}^d \to \mathbb{R}$. The expected fitness under the search distribution is then expressed as:

$$J(\theta) = \mathbb{E}_\theta[f(z)] = \int f(z)\pi(z|\theta)dz \tag{2.7}$$

The gradient with respect to the parameters can be written as:

$$\nabla_\theta J(\theta) = \nabla_\theta \int f(z)\pi(z|\theta)dz = \mathbb{E}_\theta[f(z)\nabla_\theta log\pi(z|\theta)] \tag{2.8}$$

from which a Monte Carlo estimate is obtained

$$\nabla_\theta J(\theta) \approx \frac{1}{n}\sum_{i=1}^{n} f(z_i)\nabla_\theta log\pi(z_i|\theta) \tag{2.9}$$

of the search gradient. The key step then consists of replacing this gradient by the natural gradient

$$\widetilde{\nabla}_\theta J = F^{-1}\nabla_\theta J(\theta), \tag{2.10}$$

where $F = \mathbb{E}[\nabla_\theta log\pi(z|\theta)\nabla_\theta log\pi(z|\theta)^T]$ is the Fisher information matrix. The search distribution parameters are iteratively updated using natural gradient ascent

$$\theta \leftarrow \theta + \eta\widetilde{\nabla}_\theta J = \theta + \eta F^{-1}\nabla_\theta J(\theta) \tag{2.11}$$

with the learning rate parameter, η.

2.10.5 Separable NES (SNES)

SNES (Separable Natural Selection Strategies) is a univariate version of NES [91]. SNES, which was introduced in 2011, is an instantiation of the NES family which was designed to minimise complexity and to optimize for the high dimensionality problem. Stollenga et al. [99] stated that even though SNES might be less powerful than NES it is better in terms of efficiency. SNES employs only Gaussian distribution with the diagonal covariance matrix for the search distribution - instead of using the full covariance matrix parameterization; this corresponds to the following search distribution:

$$p(z|\theta) = \prod_{i=1}^{d} \widetilde{p}(z_i|\theta_i) \tag{2.12}$$

where $\widetilde{p}$ is a family of densities on the reals, and $\theta = (\theta_1, ..., \theta_d)$ collect the parameters of all of these distributions.

Algorithm 1: SNES (Algorithm source [91])

input: $f, \mu_{init}, \sigma_{init}$

repeat

  for $k = 1....n$ **do**

    draw sample $s_k \sim \mathcal{N}(0, \mathbb{I})$

    $z_k \leftarrow \mu + \sigma s_k$

    evaluate the fitness $f(z_k)$

  end

  sort $\{(s_k, z_k)\}$with respect to $f(z_k)$ and assign utilities u_k to each sample

  compute gradients $\nabla_\mu J \leftarrow \sum_{k=1}^{n} u_k.S_k$

  $\nabla_\sigma J \leftarrow \sum_{k=1}^{n} u_k.(S_k^2 - 1)$

  update parameters $\mu \leftarrow \mu + \eta_\mu.\sigma.\nabla_\mu J$

  $\sigma \leftarrow \sigma.exp(\frac{\eta_\sigma}{2}.\nabla_\sigma J)$

until *stopping criterion is met*;

2.11 Summary

Most of emotion detection within text approaches rely on a limited number of emotion keywords or a word lexicons. Therefore, we are looking forward to exploit the word embeddings in emotion detection from text, in order to avoid relying on emotion keywords or a word lexicons. In the existence of word embeddings which transform words into vectorial representations, the idea of distance calculations between two documents become accomplishable. We have not come across any study that has adapted WMD in order to detect emotions from text. Thus, we seek to adapt the distance calculation methods by exploiting the word embeddings to classify the emo-

tion type in short text. In terms of informal text, we have not yet come across any study that has proposed an unsupervised method for detecting emotions laden in short informal text (tweets). Thus, this has been a motivation in part of our study. Furthermore, in the domain of supervised approaches we have not come across any study that used Evolution Strategies to optimise vectors or identify words in word embeddings space that can represent emotion categories in order to detect emotion from short text (tweets precisely).

Furthermore, as we were intending to optimise vectors which can represent emotion categories in the word embeddings space, we thought about using black-box optimisation methods in order to search through a high-dimensional search space without using gradient information. We think it is the best option to start with for these two reasons: the ease of use and it saves computation time as there is no need to calculate the gradient. ES (Evolution Strategies) is a popular type of black-box optimisation. Chapter 4 shows how we have utilised a number of ES algorithms for optimisation purposes. We decided to use both CMA_ES and SNES for three reasons. First, both of them are among the derivative-free algorithms which is a practical choice for continuous optimisation problems. Secondly, they differ in the type of distribution they employ: CMA_ES uses a full covariance matrix while SNES uses only a diagonal of the covariance matrix - for the search distribution. Thus, we can compare the results of both and find out which algorithm is the best for our experiments. Finally, SNES is more generic and more stable in higher dimensions while on the other hand CMA-ES is considered to be more robust to noise.

Chapter 3

Emotion detection using WMD

3.1 Introduction

Mobile devices have now become prevalent given the low cost of production and variety of types and brands. Such devices have led to a range of applications being developed including social media applications. Social media applications have taken off in the last few years with millions of users communicating online and using such applications as a means to stay in touch with family and friends [61], but also to express opinions, thoughts and emotions. The proliferation of text and unstructured data has given rise to the term "big data". People use social media postings to express emotions so being able to recognise and identify emotions automatically has become not just desirable but necessary for a range of applications [112]. For instance, companies mostly are aiming to understand customers' reactions to products automatically from reviews or postings. Thus, textual emotion detection has become a topic of interest and a challenging issue within the subject areas of Natural Language Processing (NLP) and Adaptive Computing (AD) over the last few years.

Many approaches have been proposed towards Sentiment Analysis. Sentiment Analysis is a computational technique used to identify the polarity of a review in terms of three aspects: positive, negative and neutral [37]. Moreover, many other approaches have been proposed which go beyond that - to the detection of the emotional category most represented in the text. Some of these approaches have been mentioned in

Chapter 2 Section 2.5. However, we have also identified the fact that most of these approaches relied on a limited number of emotion keywords or a word lexicon and number of other such approaches were focused only on specific domain/s and tend to be inadequate in terms of being generalised to other domains.

When detecting emotions from textual data, the only data which is available (generally) is the text itself, and this makes the task particularly intractable. The word vector representation is one of the developed techniques proposed for approaching such problems. Word2Vec [66] is a popular word embedding predictive model. It can be argued that word vectorization with representations for semantic relationships would be useful for improving the effectiveness of NLP applications [67].

The current availability of vast amounts of both online and locally-stored data may well augment our understanding of emotions via studies of scale previously unthinkable. Moreover, during the last decade a neural-network methodology has been developed to transform words into their vectoral representations [67]. Using the results of such transformations, it is possible to calculate the distance between emotion-words and general text [57].

Our work was focused on textual emotion detection using the word embedding approach. In particular, we have been motivated to the Word Mover's Distance (WMD) technique [57] as it was a promising technique. Unlike other approaches, training data labeling, ontologies and term extractors were not required for this approach. To the best of our knowledge, this work presents the first investigation into using Word Mover's Distance to detect emotions within text.

As part of this work, two main frameworks were proposed which aimed at identifying emotions from news headlines. The first one was constructed using the WordNet dictionary and the NLTK (Natural Language Toolkit). The other approach was built mainly on the basis of Word Mover's Distance (WMD) and Word2Vec. These two approaches will be described further below.

The remainder of this chapter is organised as follows. In the next section (section 3.2), the dataset will be described. The methodologies employed will be presented

in section (3.3). Section (3.4) will provide a description of the experiments, and the results will be presented in section (3.5). Section (3.6) will provide a discussion on different work presented in this chapter. The final section (section 3.7) presents a summary of the chapter.

3.2 Dataset

We chose to work on the dataset from [102] for many reasons. First, this dataset was annotated according to the Ekman basic six emotions. Second, its instances were news headlines; these were typically written in emotional language and mainly to intrigue people. Third, the database was constructed using texts written in formal language which is compatible with Google's pre-trained Word2Vec embedding system which was trained on the google news dataset. Finally, the structure of the news headlines was more appropriate since our goal was to conduct experiments on short texts.

Strapparava and Mihalcea [101] employed the same dataset as was developed for SemEval 2007 [102]. This dataset is comprised of news headlines gathered from newspapers and news websites. CNN, the New York Times, BBC News and Google News were the major sources from which the data was extracted. The data were divided into two sets: development data and test data. Across these two, a total of 1250 annotated headlines were included - the former consisting of 250 instances while the latter included 1000 annotated headlines [101].

Ekman's six basic emotions were used to annotate the dataset, and each headline was associated with seven slide bars: six slide bars for the emotions and the last slide bar for the valence. Each emotion label ranged from 0 to 100 with 0 implying that the emotion is missing in this headline and 100 denoting that the given headline is replete with this emotion. Based on the implicit feeling of the text or on the occurrence within it of emotional phrases or words, six annotators were asked to tag the headlines with one or more appropriate emotions [101].

3.3 WMD-ED approach

This approach is called Word Mover's Distance Emotion Detection (WMD-ED). The approach WMD-ED employed here was to adapt the Word2Vec [66] as well as the WMD [57] methods in order to detect emotions in news headlines from the same dataset as used for the benchmark in [102]. By taking advantage of the embedded words of each headline and the emotion words which could be found in the WordNet-Affect [103] emotion dictionary, the dissimilarities between these two types of words were derived, a pictorial representation of this can be seen in Figure 3.1. For each news headline in the dataset, stop words such as ("the" and "to") were first removed; this was for two main reasons. First, so that the ground truth follows the same pre-processing as the benchmark. Secondly, and more importantly, because such words contribute little to the content of a sentence especially as regards emotion. The WMD distance between the sentence and each word in the emotion lists were then calculated. The closest emotion to the headline's overall emotion was the emotion which was closest in terms of this distance. The distance was calculated by measuring the minimal cumulative distance of all the words in the news headline that must travel in order to match the emotion words. The results yielded by this phase can be compared to the predetermined annotations and the results from the approaches in [101]. Furthermore, a comparison can be carried out with the WordNet Affect experiment attempted below.

3.4 Experiments

3.4.1 Word Net Affect experiment

A preliminary experiment was carried out using Python, NLTK and the six lists of emotion words from the WordNet-Affect dictionary (in the Synset format). The benchmark dataset of news headlines from [102] was used to evaluate the experiment. The coarse-grained predefined emotion annotations [101] mentioned above were compared with the results by selecting only the highest emotion score as the label for

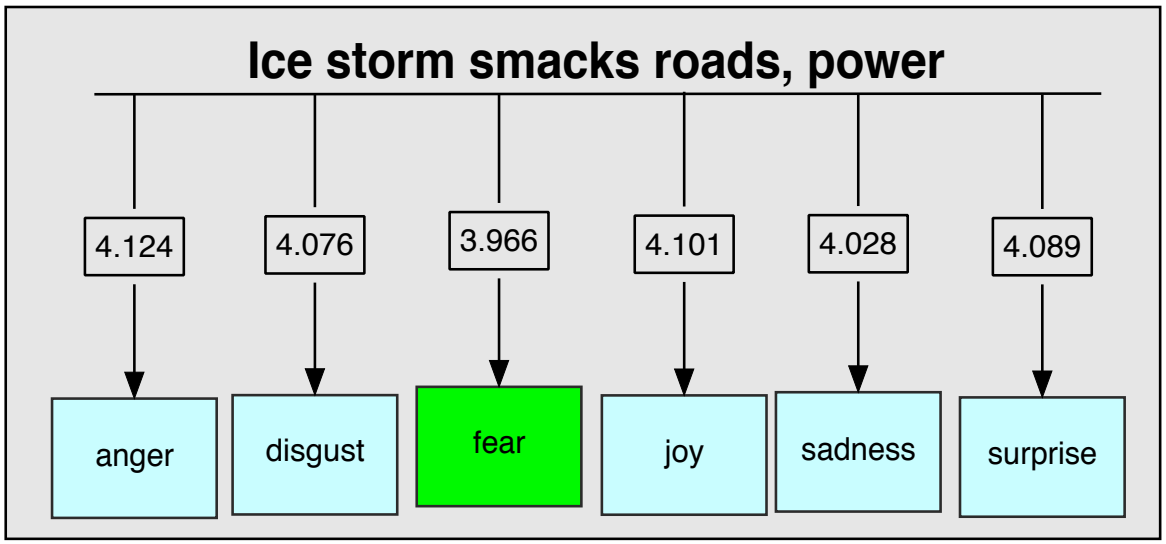

Figure 3.1: An example of WMD distances calculated directly for the six different emotions - the algorithm relates the sentence to "fear" - the word with the shortest distance from the sentence.

each headline. However, as stated in [112], people differ in their emotional reactions to similar events. Thus, in this subsequent experiment the benchmark's baseline [101] was examined in a different light by selecting the two highest emotion scores.

After the pre-processing and tokenization phases were completed, the frequency of occurrence of each emotion in a sentence was calculated. The next step was to do the processing required to maintain a list of emotion counts for each sentence separately. The results were then passed via an array in order to calculate the precision, recall, and F1-score. Following this, in order to measure the classification accuracy for each emotion category, the F1 score was calculated as defined in equation 3.1. The embedding model adapted for use in this experiment was Google's pre-trained model, described in the methodology section above. This result can then be compared with the results of the approaches shown in [101].

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{FN + TP} \\ F1 &= 2 * \frac{precision * recall}{precision + recall} \end{aligned} \tag{3.1}$$

- TP: true positives or the number of tweets correctly classified as belonging to an emotion category.
- FP: false positives or the number of tweets incorrectly classified as belonging to an emotion category.
- FN: false negatives or the number of tweets incorrectly classified as not belonging to an emotion category.

3.4.2 WMD-ED experiment 1

After the preliminary experiment above was carried out, we wanted to perform our own experiments. Thus, the procedure described in the WMD-ED approach (section 3.3) was followed in order to detect emotions in headlines, using basic emotion words only: anger, disgust, fear, joy, sadness, and surprise. Using these basic seed words only, nearly all of the headlines in this experiment were classified as mostly expressing either surprise or fear; this will be discussed in the results next. Subsequently, the following procedure was adopted.

3.4.3 WMD-ED experiment 2

The above experiment did not give a good results by using only one word to represent the emotion category. While, the news headlines consisted of multiple words; therefor, it came the idea of using several words to represent each emotion category in order to calculate the distance between the emotion categories and the news headlines. In this experiment, the same WMD-ED approach was employed but the words used were from six lists (one for each emotion category) of four seed words each were prepared: for example, joy = [*joy delight triumphantly ebullient*]. The results yielded by this experiment were much better than those yielded by the first. The emotions detected were spread out over all six categories. In addition, the F1 score for some emotion categories was higher than the best that was achieved by the benchmark results [101] - while for others it was slightly lower.

To gain better results across all the emotion categories the following random seed word selection was performed. All the emotion words for each emotion category were retrieved from WordNet Affect and all were stored in separate lists. A random selection function was performed which took, as parameter, the number of seed words required and the number of iterations to be used. The above experiment was re-run, but with this inclusion, and then in the emotion categories for which the F1 score achieved was better than the benchmark [101], the seed word lists were retained. The lowest F1 scoring category was then caused to use a number of further iterations, and for each iteration random seed words were selected. The WMD-ED approach as described above was then followed and the F1 score was then re-calculated. In consequence, the best seed words for the emotion category were found. The same process was repeated for the (next) current lowest F1 scoring category until all the categories' scores exceeded the benchmark results. Table 3.1 contained the emotions seed words that achieved the best results, as will be shown in the results section (3.5). The seed words were not stemmed or lemmatised because the Google's pre-trained Word2Vec embedding (section 2.8.3) did not mention any stemming or lemmatisation in their training process [66] [67]. Furthermore, as the seed words were randomly selected there might be seed words repetition in a few cases because of two reasons: First, the basic seed words were forced to be included in some experiments. Second, the words in Word2Vec were not stemmed or lemmatised. However, from different experiments performed, it has been noticed that the duplication of any seed word did not have an impact on the results.

3.4.4 WMD-ED Experiments 3a and 3b

As the above experiment gave a good results while the basic seed words were forced to be included in the selected seed words. It worth to demonstrate, which one is the best to include the basic seed words among the selected seed word or not ?. The following two subsections will discuss this in more detail.

Emotion category	Seed words			
Anger	anger	pique	annoyance	annoy
Disgust	disgust	loathing	odium	detestation
Fear	fear	terror	panic	scare
Joy	joy	delight	triumphantly	ebullient
Sadness	sadness	misery	mournful	joylessly
Surprise	surprise	daze	wonder	startle

Table 3.1: The emotion seed words that achieved the best results

WMD-ED Experiment 3a

This experiment was designed to randomly select seed words for all of the emotion categories, using 30 iterations. In this experiment, the basic emotion words (*anger, fear, disgust, joy sadness* and *surprise*) were used in every iteration and the one randomly selected seed was added to them for each category. Thus, the first experiment, was represented by two seed words: the basic emotion word + one randomly retrieved seed word. Then, other experiments were run using three, four, five, six and seven seed words for each category, with the basic word included in each, and for 30 iterations each. For each iteration, the WMD-ED approach was adopted, and the F1 score calculated for every emotion category.

WMD-ED Experiment 3b

Another experiment that followed on from this was to randomly select seed words for all of the emotion categories for 30 iterations. In this experiment, all the words were randomly selected and no seed words forced to be included. This meant that the process started with two seed words then it was run with three, four, five, six and seven seed words - for 30 iterations. For each iteration, the WMD-ED approach was applied, and the F1 score was calculated for every emotion category.

3.4.5 Bootstrapping

As the experiment 3.4.3 gave good results , it came to mind the idea of trying to run the same experiment by using the bootstrapping. This experiment was designed to randomly select seed words for all of the emotion categories, using 100 iterations . In this experiment, the basic emotion words (*anger*, *fear*, *disgust*, *joy*, *sadness* and *surprise*) were used in every iteration and the three randomly selected seed were added to them for each category. Thus, this experiment, was represented by four seed words: the basic emotion word + three randomly retrieved seed word. For each iteration, the WMD-ED approach was adopted to calculate the distance between the headlines and the emotional words for every emotion category. Then the distance results were stored in lists. The mean and the minimum (in separate calculation) of the distances for each headline sentence and each emotion category was calculated from the lists and stored in a new lists. The F1 score was calculated for every emotion category using the mean/ minimum results from the 100 iteration bootstrapping. The results will be shown in the results section (3.5).

3.5 results

3.5.1 Word Net Affect experiment

Table 3.2 illustrated the results of the initial experiments and Table 3.3 showed the overall averages. It can be observed from the overall average that only the highest yielded F1 score is better than the benchmark's baseline results [101]. The reason for this better result was that the benchmark's approach [101] was using specific components of Word Net Affect [101]; while the whole updated version of Word Net Affect was used in this experiment. However, this was an obvious control example which showed the limitations of using dictionaries/lexicons. Table 3.3 illustrated that

F1 score	anger	disgust	fear	joy	sadness	surprise
WN-affect presence[1]	6.06	-	3.33	1.1	6.61	6.9
highest value only	14.28	0	24.56	8.48	6.95	5.88
highest 2 values excluding 0	3.52	0	11.17	10.13	4.12	6.69

Table 3.2: Performance of the Word Net Affect experiments

Overall averages		WN-affect presence[1]	highest value only	highest 2 values excluding 0
prec.	Average	38.28	24.95	56.53
	SD	34.92	23.20	32.96
	Sample size (N)	6	6	6
Rec.	Average	1.54	6.59	3.17
	SD	1.85	5.24	2.31
	Sample size (N)	6	6	6
F1	Average	4	10.03	5.94
	SD	2.50	8.47	4.24
	Sample size (N)	6	6	6

Table 3.3: Overall average results

the highest two values did not give any better results.

3.5.2 WMD-ED Experiment 1

Using the WMD-ED approach (section 3.3) to calculate the distance between the news headlines and the emotions represented by basic seed words only, it can be seen, from table 3.4, that the detection of two emotions surprise and fear dominated in 997 headlines, while the distribution as defined by the benchmark [102] for the coarse-grained evaluation is that anger: 21, disgust: 12, fear: 92, joy: 113, sadness: 104 and surprise: 42. Of course, this result includes both correct and incorrect classifications. However, the major reason behind this limited result is that the WMD distance, as described in (section 2.9), computed the distance between two text documents

[1]benchmark baseline results [101]

while in this case the emotion-expressing 'document' that contained the emotion seeds only contained one word. In addition, when we compared the distance between the headline and all the six categories of emotions, only the emotion (of the six) which had the shortest distance from the text was selected to be the emotion category for the headline. This yielded results which was unsatisfactory. Thus, the following procedure was adopted to utilise more seed words for each emotion category.

emotion	number of headlines classified
anger	0
disgust	1
fear	533
joy	2
sadness	0
surprise	464

Table 3.4: WMD-ED Experiment 1 results - including correct and incorrect classification

3.5.3 WMD-ED Experiment 2

For general comparison the average F1 score across all six emotions yielded by our system was 22.01. This shows that it was the best over all as compared to the other systems mentioned in the benchmark [101] - including the three SEMEVAL systems (SWAT, UA and UPAR7) see Table 3.5 and our Word Net Affect experiment Table 3.3. Moreover, in comparison with the results achieved by the five approaches developed by Strapparava and Mihalcea [101], it was found that our approach obtained the best F1 scores in relation to all the emotion lists, see Table 3.6. In terms of the benchmark systems, in contrast, not one of them achieved the best F1 scores across the majority

of the emotion categories. Moreover, in comparison with the SEMEVAL systems (SWAT, UA, and UPAR7), our approach was the best overall with the exception of the sadness F1 score in UPAR7, which was 30.38 Table 3.6.

Approach	Average of F1 scores
WN-affect presence	4
LSA single word	16.37
LSA emotion synset	13.38
LSA all emotion words	17.57
NB trained on blogs	13.22
SWAT	11.57
UA	9.51
UPAR7	8.71
WMD-ED Experiment 2	**22.01**

Table 3.5: WMD-ED Experiment 2 overall F1 average results

The seed words (from table 3.1) that gave the best results were here projected using Google's pre-trained Word2Vec embeddings into two-dimensional space, see Figure 3.2. For this projection we used Principal Component Analysis (PCA) which reduces the dimensionality of the word embeddings in order for them to be presented in two-dimensional space. From Figure 3.2, it can be seen that the joy and sadness embedding words were overlapped in the top left corner of the vector space. Also, the words corresponding to anger were not close to the embedding vector of the word "anger". Moreover, the words representing disgust embeddings were placed at the bottom left corner of the vector space but the embedding of word for anger takes a

F1 score	anger	disgust	fear	joy	sadness	surprise
WN-affect presence	6.06	-	3.33	1.10	6.61	6.90
LSA single word	11.43	4.68	22.80	25.88	21.20	12.23
LSA emotion synset	13.45	3.00	22.00	30.55	23.06	13.38
LSA all emotion words	11.58	3.87	21.91	30.83	20.61	14.10
NB trained on blogs	16.77	-	5.63	32.87	21.43	2.63
SWAT	7.06	-	18.27	14.91	17.44	11.78
UA	16.03	-	20.06	4.21	1.76	15.00
UPAR7	3.02	-	4.72	11.87	**30.38**	2.27
WMD-ED Experiment 2	**18.75**	**7.69**	**27.67**	**34.50**	24.06	**19.39**

Table 3.6: WMD-ED Experiment 2 F1 score results for each system for every emotion category

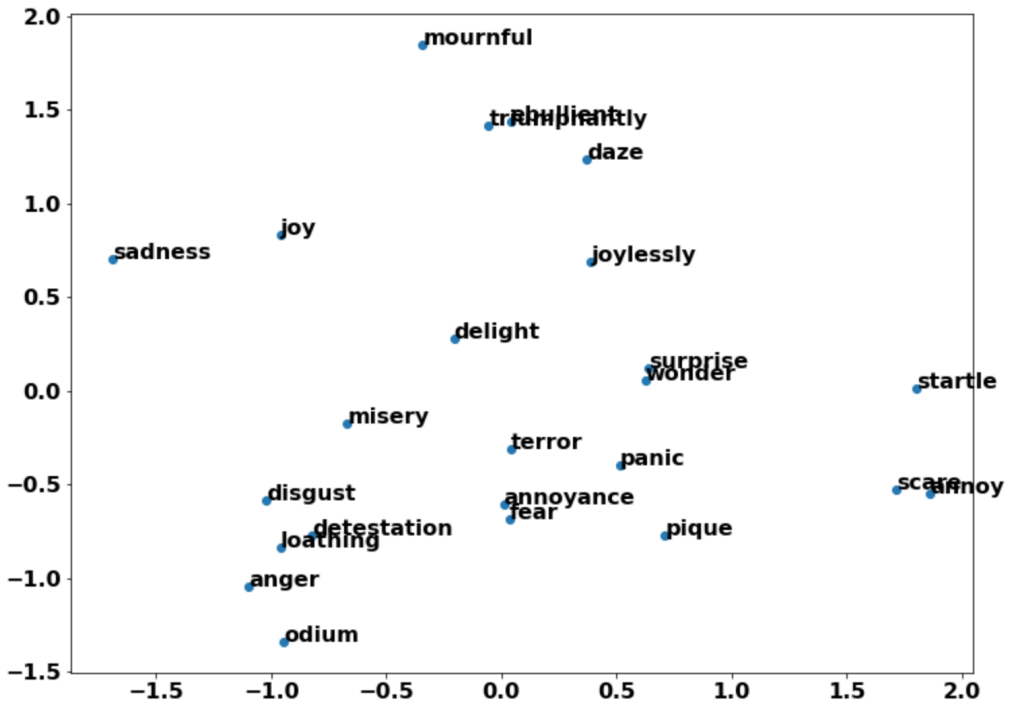

Figure 3.2: PCA of WMD-ED Experiment 2 seed words)

place in between.

Recent results from SEMEVAL

Chatzakou, Vakali, and Kafetsios [16] performed machine learning and lexicon-based approach on the same dataset which was published in December 2017. While, the approach initially proceeded with lexicon-based to extract two features: sentimental and emotional. Then, machine learning approach followed to detect emotions. The lexicon-based in this approach was either the WordNet-Affect (WN) alone or combined with the EmoLex corpus (WN-EL). Table (3.7) illustrated a comparison between these two approaches and our novel experiment results in this section (WMD-ES Experiment 2). It can be seen that our results for two categories *joy* and *surprise* exceeded their result; while two categories *anger* and *fear* were close to their best results in WN. Even though our results in the remaining two categories were less than their results but our results were published in October 2017 [5] which is before their publication.

F1 score	anger	disgust	fear	joy	sadness	surprise
WMD-ED Experiment 2	18.75	7.69	27.67	34.50	24.06	19.39
WN [16]	18.87	29.41	28.26	21.33	38.67	6.73
WN-EL [16]	16.84	10.57	15.20	16.67	31.18	6.76

Table 3.7: WMD-ED Experiment 2 F1 score results compared with recent results from [16]

3.5.4 WMD-ED Experiments 3a and 3b

WMD-ED Experiments 3a and 3b results

Boxplot visualization has been plotted for each emotion category and for both WMD-ED Experiments, 3a and 3b separately. These boxplots presented a comparison between the F1 scores obtained from a different number of random seed words. Figures (3.3, 3.4, 3.5, 3.6, 3.7 and 3.8) showed results from WMD-ED Experiment 3a while Figures (3.9, 3.10, 3.11, 3.12, 3.13 and 3.14) illustrated results from WMD-ED Experiment 3b. From Figures (3.9, 3.10, 3.11, 3.12, 3.13 and 3.14) it can be seen that, mostly, the F1 scores start from zero, while Figures (3.3, 3.4, 3.5, 3.6, 3.7 and 3.8) showed that the F1 scores obtained with the inclusion of the basic words, in many cases, starts with a number higher than zero. It is also noticeable from Figures (3.3, 3.4, 3.5, 3.6, 3.7 and 3.8) that WMD-ED Experiment 3a gave more consistent results because it has a low standard deviation from an overall point of view. Figures (3.9, 3.10, 3.11, 3.12, 3.13 and 3.14) showed that when the basic emotion words were not used, the standard deviation was higher and the F1 scores were spread out over a wider range. Moreover, the box plots indicated that 4 and 5 can be optimal numbers of seed words to be used in general.

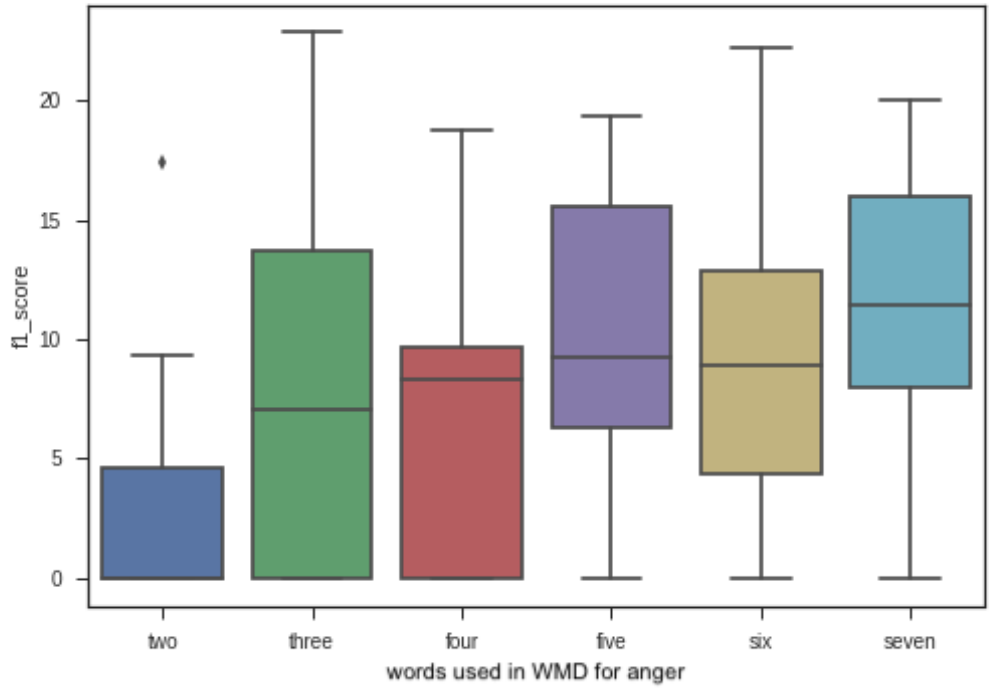

Figure 3.3: WMD-ED Experiment 3a for anger - with basic word "anger" included in the seed words

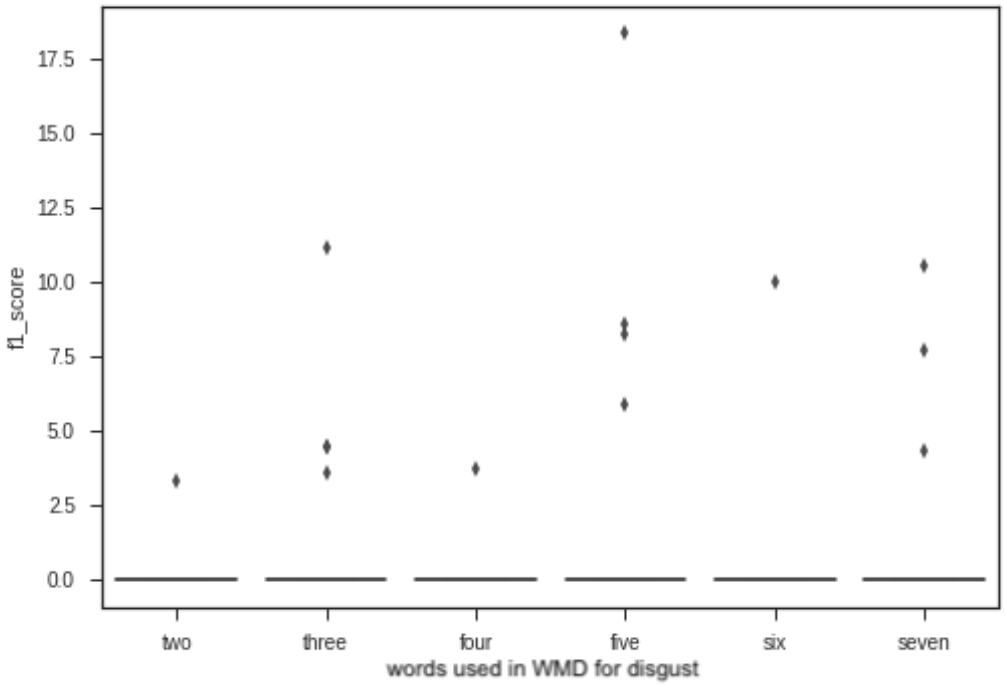

Figure 3.4: WMD-ED Experiment 3a for disgust - with basic word "disgust" included in the seed words

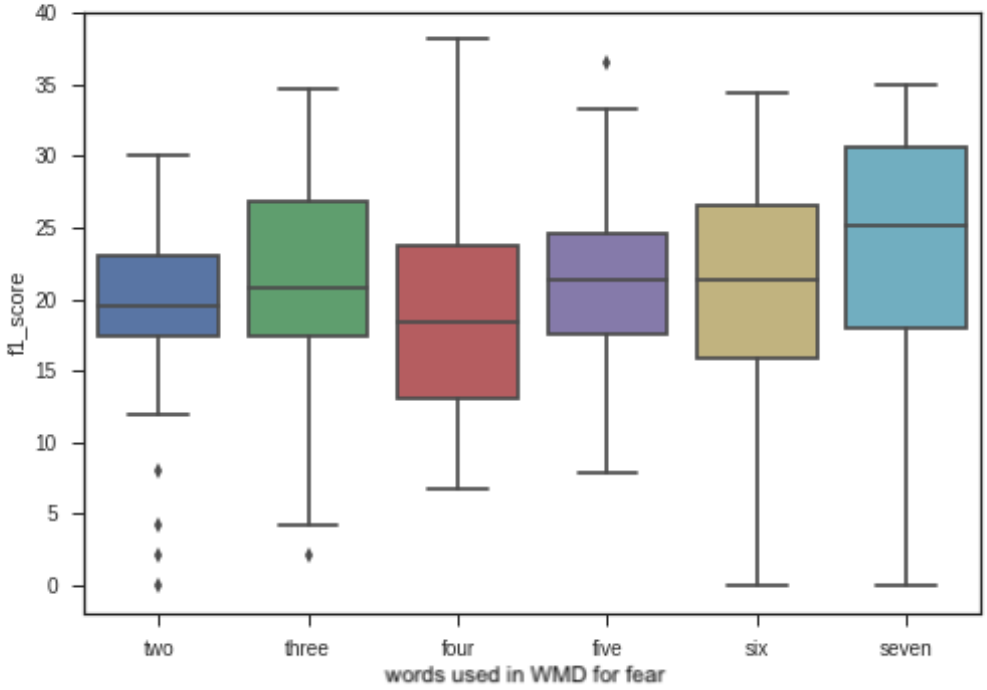

Figure 3.5: WMD-ED Experiment 3a for fear - with basic word "fear" included in the seed words

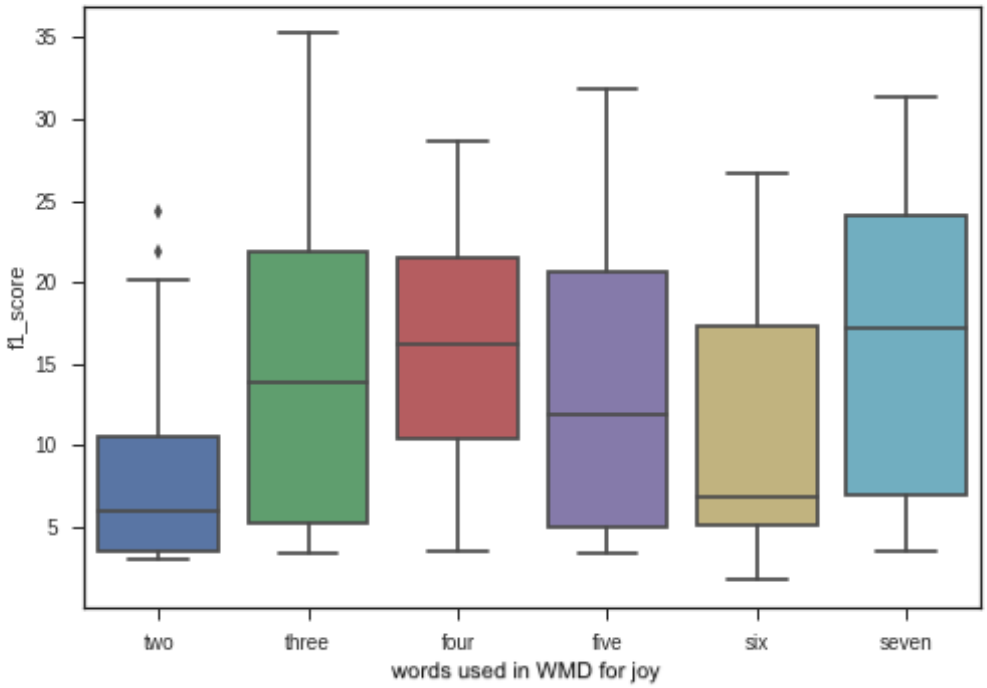

Figure 3.6: WMD-ED Experiment 3a for joy - with basic word "joy" included in the seed words

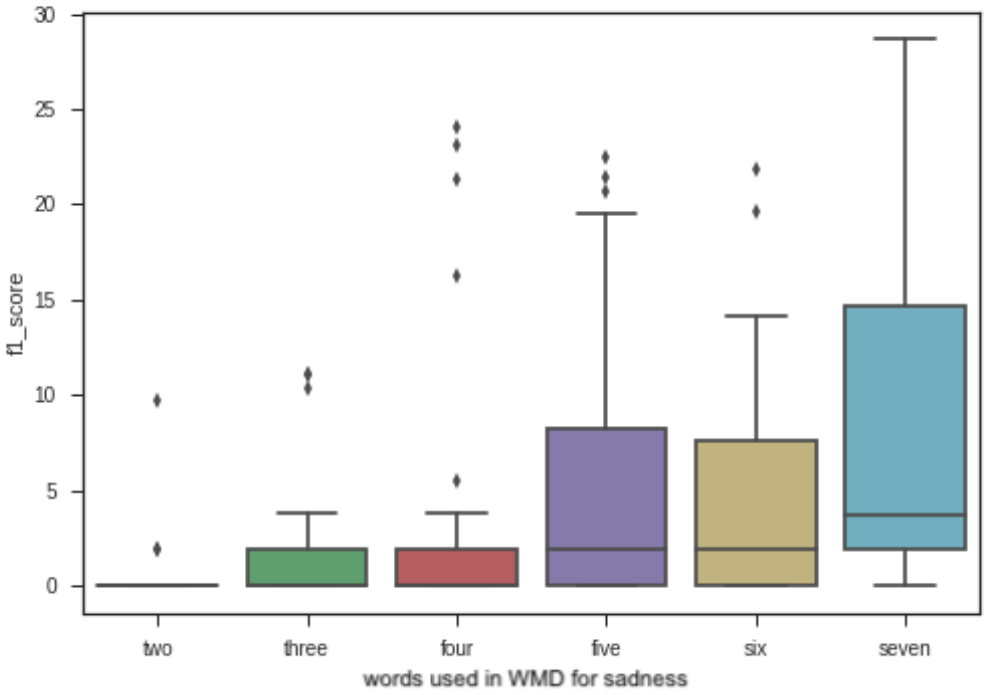

Figure 3.7: WMD-ED Experiment 3a for sadness - with basic word "sadness" included in the seed words

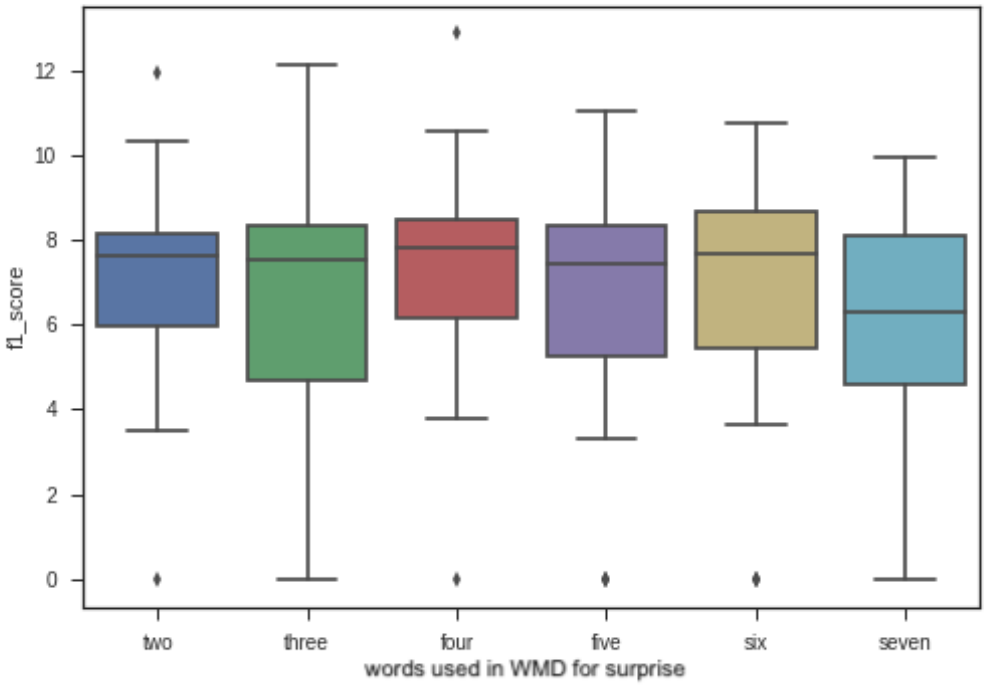

Figure 3.8: WMD-ED Experiment 3a for surprise - with basic word "surprise" included in the seed words

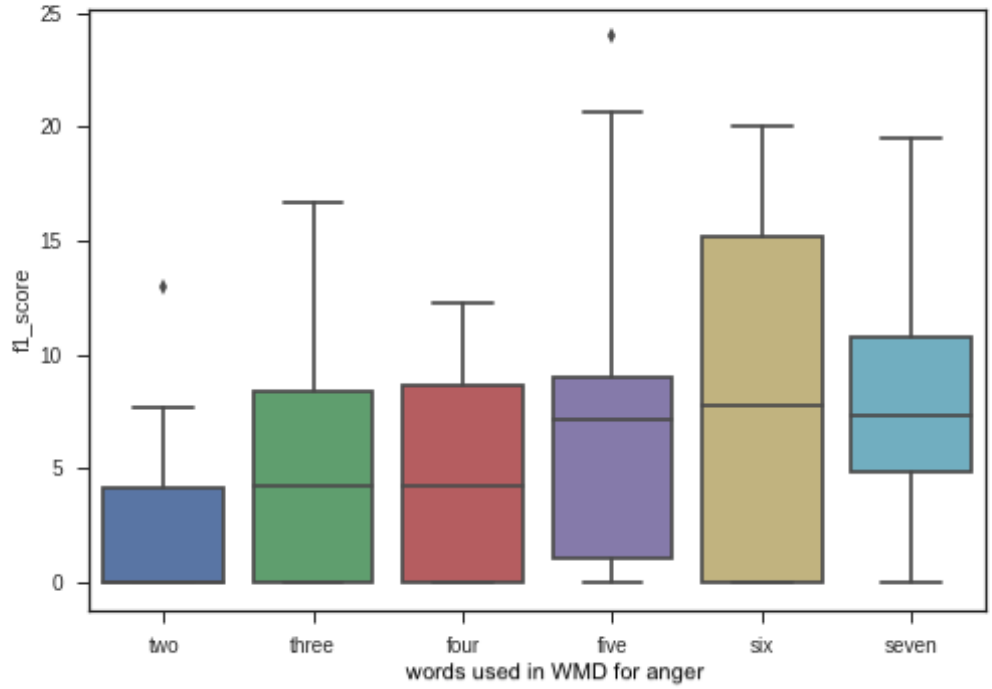

Figure 3.9: WMD-ED Experiment 3b - basic word "anger" is deliberately not included in the seed words

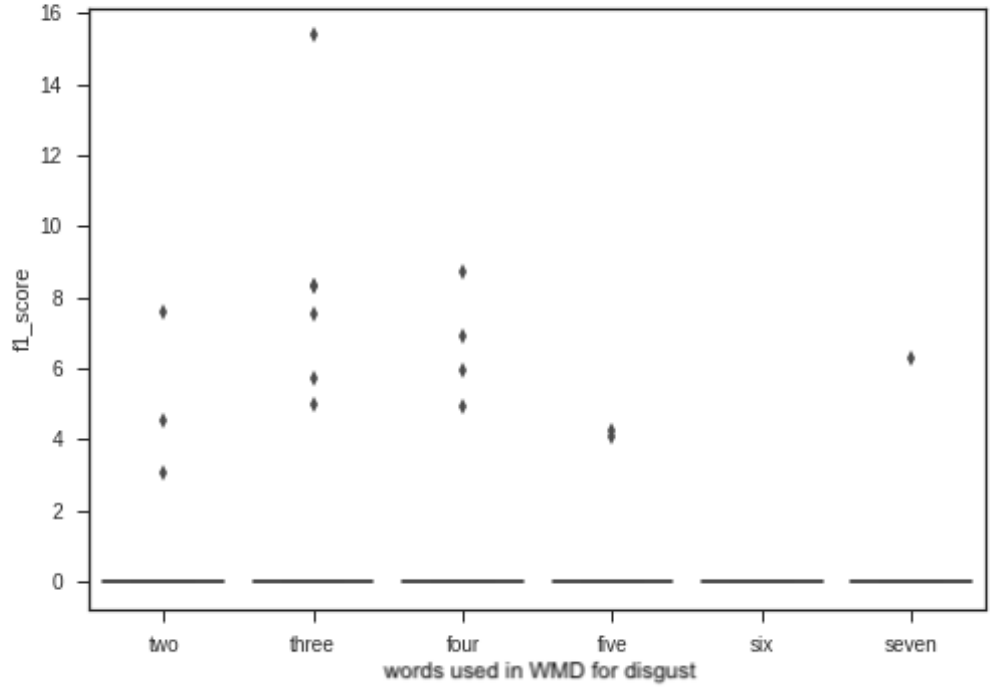

Figure 3.10: WMD-ED Experiment 3b - basic word "disgust" is not deliberately included in the seed words

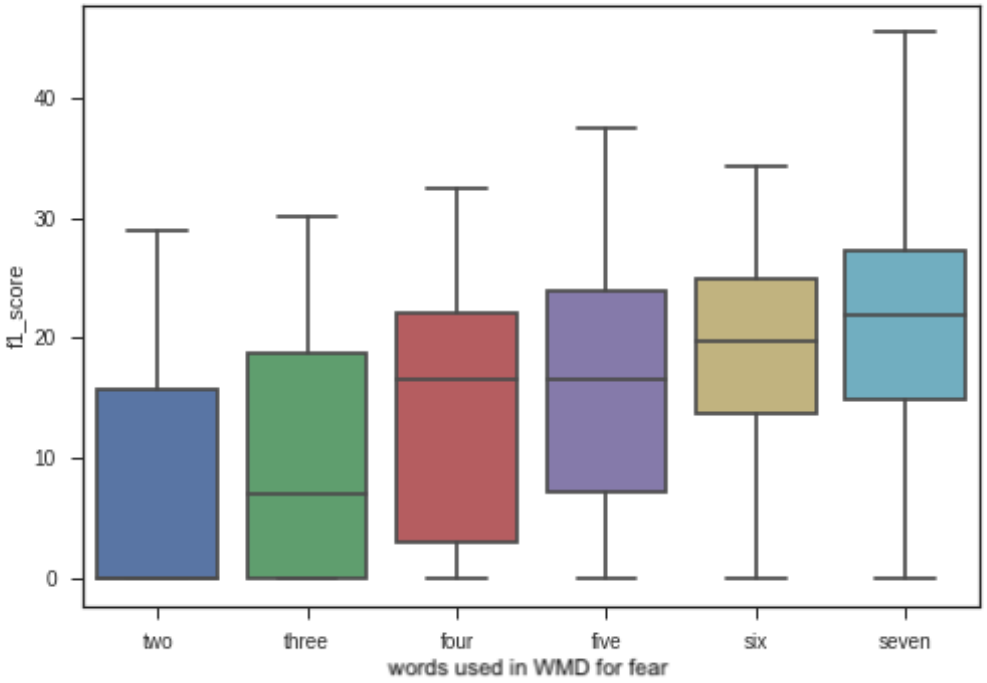

Figure 3.11: WMD-ED Experiment 3b - basic word "fear" is not deliberately included in the seed words

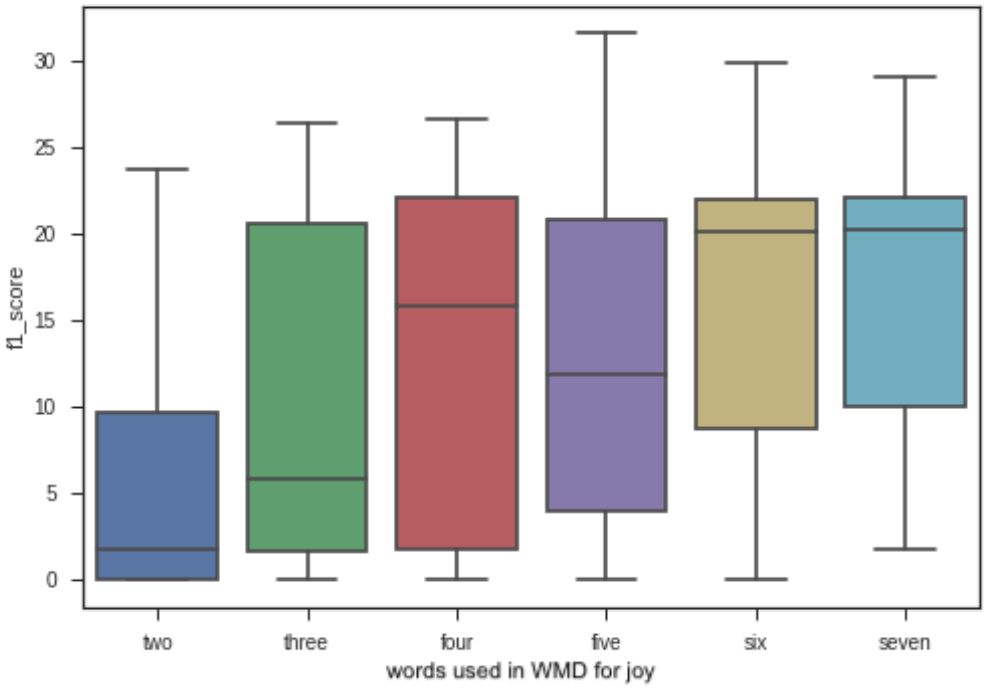

Figure 3.12: WMD-ED Experiment 3b - basic word "joy" is not deliberately included in the seed words

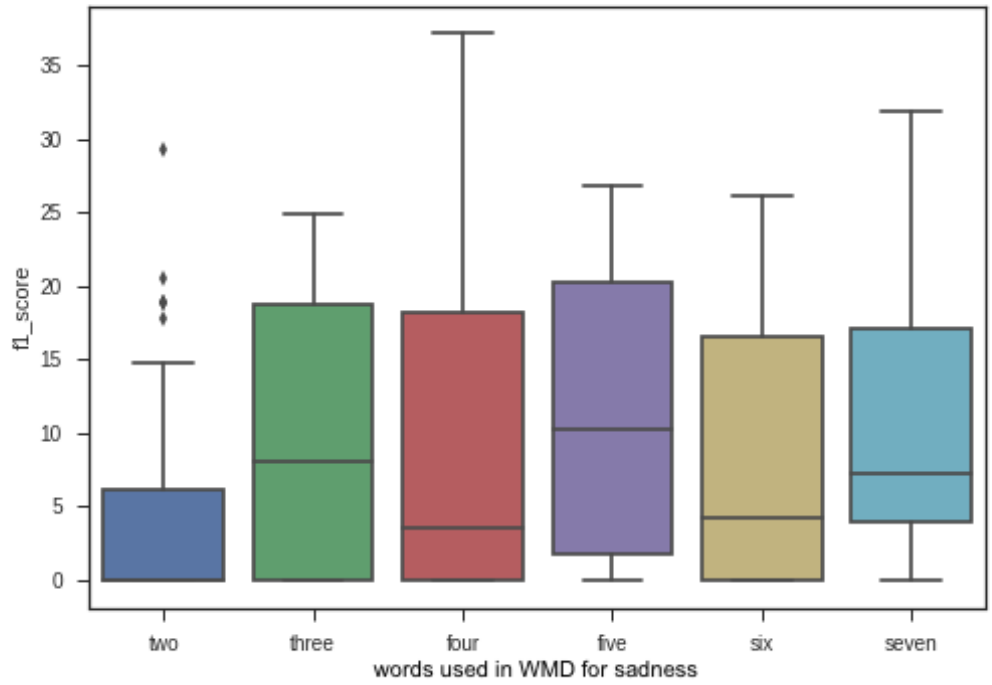

Figure 3.13: WMD-ED Experiment 3b - basic word "sadness" is deliberately not included in the seed words

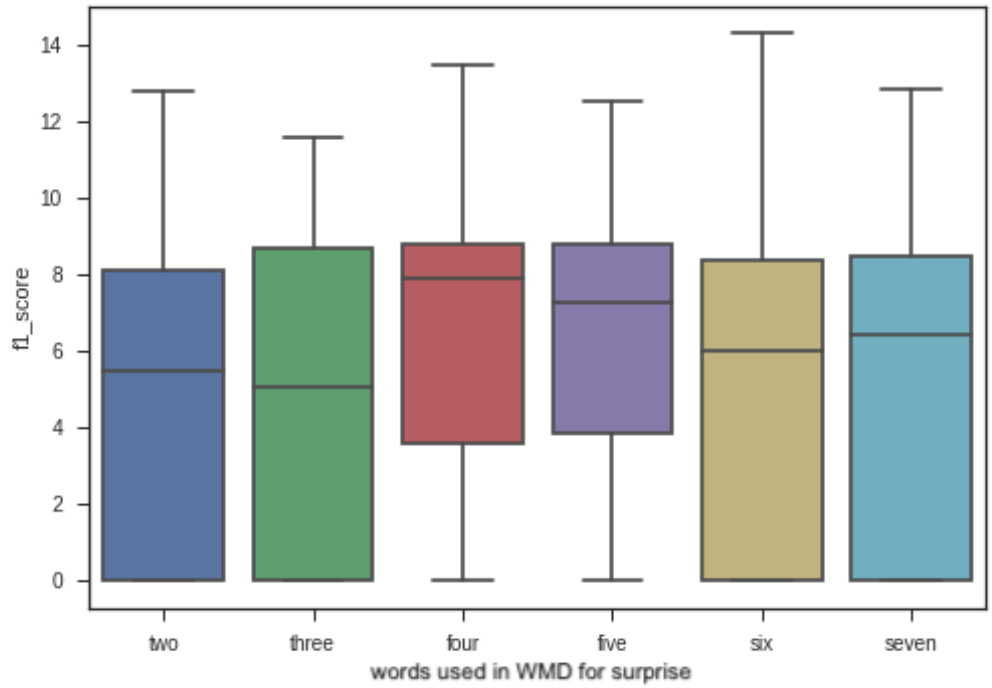

Figure 3.14: WMD-ED Experiment 3b - basic word, "surprise" is deliberately not included in the seed words

F1 score	WMD-ED Experiment 2 (3.6)	Min	Mean
anger	18.75	18.46	17.39
disgust	7.69	0.0	0.0
fear	27.67	28.68	21.47
joy	34.50	17.08	10.52
sadness	24.06	19.31	1.90
surprise	19.39	8.0	5.80

Table 3.8: Results obtained using the bootstrapping

3.5.5 bootstrapping

Table 3.8 represents the results of using both mean and minimum of the resulted distances. In general, this experiment didn't show any better results in comparison to those we have achieved in WMD-ED Experiment 2 (3.5.3). Furthermore, both mean and minimum did not detect the emotion category disgust. However, it can be noticed that the results using the minimum provided better results than using the mean in all the categories with the exception of disgust where both gained 0 scores.

3.5.6 Surrounding seed words

Emotion	Seed words				
Anger	anger	resentment	fury	outrage	discontent
Disgust	disgust	dismay	exasperation	frustration	indignation
Fear	fear	fearful	fears	fearing	worry
Joy	joy	sheer_joy	exhilaration	unbridled_joy	joyful
Sadness	sadness	sorrow	grief	profound_sadnes	anguish
Surprise	surprise	suprise	surprising	pleasant_surprise	shocker

Table 3.9: The emotion seed words which achieved the best results

```
import gensim
model = gensim.models.Word2Vec.load_word2vec_format('GoogleNews-vectors-negative300.bin', binary=True)

categories = ['anger', 'disgust', 'fear', 'joy', 'sadness', 'surprise']

for e in categories:
    print e
    emo_category = model.most_similar([e], topn=10)

    for each_word in emo_category:
        print each_word[0]
```

Figure 3.15: The python code used to obtain the most similar words

For better insights, it was necessary to try suing the surrounding embedding words related to each category while avoiding overlapping (as in 3.5.3) with other emotion embedding categories as much as possible. In order to obtain the surrounding words, we did not use the word embeddings for the emotional words from Word Net Affect since these were overlapping; instead, we used the most similar words function to find the 10 most similar words from Google's pre-trained Word2Vec embeddings for each category, see figure 3.15. This *most_similar* method (figure 3.15) used here is from the gensim which is a Python library for topic modelling and similarity retrieval [1]. The *most_similar* method computes the cosine similarity between the given word vector (of basic emotion words in this case) and all other words' vectors in the model to yield the chosen number (here, ten) of most similar words. From these ten most similar words, we manually selected the words which were considered to have an appropriate relationship with the emotion category. Figure 3.16 projected the surrounding seed words (see table 3.9) and also showed that the seed words for each category were separated from other categories in the projection space. The same procedure as was used in WMD-ED Experiment 2 was carried out again but using the surrounding seed words from table 3.9 in order to compare the results. Once this was done, the F1 scores yielded by this experiment were laid out in Table 3.10; this clearly presented the finding that utilising surrounding embedding words did not provide better results. It can be seen from table 3.10 that the F1 scores of all categories other than fear were less than ten - which was not even close to the benchmark results [101]. The original WMD-ED Experiment 2 provided much better results.

[1] https://radimrehurek.com/gensim/models/keyedvectors.html#gensim.models.keyedvectors.WordEmbeddingsKeyedVectors.most_similar

Emotion category	F1 scores
anger	9.09
disgust	0.0
fear	25.58
joy	3.47
sadness	1.90
surprise	9.32

Table 3.10: Results obtained using the surrounding seed words

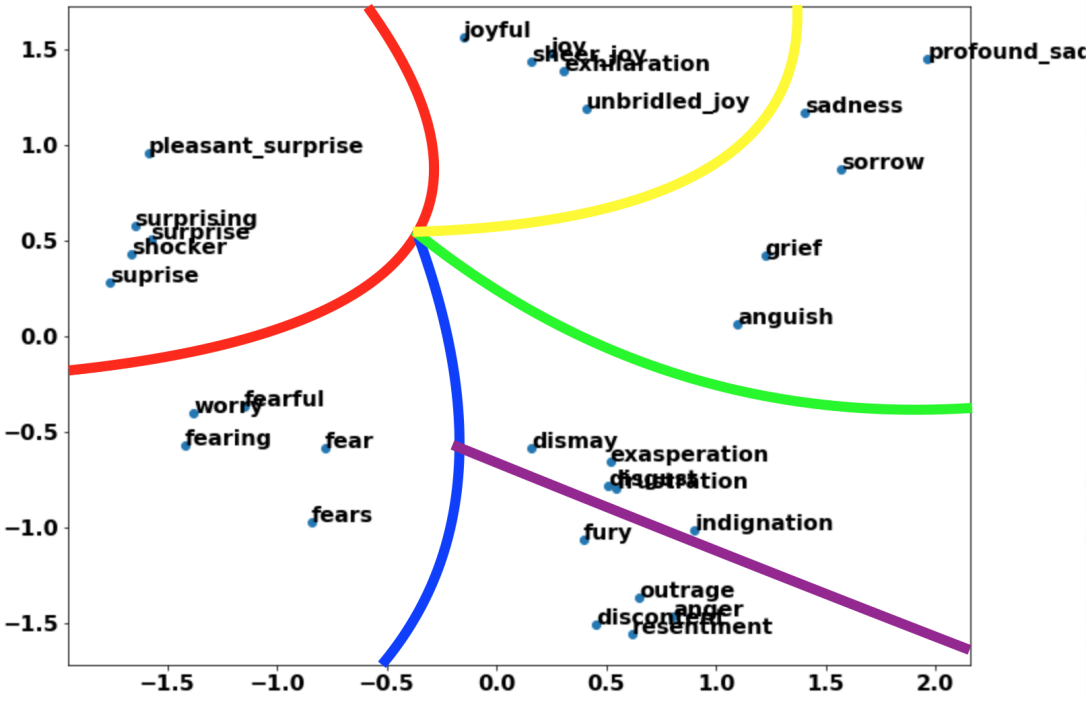

Figure 3.16: The PCA of the surrounding seed words while separated

3.6 Discussion

Our findings demonstrated that, it is possible to detect the emotions from short text by using Word Mover's Distance along side with word embedding by considering that data labeling, ontologies and term extractors are not required. Our results in

WMD-ED Experiment 2 surpassed the benchmark approaches' results [101] across all emotion categories. This model can be used as classification model for other domains as it does not require training phase. The motivation for focusing on short text was because word embedding retrieving and distance will be calculated therefore, short text is better to be used for less computational time. The same approach can be performed on a longer text but it would take more computational time. For future study, we are looking to improve the robustness of this approach by considering to train new Word2Vec model for the domain of emotions.

3.7 Summary

We presented textual emotion detection by using the word embedding approach. A preliminary experiment performed using NLTK and WordNet-Affect dictionary. It was based on the frequency of presence of the emotional words in each sentence. The second experiment, WMD-ED Experiment 1 was using the WMD-ED approach (section 3.3) to calculate the distance between the embeddings of the news headlines and the emotions which represented by the basic seed word only. This experiment did not achieve a good results. More importantly, our novel WMD-ED Experiment 2 was using the WMD-ED approach (section 3.3) to calculate the distance between the embeddings of the news headlines and the emotions which represented by four seed words retrieved from WordNet Affect randomly. The results from our original WMD-ED Experiment 2 exceeded the benchmark's results [101] in general. The approach employed in WMD-ED Experiment 2 achieved the best F1 scores across all the emotion categories while the benchmark systems [101], in contrast, did not achieve the best F1 scores for the majority of these categories.

It was also found that when basic words were included in the seed word lists, as happened in WMD-ED Experiments 3a and 3b, even better results were obtained. Another experiment performed next was to asses the use of bootstrapping using the same approach in our novel WMD-ED Experiment 2. This experiment was based on random selection of seed words for all of the emotion categories and run for 100

iterations. This bootstrapping experiment did not attain and better results.

In conclusion, we believe that using Word Mover's Distance to detect emotions within text has not been done before [5]. Moreover, the achieved results confirmed the value of our novel approach by using of word vectorization (Word2Vec) together with WMD for detecting emotions, as the approach provided optimal results across all the emotion categories. By keeping in mind, the fact that, we have achieved an optimal results in this approach while data labeling, ontologies and term extractors were not required. We will investigate the potential of adopting this approach in different data domains, such as tweets in the next chapter (chapter 4).

Chapter 4

Evolutionary strategies for emotion detection

4.1 Introduction

In the previous chapter (3), a novel results has been achieved for emotion detection from formal text (news headlines) by using the novel approach WMD-ED (section 3.3). In this Chapter, we will follow the same WMD-ED approach or a different approaches such as evolutionary strategies on informal text (tweets) dataset in order to gain an optimal results.

Written text is used as a means of communication in many different ways, and it is often the case that textual communications online involve short informal messages. One of the most common Natural Language Processing (tasks) is that of trying to identify the emotions evoked by a sentence. The NLP methods that attack this problem fall broadly under the joint umbrella of Sentiment Analysis (SA) and emotion detection. Whereas SA is used to identify just positive, neutral or negative polarities, emotion detection techniques are used to classify the emotions expressed by a text into more specific categories like anger, fear, sadness and love.

Inferring emotions from textual data is a challenging task because efforts to achieve this have to depend on the information retrieved from text alone and no other ex-

pressive features, like facial affect or tone of voice. Textual data (as we mentioned in chapter 2 section 2.2) can be divided into two types: Formal text and Informal text. It is generally accepted that recognising emotions from the latter is more complicated than from the former [79].

In this chapter, we have chosen to focus on detecting emotions from, among the range of social media communications we could have chosen, tweets. Tweets are limited to 280 characters [32]. Twitter, as a social media platform, is globally available and freely accessible to people regardless of their culture, age or education. Furthermore, it is one of the two most popular social media platforms [58] in existence, with almost 500 million tweets being posted on a daily basis [110]. Users of Twitter (tweeters) may use either a mobile phone app or a web-based system to express their thoughts and emotions in real-time on a daily or shorter-duration basis.

In this chapter, we explore two different methods used for emotion detection; the first one, just assumes the system has a general background knowledge about emotive words (and is closely aligned to unsupervised learning), while the second one (which is the main contribution discussed in this chapter) involves using evolutionary strategies in order to detect multi-categorical emotions expressed in informal text. Although unsupervised models which can detect emotions from tweets are not abundant, there are examples of studies that use unsupervised methods for SA, e.g., [52] and [108].

The bulk of the work discussed in the following involves the extraction of emotions (*anger, disgust, fear, joy, sadness* and *surprise*) from tweets through the aid of word embeddings, specifically those implemented via Word2Vec [66]. Word embeddings transform words into vectorial representations, thus allowing distance calculations.

The new methodology proposed in this chapter is based on identifying "idealised" words that capture the essence of an emotion. We used the word embedding of the "idealised" emotion words and then employed the Word Mover's Distance (WMD)[57] function to calculate the distance between the tweets import and the "idealised" emotional words. Both WMD and Word2Vec were covered in more detail in Chapter 2 section 2.9.

Because of the dearth of appropriate unsupervised learning methods, supervised learning methods have been used most extensively for emotion detection relating to tweets [112, 9, 44]. And in this chapter, we propose a new method for employing supervised learning in this arena. We search a word vector space for an "idealised" emotion vector. In order to identify emotions, we employed a Euclidean Distance function to calculate the distance between the word embedding of tweets and the "idealised" emotion vector. To the best of our knowledge, this work is the first to explore detecting emotions from tweets by adopting an evolutionary strategy involving Word2Vec via distance functions.

The rest of this chapter is organised as follows. In the next section (4.2), we describe the dataset used. After that, in section (4.3) we present the benchmark approach. Section (4.4) will present the methodologies employed, including the main methods and models the work discussed in this chapter used. In Sections (4.5 and 4.6), we describe the experimental work. Section (4.7) presents the results of the experimental work. In Section (4.8) we describe different datasets experimental work and results. Section (4.9) will describe different embedding experimental work and results. Next section (4.10) presents the discussion. The chapter closes with the conclusions (4.11) drawn from this work.

4.2 Datasets

In this chapter, we chose to work on detecting emotions from tweets. Tweets were chosen because they are considered to be short text. Furthermore, emotion communicating hash-tags included in tweets have been used by some researchers to compile self-labeled datasets such as [69, 20, 112]. Wang et al. [112] discussed the idea that emotion-communicating hash-tag datasets are more accurate than manually labeled datasets because the former has, in effect, been labeled with the hash-tag emotion by the authors of the tweets rather than by somebody else. Moreover, we have chosen to work with the TEC dataset [69] for many reasons: it consisted of self labeled tweets, it represented the basic six emotions and it included about 21000 tweets which is a

good number of tweets to be used as the basis for experiments and evaluation.

The term TEC refers to the Twitter Emotion Corpus which was created by Mohammad [69]. It was constructed by searching for tweets which included one of the basic six emotions hash-tags, and this search was performed using the Twitter search Application Program Interface (API). Figure 4.1 shows the Word-cloud for the TEC corpus, including the hash-tags that have been used to label the data. Once the aforementioned search had taken place, tweets which contained fewer than three English words were eliminated. Also, the re-tweeted ones, which included the "RT" (retweet) prefix, were discarded in order to avoid redundancies. Badly spelled tweets and the tweets which did not have one of the basic six emotion hash-tags at the end of the tweet were also removed. In [69], it was argued that when the hash-tag label is in the middle of the tweet, they seem not to be so appropriate or accurate for use as a label. After the data cleaning process, the TEC dataset ended up with 21,051 emotionally self-labeled tweets from 19,059 different tweeters. We noted that the distribution of the emotions in the TEC dataset is imbalanced. For example, joy has the highest percentage, of 39.1% while the percentage of tweets mentioning disgust is less than 4%. Moreover, for the ground truth, the hash-tags that have been used as labels were removed from the tweets in the dataset, thus all the tweets within it may be used for both training and testing. It can be seen from the Word-cloud in Figure 4.2 that words like *joy, sadness, surprise* and *fear*, which were used for annotation, have disappeared.

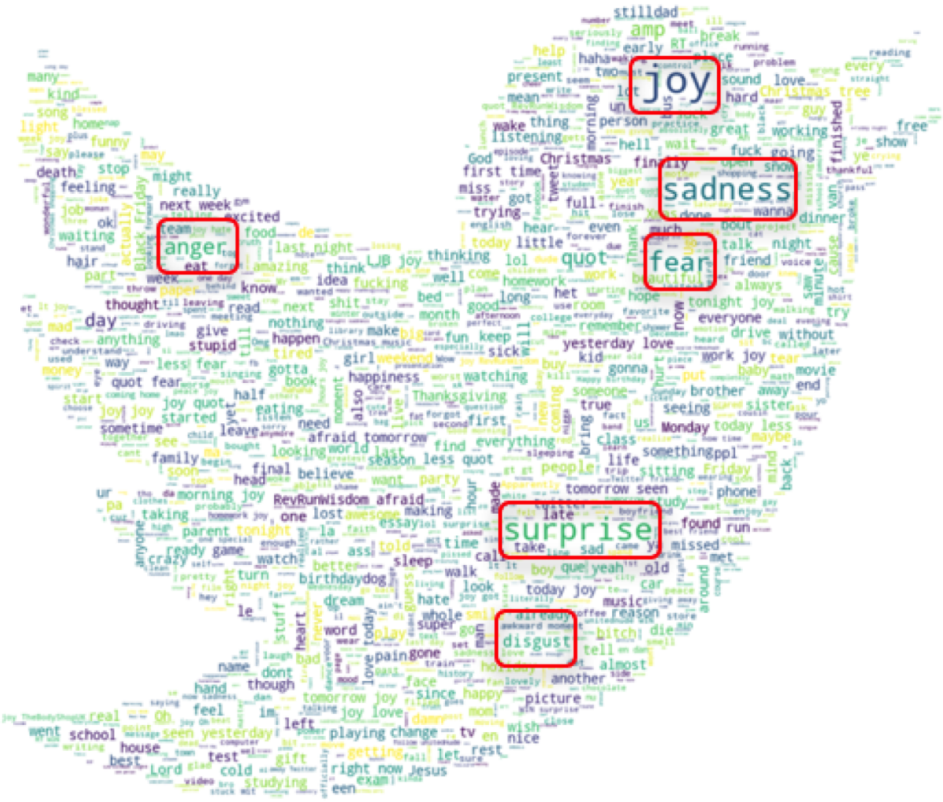

Figure 4.1: Word-cloud of TEC corpus including annotation (hash-tag) words like joy, sadness and surprise.

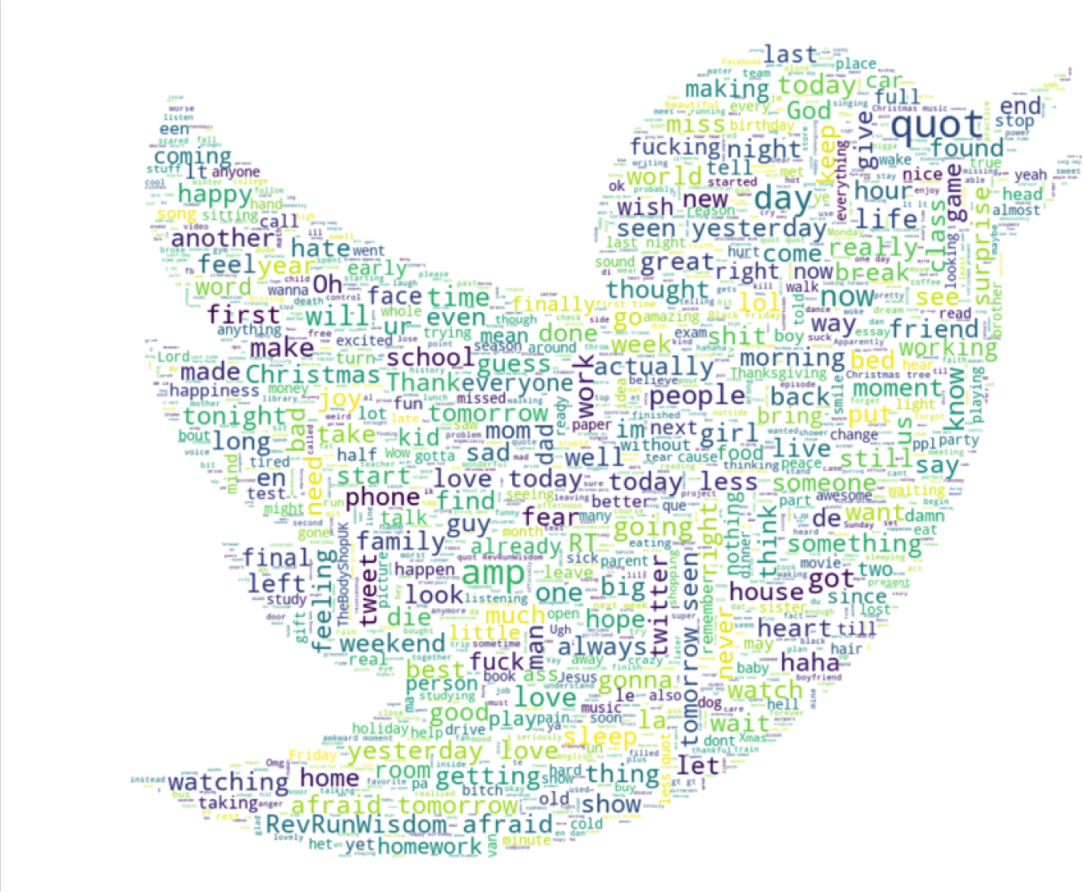

Figure 4.2: Word-cloud of TEC corpus after removing the annotation words, like joy and sadness (so that corpus can be used for training and testing).

4.3 Benchmark Approach

For each of the basic six emotions Mohammad [69] developed a binary classifier specifically to identify it, using WEKA [40]. For example, the classifier Fear-NotFear classifies whether a tweet expresses fear or not. Sequential Minimal Optimization (SMO) was the algorithm chosen in [69] to be used with SVM; binary features are taken account of in order to classify whether particular unigrams are present or absent. Unigrams that occur fewer than twice were discarded. The author reported the mean results of a 10-fold cross-validation, and we followed this approach.

4.4 Methodologies

4.4.1 WMD-ED

Word vectorization models, in general, map words into vectors using a dictionary. Word2Vec is a popular predictive word embedding model that was proposed by [66]. It was pre-trained by Google and made available as open source software. It contained 3 million embeddings for English words and phrases from Google News [57] and consisted of 300 dimensional corresponding word vectors. It has been described in more detail in chapter 2 section 2.8.3.

Word2Vec is capable of capturing the syntactic and semantic relationships between words without the supervision of humans. For instance, it will predict the strong country and capital city relationship between the words "Italy" and "Rome" [66]. Moreover, it is able to identify other semantic relationships such as the "Male" and "Female" relationships [65].

In a different domain, Godin et al. [34] trained a Twitter Word2Vec model on 400 million English language Twitter messages. The skip-gram process was performed in the course of this training, and a 400 dimensionality was obtained for the word embedding [34].

Our work relied on the approach called Word Mover's Distance (WMD) [57]. WMD utilizes word vectorization features such as Word2Vec embeddings. This has been described in more detail in chapter 2 section 2.9.

Our experiments, as described in chapter 3 were performed using WMD [57] with Word2Vec [66] to determine the emotions expressed in formal texts and it yielded good results. A similar method was adopted employing both Word2Vec [66], Twitter Word2Vec [34] and WMD [57] together to investigate the performance of unsupervised emotion detection in relation to short informal texts (Tweets), using the dataset, TEC, provided in [69]. By looking at the embedded words of each tweet message along with the Word Net-Affect emotional seed words [103], the dissimilarity between the tweets and the emotion seed words were derived. Stop words were removed from the tweets

first. Then, the WMD distance between the tweet sentences and the seed words representing each emotion category was calculated. The closest emotion to the tweet message was the emotion category that yielded the shortest distance in these terms. This distance was measured by determining the minimal cumulative distance of the remaining words of the tweet message that was needed in order to transpose them to correspond to the emotional words. At this stage, the results may be compared to the results from [69].

4.4.2 CMA-ES

The covariance matrix adaptation evolution strategy (CMA-ES) is a stochastic optimization algorithm which was proposed by Hansen et al, [42]. The CMA-ES is an iterative evolutionary algorithm that is based on the creation of a set of solutions for continuous optimization. It generates λ new vector solutions from a multivariate Gaussian distribution according to:

$$x_i \sim \mathcal{N}(m_k, \sigma_k^2 C_k) for\ i = 1, ..., \lambda \tag{4.1}$$

with mean m_k, covariance matrix C_k and step size σ_k.

In this work, we used CMA-ES as a black-box optimisation method in order to search through a high-dimensional search space without using gradient information.

4.5 CMA-ES Experiments

As discussed in the literature review (chapter 2), informal text has many challenges and it is more difficult than formal text in terms of retrieving information. The main purpose of our experiments here was to demonstrate that optimal results can be achieved by using "idealised" words and their WMD or Euclidean distances to identify emotions in tweets.

4.5.1 WMD-ED Experiment 1

In this experiment, a list of five "idealised" words for each emotion of Ekman's six [23] basic emotion categories were selected randomly from the Word Net Affect dictionary using a random selection function. For example, joy = [*joy content exhilarating cliff-hanging enthusiastically*]. As described in chapter 3 section (3.4.4) when the basic, definitive words (such as 'joy') were included in the "idealised" words, the WMD-ED experiments yielded better results. Consequently, the basic, definitive emotion words were always selected and then the randomly selected "idealised" words were added to them. As it can be seen from the example here, the main seed word (joy) was forced to be at the beginning of the joy category "idealised" words. Then in order to detect emotions in the tweets, the WMD distance is measured between the embedding of tweet and the "idealised" words, as described in WMD-ED methodology. The list of emotion "idealised" words that was the shortest distance from a tweet was considered to be the emotion category for that tweet. Following this, in order to measure the classification accuracy for each emotion category the F1 score was calculated. In this experiment the embedding model adapted here was Google's pre-trained model as described in the methodology section above.

Iteratively, the emotion "idealised" words for the category with the lowest F1 score category were randomly selected again while the other categories' "idealised" words remained the same. We follow the above process again and examine the new set of F1 scores for the emotion categories. This step was repeated so that it was undergone by all the categories - in order to achieve better results. The best discovered "idealised" words from which the best results were retained across all the emotion categories after 100 iterations of this process can be seen in Table 4.1.

4.5.2 WMD-ED Experiment 2

Since the writing style used in tweets is mostly very informal, in this experiment we tried to exploit word embeddings that have been trained on informal text. The embedding model adopted here for this experiment was the Twitter Word2Vec model

Emotion category	"idealised" words
anger	anger angry infuriation misanthropy anger
disgust	disgust nauseating disgusting hideous nauseated
fear	fear intimidate diffident afraid fear
joy	joy content exhilarating cliff-hanging enthusiastically
sadness	sadness sad demoralizing blue sorrowfully
surprise	surprise dumbfounded astonishing surprisedly stupefaction

Table 4.1: The emotion "idealised" words that achieved the best results for Experiment 1

as described in the methodology section.

The same steps as were carried out in WMD-ED experiment 1 were followed here, but using Twitter Word2Vec for the word embeddings. Also, the number of random "idealised" words for each category in this experiment was fixed at four. The best "idealised" words which this process discovered, that gained the best results across all emotion categories after 100 iterations, can be seen in Table 4.2.

Emotion category	"idealised"
anger	anger displease indignantly irascibility
disgust	disgust disgustedly abhorrent hideous
fear	fear cruelty presage hesitance
joy	joy anticipation titillating exultant
sadness	sadness desolate despondent plaintively
surprise	surprise wonder surprisedly surprise

Table 4.2: The emotion "idealised" words that achieved the best results for Experiment 2

4.6 Evolutionary Search Experiments

4.6.1 SNES

Using the same TEC dataset, in this experiment we were trying to find the "idealised" vectors that will perform optimally for emotion detection using SNES. SNES was

chosen as the evolutionary algorithm to be applied initially here because it uses only a diagonal of the covariance matrix for the search distribution and so will cost less in terms of computational time. In addition, to reduce the computational time further, Euclidean distance was employed as the measure of the distance between the tweets and the "idealised" vectors. Ten-fold cross validation was applied, as in the benchmark approach [69]. The general procedure was as follows: The vectors were retrieved for each of the words of a tweet from Google's pre-trained Word2Vec model [66]. Then the mean of the vectors was calculated - for each tweet. The resultant dataset (the mean vector for each tweet) was then split into ten groups. For each unique group:

1. Take one group as the test dataset.

2. Take the remaining groups as the training dataset.

3. Fit a SNES model to the training set and then evaluate it with the test set.

The number of SNES generations was set to 200, the population to 100 and the dimensionality to 1800; the latter because a vector of 300 dimensions was needed for each emotion category: *anger, disgust, fear, joy, sadness* and *surprise*. Initialize the σ step size to be 0.1. For each generation the algorithm is presented in Algorithm 2.

Algorithm 2: SNES Evolutionary Search

```
for each generation do
    Select mini batches (8000) randomly from training data.
    for each element in population do
        Calculate Euclidean distance between the "idealised" vector and the
        mean vector of each tweet of the mini batch.
        Set the emotion of the tweet based on the shortest distance.
        Calculate the F1 score.
    end
    Update the parameters for the next generation based on the results and
    the current "idealised" vector.
    if generation is even then
        Use the "idealised" vector to evaluate the model on the test data set.
        Retain evaluation scores.
    end
end
```

4.6.2 CMA-ES

Using the TEC dataset, in this experiment we attempted to find the "idealised" vectors that will perform optimally in emotion detection using the CMA-ES. The TEC dataset contained more than 21000 tweets and in order to work with word embeddings, it ends up with a very high dimensionality; therefore, CMA-ES was chosen as the evolutionary algorithm to be employed in this experiment. In addition, to reduce the computational time, Euclidean distance was applied in order to calculate the distance between the tweets and the "idealised" vectors. Ten-fold cross validation was adopted, as in the benchmark approach [69]. The general procedure was as follows: The vectors were retrieved from Google's pre-trained Word2Vec model [66] for each word of each tweet. Then the mean of the vectors was calculated for each tweet. The resulting dataset (the mean vectors for each tweet) was then split into ten groups. For each unique group:

1. Take one group as the test dataset.

2. Take the remaining groups as the training dataset.

3. Fit a CMA-ES model to the training set and evaluate the result on the test set

The number of CMA-ES generations was set to 200, the population to 100 and the dimensionality to 1800 - because a vector of 300 dimensions was needed, to corresponds with Word2Vec dimensions, for each emotion category: anger, disgust, fear, joy, sadness and surprise. The σ step size was initialised with 0.1. For each generation (the exact algorithm is presented in Algorithm 3):

1. Mini batches (8000 unique vectors) were selected randomly for training purposes from the training set for each generation of CMA-ES.

2. For each population, the Euclidean distance was calculated between the "idealised" vector and the mean vector for each tweet of the mini batch; thus, the emotion that was separated from the tweet by the shortest distance was se-

lected as the emotion category for the tweet. Following this, the F1 scores were calculated with respect to the results obtained.

3. Based on the evaluation scores yielded with respect to the selected emotion category and the "idealised" vectors of the current generation, the parameters were updated for the next generation.

4. For every even generation, the "idealised" vector from CMA-ES were used to evaluate the model on the test dataset.

5. The evaluation scores, the maximum score for each generation and the best fitness scores were stored.

The above procedure was carried out for the selected emotion category and was then repeated for all the others separately. The computational time used was 10 seconds per generation.

Algorithm 3: Evolutionary Search

```
for each generation do
    Select mini batches (8000) randomly from training data.
    for each element in population do
        Calculate Euclidean distance between the "idealised" vector and the
        mean vector of each tweet of the mini batch.
        Set the emotion of the tweet based on the shortest distance.
        calculate the F1 score.
    end
    Update the parameters for the next generation based on the results and
    the current "idealised" vector.
    if generation is even then
        Use the "idealised" vector to evaluate the model on the test data set.
        Retain evaluation scores.
    end
end
```

emotion\F1	benchmark	WMD-ED exp 1	WMD-ED exp 2	CMA-ES	SNES
anger	27.9	17.53	19.70	33.82 ± 1.87	14.16 ± 0.80
disgust	18.7	16.13	19.12	29.07 ± 3.16	8.41 ± 1.84
fear	50.6	40.52	37.41	52.30 ± 1.24	23.96 ± 0.62
joy	62.4	57.51	45.92	66.88 ± 0.56	57.81 ± 0.90
sadness	38.7	18.4	27.58	43.31 ± 0.79	31.64 ± 0.90
surprise	45	22.84	34.33	47.24 ± 0.87	31.88 ± 1.07

Table 4.3: Experiments results

4.7 Results

4.7.1 WMD-ED Experiment 1

Table 4.3 showed the best results yielded from WMD-ED Experiment 1 as well as the results reported in [69]. Table 4.1 contained the "idealised" words that have been used to obtain these results. It can be seen that for both categories joy and disgust our F1 scores were very close to those of the benchmark approach in [69]. However, the F1 scores for *anger, fear* and *sadness* were 10 scores below the benchmark approach, and the score for *sadness* and *surprise* were approximately less than half of the benchmark approach.

4.7.2 WMD-ED Experiment 2

Table 4.3 represented the F1 scores achieved using the emotional "idealised" words from Table 4.2 in our WMD-ED experiment 2. In terms of general comparison, the F1 scores of four emotion categories yielded by WMD-ED Experiment 2 are better than the corresponding F1 score yielded by WMD-ED experiment 1 Table 4.3. Furthermore, it can be seen that the F1 score for disgust (yielded by Experiment 2) exceeded the corresponding F1 score of the benchmark approach of [69] as can be seen in Table 4.3. In addition, the scores for *sadness* and *surprise* increased by about 10 marks as compared to our WMD-ED Experiment 1 scores.

4.7.3 Evolutionary Search Experiments

SNES

As can be seen from table 4.3, none of the results from this experiment has exceeded, in terms of F1 scores, the benchmark results. Noticeably, there was a big gap between the quality of the results yielded by this experiment and that of the results from the benchmark, especially in relation to three particular emotions: *anger, disgust* and *fear*. However, as mentioned earlier SNES was chosen initially for this task simply because it costs less computationally, but since its use did not achieve good results we decided to replace SNES with CMA-ES, as described next in 4.6.2, for further tests because CMA-ES method uses the full covariance matrix for the search distribution.

CMA-ES

Table 4.3 shows the average F1 score and the error bounds for the 95%th confidence interval derived from the ten-fold cross validation of the final generation test undertaken in the evolutionary search experiment. It can be seen that the evolutionary search results exceeded the results yielded in the other two WMD-ED experiments as well as the benchmark results [69]. The evolutionary search results achieved the best results over all the emotion categories; Table 4.3. Figures 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8 showed the line graphs of the average F1 results along with the confidence intervals (again, the 95%th interval) as regards using the CMA-ES mean vector for the ten-fold cross validation using the testing dataset in relation to the emotions: anger, disgust, fear, joy, sadness and surprise respectively. From these figures it can be seen that the F1 average of the scores yielded by CMA-ES run on the test dataset increases rapidly according to generation.

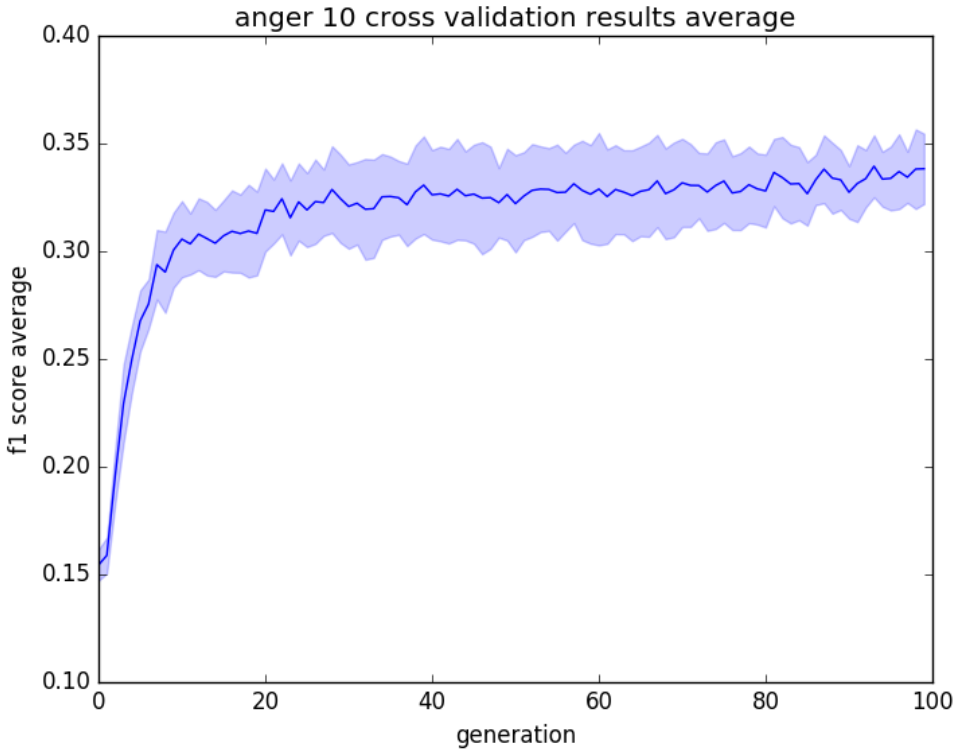

Figure 4.3: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for anger

Table 4.3 demonstrated that the results of the evolutionary search experiment have surpassed all previously published benchmarks. As mentioned in the datasets section, *disgust* had very low percentage of occurrence in the dataset and so, as could be expected, it has larger confidence intervals, see table 4.3. In contrast, *joy* had the largest percentage of occurrence in the dataset therefore all the experimental runs ended up yielding more or less similar results for this emotion - see Table 4.3.

Using the CMA-ES method to continuously update the "idealised" vector for each emotion in order for these to be employed for emotion detection using the Word2Vec embedding boosts the results as discussed above, and it was also found that the "idealised" vectors moved in the right direction – towards the exemplifying of the intended emotion category with relevant words. It can be seen that the "idealised" vector generally represented the selected emotion exemplified by the first most similar word in the Word2Vec distribution space. The *most_similar* method was used here to find the

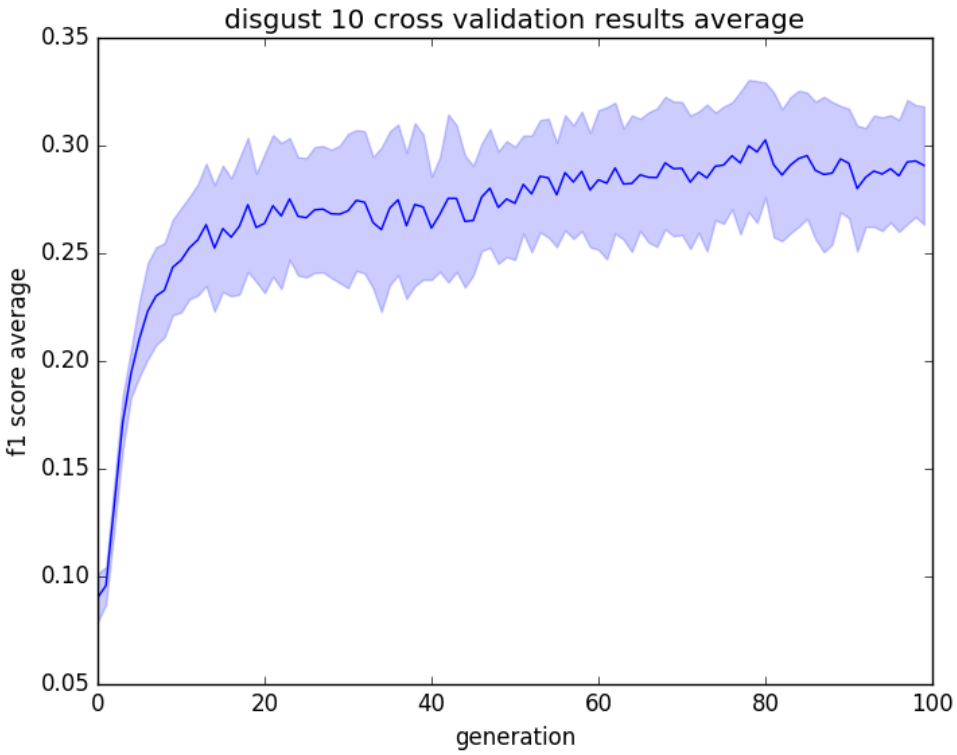

Figure 4.4: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for disgust

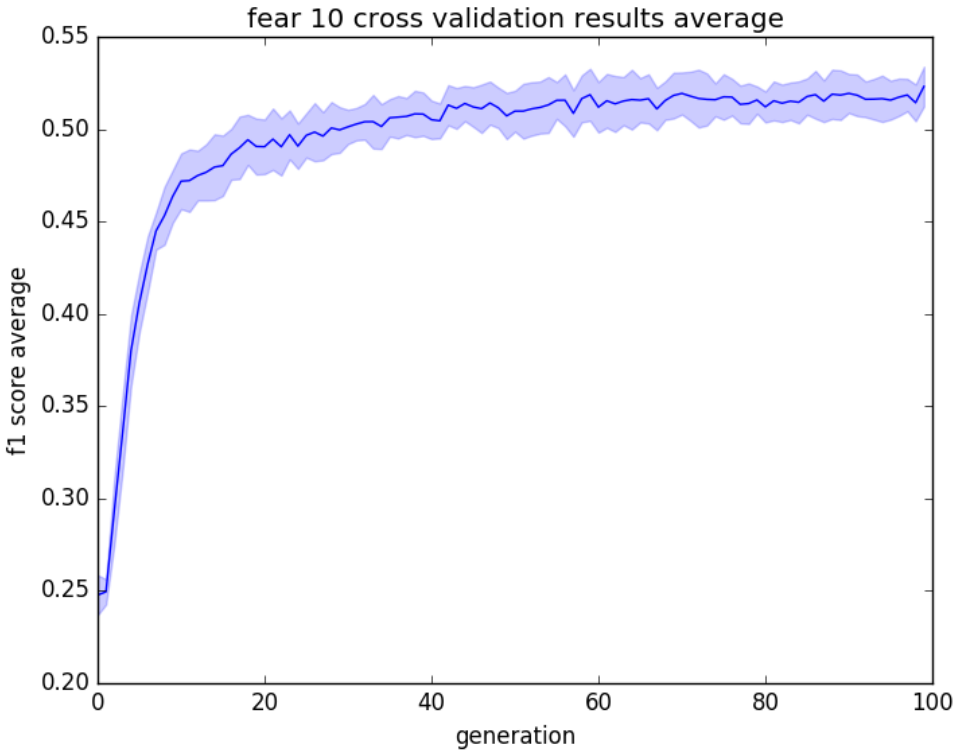

Figure 4.5: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for fear

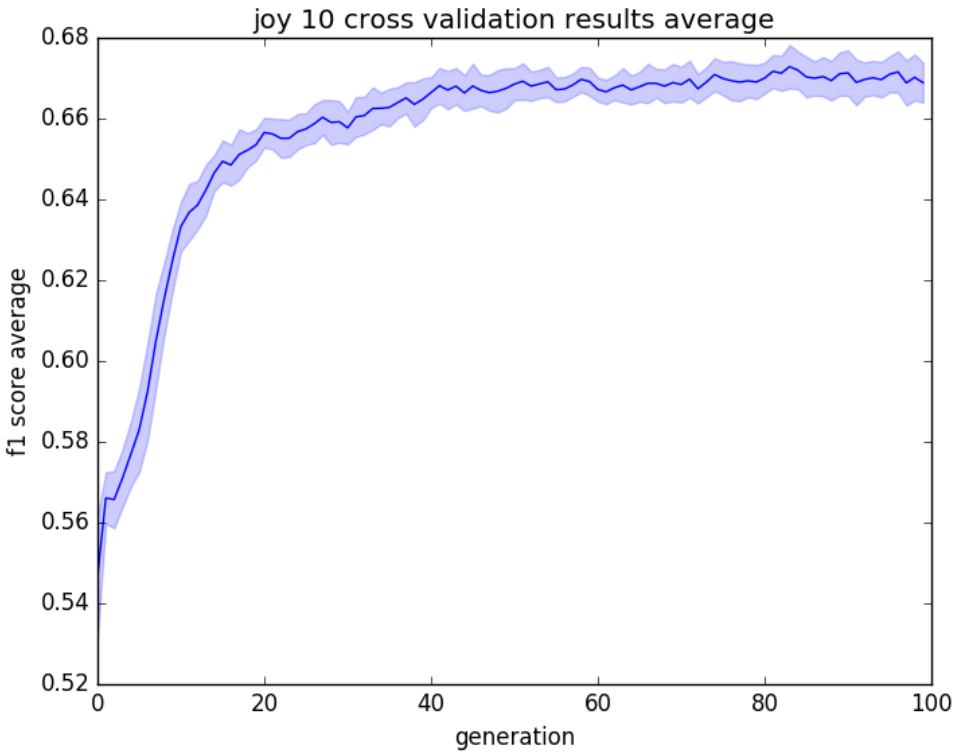

Figure 4.6: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for joy

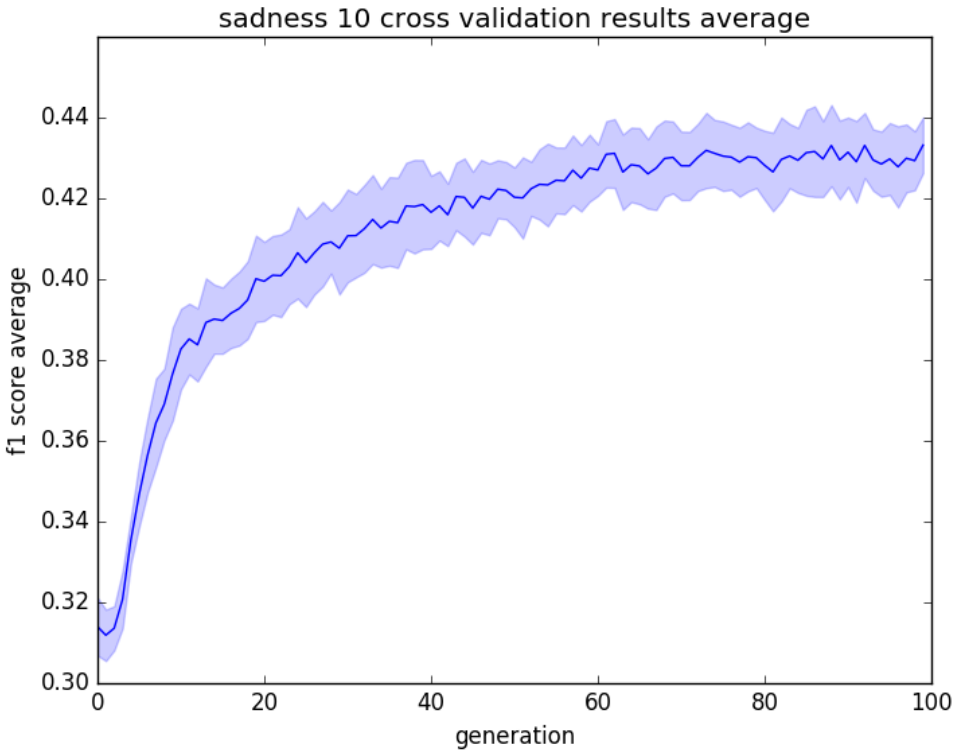

Figure 4.7: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for sadness

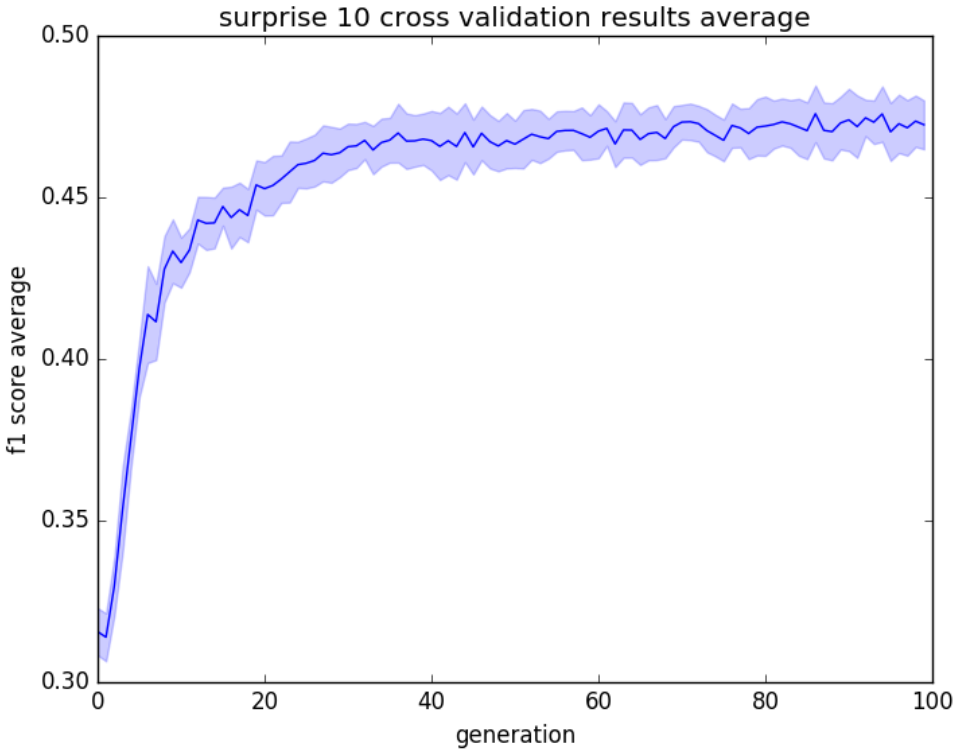

Figure 4.8: 10-fold cross validation plots of CMA-ES best of generation scores, for the 95th percentile for surprise

top most similar words which works by computing the cosine similarity between the "idealised" vector and vectors of the words in the model. For instance, the generated vectors for *anger* and *fear* absolutely represented their corresponding emotion category, see Table 4.4. Furthermore, some other words might be emotionally laden but not emotion key words. For instance, in relation to the emotion surprise, some of the most similar words such as `#/#-caratdiamondring` and `bag` might be surmised to be related to surprise because they are both associated with surprise gifts. Another example in terms of surprise category is the 'most similar' word, `JohnBelawsky`; In relation to this, after searching via Google we found many news articles (such as Fox-News[1]) presenting the information that he had won a big lottery jackpot ($330 Million) in 2007.

Furthermore, from the following tables (Table 4.5, Table 4.6, Table 4.7, Table 4.8, Table 4.9, Table 4.10) it can be seen that the vectors which were in the not-intended-

[1] `https://www.foxnews.com/story/retired-new-jersey-couple-claims-portion-of-330m-mega-millions-lottery-jackpot`

Emotion category	1st most similar words for each cross-validation	Most similar word to the mean
Anger	Angery, Zionist_regime_commits, anger, anger, anger, anger, ANGER, Seething, anger, sheer_stupidity	anger 0.48
Disgust	Racial_epithets, stinky, disgusting, sexual_harrassment, meatheads, crotches, disgusting, discarded_hypodermic_needles ,slurs, Ewww	Disgusting 0.44
Fear	fear, fearful, fear, fear, fear, fear, fear, fear, fear, fear	fear 0.56
Joy	work, MT_MLT, Alerion, Lectionary, quilting_sewing, Kopachuck_Middle, novena_prayers, novena_prayers, nutritious_breakfast, Century_Bike_Ride	T'ai_Chi_Chih 0.42
Sadness	actors_Mitchelson, reminders_bobbing, Lenny_Martelli, Bobbi_Boland, Razr, DVR'ed, Previous_ceasefires, Carol_Dycus, Jancy_Thompson, Turkoman_Shiites	reminders_bobbing 0.38
Surprise	hinted, #/#-carat_diamond_ring, overnighted, 4GB_iPod_Nano, Arvold, John_Belawsky, bag, overnighted, Keuylian,Traude_Daniel	Overnighted 0.36

Table 4.4: Most similar words to each vector from cross-validation for each emotion and most similar word to the mean

emotion categories mostly correctly included only not-intended-emotion words. For example, in Table 4.5 the intended emotion was *anger*; therefore, it can be noted that anger relevant words were not included in the other not-anger categories (e.g., *disgust, fear, joy, sadness* and *surprise*). Moreover, by using WMD distance, the minimal distance between the mean of the vectors of the top most similar words and the tweets in the dataset was calculated. However, the result that can be seen in Table 4.11 were unexpected because these were not good results in relation to the intended emotions in the top rows; also these were not good results in terms of using the most similar words of the mean of all of the emotion at the same time. Just to clarify that, here in Table 4.11, the words that were used for the distance measures were not the "idealized" vectors but only the top most similar word to the mean of them. It can be seen that, even though the top most similar words to the mean included words relevant to the emotion categories, these words were still not very close to the "idealised" vectors - see the distances in Table 4.4.

Emotion category	Mean words from						
	anger Table 4.5	disgust Table 4.6	Fear Table 4.7	Joy Table 4.8	Sadness Table 4.9	Surprise Table 4.10	Table 4.4
Anger	0	13.75	13.75	13.75	0	0	0
Disgust	6.97	0	0	0	0	6.97	0
Fear	0	0	0	0	23.59	0	0
Joy	0	0	0	0	0	0	56.2
Sadness	0	0	0	0	0	0	0
Surprise	0	0	0	0	0	0	0

Table 4.11: The F1 score of the results yielded by calculating the WMD distance between the mean vectors and the tweets

Emotion category	1st most similar words for each cross-validation	Most similar word to the mean
Anger	Angery, Zionist_regime_commits, anger, anger, anger, anger, ANGER, Seething, anger, sheer_stupidity	anger
Disgust	Nusa_Urbancic, Allard_J2X, Jean_Lesage_International, tech_savvy_counterfeiter, Professor_Pillinger, kU, WXOS_###.#, slotman, Shiwakoti, Adriano_Sofri	bustling_Sandaga_ Market
Fear	DYFS, Indelicato, spooky_Halloween, lingered, gradations, Pflipsen, ,grandest_stages, swimsuit_calendars, El_Cerro, Vander_Pluym	scouting
Joy	Approved_Auto_Repair, Chen_Jingni, Myanmar_Ibrahim_Gambari, Abbe_Raven, Azoreans, Hand_Arendall, Darlingtons, Shortcut, WILMINGTON_Del., Dolly_Lenz	Guard_adjutant_ge neral
Sadness	UPMC_Montefiore_Hospital, Tannia, Alfred_Binet, tidal_turbine, Murrell_Inlet, Investment_Insurance_NEXI, Mohammed_Nuru, Desert_Breeze, Reshef, McMenamins_Old	Exemplary_Teache r
Surprise	JG_Buzanowski, Soca_Warriors, Ivanovich, Bolero_jeeps, Victim_Assistance_NOVA, Toy_Collectors, Eric_Raimy, l_ocal, Andrea_Keilen, ##-#-#_##-#-#_ECACHL	BUNGAY_England

Table 4.5: The most similar words to the emotions anger and not-anger, the latter being all other emotions

Emotion category	1st most similar words for each cross-validation	Most similar word to the mean
Anger	AEIF, Clarian_Health_Partners, Inscriber_\xae, Ltd._TASE_MISH, Experimental_Prototype, AutoExpo, Flying_Objects, Picciolo, Enrico_Piperno, Mentioned	novonordisk.com
Disgust	Racial_epithets, stinky, disgusting, sexual_harrassment, meatheads, crotches, disgusting, discarded_hypodermic_needles, slurs, Ewww	disgusting
Fear	Pty_Ltd, MATI_Davao_Oriental, KITT, Ladie, Downtrend, Banked, Rueter_Hess, Intentions, YOU_WERE_READING, DATELINE_BANGLADESH	Machinery_Movers
Joy	By_BUTCH_HEMAN, divers_scoured, By_TIM_McDONOUGH, Alyeska_Ski_Club, Monastir, Aguas_Calientes, Moorhead_Minn._WDAY_TV, Ken_Dreifach, Dog_Whisperer, Dr._Toby_Litovitz	Ngari_Prefecture
Sadness	Tuitele, KUWAIT_CITY_Kuwait, Neil_McKissock, Knoah_Solutions, Great_Lakes_Lighthouse_Keepers, www.fbo.gov, ang_aming, Barretts_Juvenile, Rescuer, mullah_Ali_Khamenei	STAR_OCEAN
Surprise	Maghreb_Minerals, Ruger_M##, USEC, Grenadian, Massage_Therapy_Clinic, coach_Phil_McNichol, Guillermo_Cock, drab_browns, Embarkation, baking	Blazes

Table 4.6: The most similar words to the emotions Disgust and not-Disgust, the latter being all other emotions

Emotion category	1st most similar words for each cross- validation	Most similar word to the mean
Anger	CPI_Aero_SEC, Crofut, ice_cream_scoopers, Higher_Education_Supplement_THES, logical_progression, Excellent, flamin, Onnerud, T'nalak_Festival, President_Shintaro_Tsuji	Cake_Auction
Disgust	Anstaett, Calfrac_Well_Services , Mass_Appraisal, Paratek_Pharmaceuticals, bondage_outfits, Magic_Hat, Lina_Tipnis, Matchmaker, PolyMedix, polyadenylation	JHT_Holdings
Fear	fear, fearful, fear, fear, fear, fear, fear, fear, fear, fear	fear
Joy	Farmer_Lender_Mediation, Chenoa_Maxwell, Tricentennial, alley_oop_lobs, Arasaratnam , VIAGRA_SIDE_AFFECTS, Shelley_Duvall_Faerie_Tale, Eichenberger, hysteroscopic, ComiCon	mouth_agape
Sadness	auxiliary_input, Gromov_Flight, Shotaro_Yachi, chairback ,RM###.#m, Elmerton_Ave, Tarian, avalable, sipped_vodka, Schwinn_Varsity	#/#-season
Surprise	freecreditscore.com, Chaigneau, ##,###,###-##,###,###_Prepayments, shea_butter_soap, Inti, Helio_Ocean, Kasha_Chamblin, Plains_Midstream, tithing, transport_Dominique_Bussereau	By_MATT_STRAYE R

Table 4.7: The most similar words to the emotions Fear and not-Fear, the latter being all other emotions

Emotion category	1st most similar words for each cross- validation	Most similar word to the mean
Anger	Lennox_Josh_Duhamel, Eric_Niessen, OLYMPIC_VALLEY, Samak_Sundaravej, Yum_Brands, retaliation, Rear_Admiral_Harry_Arogundade, Munich_Ludwig_Maximilians, Kournikova, Byfulgien	Alferhan_faith
Disgust	Yu_Hai, Infosys_Technologies_Satyam_Computer, economist_Philip_Verleger, self_destruct, NORTH_ADAMS_Mass., Mutts, i_plas, Fighting_Gobblers, Ecumenical_Theological_Seminary, Rawkus	Petroskey
Fear	Amish_Tripathi, systematically_persecuted, traveling_northbound, By_VARIndia_Correspondent, mother_Nadine_Trintignant, Varoga, warthog, Deepwater_Horizons, Karpinski, Council_MPSP	HBGary_emails
Joy	work, MT_MLT, Alerion, Lectionary, quilting_sewing, Kopachuck_Middle, novena_prayers, novena_prayers, nutritious_breakfast, Century_Bike_Ride	T'ai_Chi_Chih
Sadness	Radhika_Coomaraswamy, chairman_Bukar_Shekau, UNWTO, DOWNE_TWP, lodges, \xc5str\xf6m, Micromanagement, Ape, Terayon_Comm, silver_medalist_Chiharu_Icho	Ewa_Baradziej_Krz yzankowska
Surprise	LHDs, Hypes, fearing_backlash, Sarpanchs, THE_PROBLEM, patent_troll, Allegedy, uninvolved_bystander, Outrageous, Southport_Conn.	Zhang_Wenping

Table 4.8: The most similar words to the emotion Joy and not-Joy, the latter being all other emotions

Emotion category	1st most similar words for each cross- validation	Most similar word to the mean
Anger	YEI, Color_Guard, analyst_Akihiro_Shiroeda, alla_prima, L'_Autre, cussing, Hug_Porcupine, photo_voltaic_cells, pantaloons, Beckhams	salespersons
Disgust	Stefan_Alois, \xfdPage_=, gyan, UGS_NX_TM, Ankit_Jain, UNWRITTEN_LAW, Heavy_Duty_Engine, gnash, Ralph_Yirikian, Pritt_Stick	shway
Fear	Incentivizing, factor_beta_TGF, Abu_Talha_al_Sudani, qPCR_assay, Bruce_Yarwood_president, By_Nick_Mokey, CLIO_Mich., Super_Ultra_Low, Gueye, PARIS_AFX_Societe_Generale	Atlas_Copco_AB_ST O
Joy	Kilo_Watt, rts, Intentional_fouls, wobblers, N_Srinivasan, PERKASIE, M\xf6tley_Cr\xfce_frontman, enraging_followers, Ted_Oberg, expositor	D.Earnhardt_Jr._## #-###
Sadness	actors_Mitchelson, reminders_bobbing, Lenny_Martelli, Bobbi_Boland, Razr, DVR'ed, Previous_ceasefires, Carol_Dycus, Jancy_Thompson, Turkoman_Shiites	reminders_bobbing
Surprise	BEA_Business_Objects, Tutong_Dakwah_Department, respite, Gloria_Kovach, HIW, CORPORATE_STOCK, Maingot, T-##/##, Freight_Line, BY_ALLISON_REDMAN	ector

Table 4.9: The most similar words to the emotion Sadness and not-Sadness which is the position of other emotions

Emotion category	1st most similar words for each cross- validation	Most similar word to the mean
Anger	disinformation_misinformation, moms_dads_grandmas, Lose_Guy, Randstad_Holding_EURONEXT_RAND, nostalgic_yearning, deeping, Leonard_L._Cavise, blights, Unclassified_SBU, meditations	purposelessness
Disgust	Inc._Nasdaq_ZEUS, concentric_spheres, noncompete_agreement, Morganza_spillway, op_ed_columns, Superintendent_Elizabeth_Yankay, pitcher_Johan_Santana, Lancer_Evolution, plagues, Movement_REM_sleep	drifting_downwards
Fear	ended_Monsignor_Weakland, strife_ridden, Screamin_roller_coaster, System_DACS, expectations, Page_B5, Cancel_crossed_paths, Agfa_Gevaert, ##Gbps_wavelength, Ladd_Everitt	palpably
Joy	Tuomas_Sandholm, Dodge_Magnum_sedans, untold_miseries, FLOSS, RealTime_Acura, orange_fireballs, Guerrieri, Wheat_physiologist, hedonistic_pursuits, Bank_Pomina_Steel	Welch_Allyn_specializes
Sadness	poisoned_Kool_Aid, blanket_prohibition, adjuncts, Jorge_Villasante, Indonesian_maids, Ron_Waksman, Oakfield_Primary, Finmark_Trust, Patriarco, Woody_Allen_Stardust_Memories	alleviations
Surprise	hinted, #/#-carat_diamond_ring, overnighted, 4GB_iPod_Nano, Arvold, John_Belawsky, bag, overnighted, Keuylian, Traude_Daniel	overnighted

Table 4.10: The most similar words to the emotions Surprise and not-Surprise, the latter being all other emotions

4.8 Different Datasets

The proposed method introduced in section 4.6.2 achieved good results over the Twitter dataset. As different types of datasets such as new headlines and tweets are conceptually different and they may convey or contain emotions in different ways. Therefore, it is useful to assess the method's effectiveness on other datasets and compare the results in order to demonstrate the generality of this approach. For this purpose, in what follows we tested our method on two other datasets from different domains. However, news headlines dataset has been used in WMD-ED experiments in chapter 3 but it was not possible to run evolutionary strategies such as CMA-ES on the same dataset because it only includes 1000 annotated headlines while for the optimisation purpose a larger amount of data is needed to attain and optimal vectors. Furthermore, for future work we intend to use the "idealised" vectors we have optimised in models trained on Twitter data to represent the same emotion category in a smaller datasets such as news headlines.

4.8.1 ISEAR Dataset

The second dataset we have used is the ISEAR (International Survey on Emotion Antecedents and Reactions) dataset [92]. ISEAR consisted of 7666 emotional sentences. 1096 students with different cultural background participated in answering the questionnaire regarding their experience in seven different emotion categories. As shown in table 4.12 the sentences were annotated with seven emotion categories: anger, disgust, fear, guilt, joy, sadness and shame.

Benchmark

Razek and Frasson [82] employed the ISEAR dataset. Exploiting the ISEAR dataset described earlier, [82] employed a text-based emotion detection approach. This approach was based on the Dominant Meaning Technique which is using the meaning of the word and allows to form the dominant meaning tree for emotion detection in text. The authors implemented the emotion detection algorithm to determine the

emotion \F1	benchmark	CMA_ES	Cohen_kappa_F1
anger	0.66	0.399	0.428
disgust	0.47	0.51	0.537
fear	0.56	0.62	0.636
guilt	0.50	0.38	0.407
joy	0.58	0.61	0.644
sadness	0.67	0.54	0.585
shame	0.55	0.39	0.418

Table 4.12: ISEAR experiments results

seven emotion categories (anger, disgust, fear, guilt, joy, sadness and shame) which were annotated in the dataset.

Experiments and results

Ten-fold cross validation was adopted, as in the benchmark approach[82]. The general procedure is that we attempted to find the "idealised" vectors using the same CMA-ES approach as in 4.6.2 in order to detect the emotion category. In this experiment, the number of CMA-ES generations was set to 400 and the population to 200. For each generation, the Mini batches here was set to use the whole training data for the training purposes because of the small size of the dataset. Furthermore, in order to gain better results we have optimised the cohen_kappa_score, which is a statistic that frequently used to test inter-rater reliability, instead of the F1 score and on the test we calculated the F1 scores. As shown in table 4.12 our approach has surpassed the benchmark in three categories. Moreover, from table 4.12 we can observe that there was an increase in our results across all categories when optimising cohen_kappa_score instead of optimising using F1. Although we have achieved better results in three of the categories, this was not across the board. This may be due to the small size of the training dataset.

4.8.2 EmoContext Dataset

The third dataset we have used is the EmoContext (Contextual Emotion Detection in Text) dataset [15]. EmoContext consisted of 30160 emotional dialogues (three-turn conversation per dialogue) as training data. Half of the training data belongs to three emotion classes: *happy, sad* and *angry*. While the other half of the training data belongs to the *others* class. The test data contained 5509 dialogues while approximately 15% belonged to the three emotion classes *happy, sad* and *angry*; the remaining of the test data belonged to *others* category.

Benchmark

Exploiting the EmoContext dataset described earlier, [106] employed the Gated Recurrent Neural Network (GRU) model. This approach used Textual information layers to concatenate with GRU layer as auxiliary layer. The model also used fasttext, which is open-source library that allows users to learn word representations and text classification, as embedding layer. For emotion detection, in each sentence they tagged every emotional word using the NRC emotion lexicon [70]. The majority occurrence emotion was picked as the sentence emotion. When all emotion tags were equal, then a random emotion was picked as the sentence emotion. While, in the absence of emotional words then the category others was picked as the sentence emotion.

Experiments and results

EmoContext training data was adopted for training as in the benchmark approach [106]. The general procedure was that we attempted to find the "idealised" vectors using the same CMA-ES approach as in 4.6.2 in order to detect the emotion category. In this experiment, the number of CMA-ES generations was set to 400 and the population to 200. For each generation, the Mini batches was set to 25000 unique vectors which was selected randomly for training purposes from the training set. For the final results we used the "idealised" vector to evaluate the model on the test data set. Furthermore, in order to gain better results we have optimised the cohen_kappa_score

emotion \F1	benchmark	CMA_ES	Cohen_kappa_F1
happy	0.491	0.337	0.349
sad	0.567	0.344	0.380
angry	0.625	0.475	0.516
others	-	0.880	0.849

Table 4.13: EmoContext experiments results

instead of the F1 score and for the final results the "idealised" vector was used to evaluate the model on the test data by calculating the F1 score. From table 4.13 we can observe that we had obtained good results, but we had not surpassed the benchmark in the three categories. Moreover, there was an increase in the results while using the cohen_kappa_score, instead of optimising using F1, across all categories in comparison to the results when optimising using F1 scores. However, two possible reasons might had an impact on our results. First, the fact that this dataset was an imbalanced dataset. Second, the sentences were a merged 3-turn conversation which means these three sentences might differ in their laden emotion. Be that as it may, more measurements for the training set need to be done and more experiments to be done to find out whether overfitting, underfitting and the imbalanced have an affect on the results.

4.9 Different embedding

4.9.1 Exploring domain specificity using GloVe

As regards domain specificity, we thought we should try to run the WMD-ED experiment exploiting a different word embedding model that was pre-trained using a tweets dataset. As mentioned in (section 2.8.3) Google's Word2Vec is a predictive model trained on Google News data [57]. Thus, we selected GloVe (Twitter based – 2 billion tweets, 27 billion tokens, 1.2 million words vocabulary, uncased, 200 dimension vectors)[2] ; this is a count-based word embedding model [77].

[2] `https://nlp.stanford.edu/projects/glove/`

Emotion	Seed words			
Anger	anger	aggravation	baffled	fury
Disgust	disgust	disgust	disgustingly	disgust
Fear	fear	suspense	hideous	timid
Joy	joy	comforting	softhearted	satisfied
Sadness	sadness	laden	contrite	sadden
Surprise	surprise	wonderfully	perplex	dazed

Table 4.14: The emotion seed words that achieved the best results in the course of the GloVe experiment

The same procedures were followed as in WMD-ED experiments 1 and 2 (sections 4.5.1 and 4.5.2), but using GloVe for the word embeddings. Also, the number of random "idealised" words for each category used in this experiment was 4 words. The best "idealised" words, those which gained the best results across all the emotion categories after 100 iterations, produced by this experiment can be seen in Table 4.14. However, Table 4.15 showed the results and it can be seen that, in general, this approach did not achieve any better results than those obtained from the WMD-ED experiments.

emotion \F1	benchmark	GloVe
anger	27.9	14.71
disgust	18.7	18.33
fear	50.6	20.57
joy	62.4	54.53
sadness	38.7	00.15
surprise	45	33.42

Table 4.15: GloVe experiment's results

4.9.2 Emotion Embedding

Again, in order to take account of domain specificity, we trained our own word embedding model on data that was related to emotions as well as tweets. In order to yield good embeddings, Word2Vec requires large quantities of text. However, although such large quantities of input text are important, we needed, also, to find text that was emotionally charged, as much as possible. For example, Tixier, Vazirgiannis, and Hallowell [109] stated that they trained an embedding focused on construction, using large quantities of text which were related to construction, and consequently they obtained competitive results and indeed outperformed Google Word2Vec in many cases which were related to construction. Therefore, we tried to find a large emotion-related text corpus. Consequently, we trained emotion-Word2Vec (using Google's Word2Vec C Source File [3]) on a large dataset (the sentiment140 dataset [33]) - which contained 1,600,000 tweets.

By performing the task of finding the most similar word (see Table 4.17) it was discovered that, in general, mostly the resultant words were related to the corresponding categories. So the majority of the words under the column entitled surprise could be considered as relating to *surprise*: such as *present, gift, gifts, celebration* and *invitation*. On the other hand, we still see a few cases of overlap – for instance, *sorrow* under the *joy* category and *anger* under the *fear* category. Even though the most similar word task yielded reasonably good results, when the same procedure as was used in WMD-ED experiments 1 and 2 (sections 4.5.1 and 4.5.2) was followed, but using Emotion Word2Vec for the word embeddings, the results were not as good as expected. However, the number of the random "idealised" words representing each category in this experiment was set to just four. The best "idealised" words discovered in this experiment, that provided the best results across all the emotion categories after 100 iterations can be seen in Table 4.16. Table 4.15 presented the results and it can be seen that, in general, for this experiment, the quality of the results in terms of the F1 score were far lower than that of the benchmark's. It can be argued that the

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

Emotion	Seed words			
Anger	anger	persecute	exasperating	vindictiveness
Disgust	disgust	disgust	disgustingly	disgusting
Fear	fear	frightened	trepidation	chill
Joy	joy	softhearted	happily	caring
Sadness	sadness	sorrow	dolefully	despondent
Surprise	surprise	sorrow	dolefully	despondent

Table 4.16: The emotion seed words that provided the best results in the Emotion-Word2Vec experiment.

reason for this relatively poor result was that Word2Vec requires large quantities of text to train with, and we only trained it with 1.6 M tweets.

emotion \F1	benchmark	GloVe
anger	27.9	11.96
disgust	18.7	12.43
fear	50.6	06.48
joy	62.4	02.91
sadness	38.7	28.34
surprise	45	04.75

Table 4.18: Emotion Wor2Vec Experiment's results

4.10 Discussion

Our findings demonstrated that, optimising an "idealised" emotional vector using evolutionary strategies (CMA-ES) for emotion detection demonstrated and optimal results. Our results in CMA-ES evolutionary search experiment surpassed the benchmark approaches' results and produced state of the art results. For the generality

10 top most similar words for					
anger	**disgust**	**fear**	**joy**	**sadness**	**surprise**
causes	spies	fears	joys	heartache	suprise
anxiety	irritate	overcome	happiness	tragedy	present
existence	whoda	disease	bliss	electricity	gift
behavior	fallow	anger	deepest	fears	bday
disease	singular	affected	lord	tragic	presents
excess	veo	constant	contribution	mankind	pressie
emotions	richardson	caused	sorrow	failure	gifts
actions	parabxc3xa9ns	anxiety	laughter	loss	celebration
curse	shaa	victim	greatness	leroi	housewarming
eeriencing	quotwtfquot	causes	blessings	verge	invitation

Table 4.17: Emotion embedding - most similar words for each category

of this approach, it can be used on a different dataset as they may contain emotion in a different ways. Although ISEAR dataset maybe small, we have achieved better results in three categories when the CMA-ES approach was used. This model can be used as classification model for other domains by optimising an "idealised" vector in embeddings space. A limitation of this approach is that, the self-annotation used in the TEC has to be removed because it was used for labelling and which means the sentence will be without the main emotional word during training and testing. For future study, we will explore the effectiveness of using a different evolution strategies techniques, such as stochastic gradient descent and newton's method, on a similar manner. Furthermore, we would evaluate the effectiveness of detecting emotion categories by using the achieved "idealised" vector in this study to represent the same emotion category on a different dataset.

4.11 Summary

In this chapter, we followed the novel approach from the previous chapter (3) but on a different dataset (tweets). Two WMD-ED Experiments was performed using the

WMD-ED approach to calculate the distance between the embeddings of the tweets and the emotions which represented by several seed words retrieved from WordNet Affect randomly. These two experiments (WMD-ED Experiment 1 and 2) employed both Word2Vec [66] and Twitter Word2Vec [34] respectively. These experiments yielded reasonably good results compared to the benchmark approach [69]. This indicates that the unsupervised methods used in these experiments were able to detect emotions from tweets without the aid of annotations, ontologies or term extraction, and still attaining results close to those of [69]. WMD-ED Experiment 2 yielded F1 scores which were better than those obtained from WMD-ED Experiment 1 and we surmise that the use of the domain specific model (Twitter Word2Vec) was the reason behind these better results – which pertained across most emotion categories. However, these experiments did not exceed the benchmark results therefore the evolutionary search experiments was adopted.

On the other hand, as different types of datasets may convey or contain emotions in different ways; therefore, we tested our novel approach that introduced in section 4.6.2 on two other datasets from different domains section(4.8). Moreover, in order to take account of domain specificity, the WMD-ED experiment was performed by exploiting two different word embedding models that were pre-trained using a tweets dataset. These word embedding models were, GloVe embedding model [77] and we trained our own word embedding model on data that was related to emotions as well as tweets.

We presented a new method for performing emotion detection in informal text (tweets). We search the word vector space for an "idealised" emotional vector. In order to identify emotion, we adopted the Euclidean Distance function to calculate the distance between the word embedding of tweets and the "idealised" emotional vector. Our approach used evolution strategies (CMA-ES and SNES) to optimise "idealised" vector in embeddings space which can represent an emotion category during the distance calculation process of emotion detection. Using CMA-ES to continuously update the "idealised" vector for each emotion in order for this to be used for emotion

detection (using the Word2Vec embedding) boosted the results across all fronts. To the best of our knowledge, this work is the first to detect emotions from tweets by adopting an evolutionary strategy along with word embeddings. Most importantly, the "idealised" vector, to a large extent, does represent the intended emotion correctly and we verified that by going through the process of finding the most similar word to the generated vector. The representation of the intended emotion was always achieved either by some of the basic emotion words like fear or anger or by words which were emotionally laden and closely associated with the corresponding category.

Chapter 5

Conclusions

Understanding emotions in short text especially informal text is a topic of interest in NLP. In this thesis, we have investigated the problem of emotion detection in text, focusing primarily on emotion detection in short text. This research had mainly focused on developing emotion detection methods so that they can effectively operate on short text. The principal goal was to develop and extend methods for detecting emotions by exploiting the word embeddings because it is a promising technique which is able to capture meaning, semantic and syntactic similarity and different relationships between words. The novelty of the work presented lies in the way that word embeddings associated with the distance functions and evolution strategies were adopted in order to develop emotion identification methods. In the first section of this chapter, we present the summary of the work that we have undertaken and the contributions that we believe we have made. Then, the second section will mention some potential future research which can be pursued of this work.

5.1 Summary and Contributions

Most of the prior work on emotion detection in text were relying on emotional keywords or word lexicons such as [56], [98], [10], [25], [13], [115], [95] and [6]. In this thesis, we went beyond that by getting rid of the reliance on emotional keywords or word lexicons. We adopted word embeddings with some metric functions for emotion

detection in short text. In this thesis, we proposed and analysed several methods for improving emotion detection from short text. In conclusion, the main contributions of our work can be summarised as follows:

We presented a novel unsupervised approach for emotion detection from short text (chapter 4), which is using word embeddings and WMD to identify emotion in news headlines. We demonstrated that our WMD-ED Experiment 2 results exceeded the benchmarks results in general. Our approach in WMD-ED Experiment 2 gained the best F1 scores for all the emotion lists while the benchmark systems, in contrast, did not achieve the best F1 scores in the majority of the emotion categories. It is worth noting that our results is the best over all as compared to the other eight systems mentioned in the benchmark which were using different approaches such as simple heuristics, latent semantic analysis and naïve Bayes classifiers. In conclusion, these results confirmed the novelty of using word vectorization (Word2Vec) and WMD in predicting emotions, as it provided optimal results across all emotion categories. We believe that this is the first investigation into using Word Mover's Distance to detect emotions within text [5]. Not to mention that, a relevant work was published after our publication [85].

We described a novel method for detecting emotion in informal text (tweets) (chapter 5), where we used evolution strategies to optimise "idealised" vector in embedding space which can be beneficial in representing emotion category during the distance calculation process of emotion detection. The "idealised" vectors were meant to capture the essence of an emotion. We define these vectors as having the minimal distance (using Euclidean distance) between a vector and the embeddings of the sentence that contains the relevant emotion. The rational behind identifying an "idealised" vector for an emotion was that we need to identify one vector or word which can represent the emotion category and would be beneficial in the process of emotion detection. Our approach using the CMA-ES to continuously update the "idealised" vector for each emotion in order to be used for emotion detection using the Word2Vec embedding boosted the results in all fronts. To the best of our knowledge, this work is the

first to detect emotions from tweets by adopting an evolutionary strategy with word embeddings.

Lastly, we demonstrated the usefulness of our approach in (Chapter 5) by undertaking further analysis and trying to identify which words correspond to or are closer to what the evolutionary algorithm discovered in the embeddings search space. The continuously updated "idealised" vector generally represents the selected emotion exemplified by the first most similar word in the Word2Vec distribution space.

5.2 Future Work

The work presented in this thesis describes methods in emotion detection from short text such as news headlines and tweets. While the presented methods in this thesis represent significant advances in identifying emotion from short text, there are still many interesting opportunities for future research that deserve further pursuit.

For future work, we are aiming to improve the robustness of the learned representation by training a new Word2Vec model for the domain of emotions especially in informal text. Very large quantities of input text which is emotionally charged is required, for training purposes, to yield good word embeddings. Furthermore, we employed the CMA-ES evolution strategy in order to optimise the "idealised" vector which represents the selected emotion category. Thus, in the future we will explore further the use of different evolution strategies techniques, such as stochastic gradient descent and Newton's method, on a similar manner. In addition, as we have achieved optimal results by optimising the idealised vector, we could extend our work by evaluating the effectiveness of using the idealised vector we have optimised in the current dataset to represent the same emotion category in the process of emotion detection on a different dataset. Moreover, the same evolution search method we have employed could be used to learn transformations in other data beyond tweets e.g. images in order to detect emotions for instance.

Bibliography

[1] Ameeta Agrawal and Aijun An. "Unsupervised emotion detection from text using semantic and syntactic relations". In: *Proceedings of the The 2012 IEEE /WIC /ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society. 2012, pp. 346–353.

[2] Ameeta Agrawal, Aijun An, and Manos Papagelis. "Learning emotion-enriched word representations". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 950–961.

[3] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. "Emotions from text: machine learning for text-based emotion prediction". In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics. 2005, pp. 579–586.

[4] Mohammed Alshahrani, Spyridon Samothrakis, and Maria Fasli. "Identifying idealised vectors for emotion detection using CMA-ES". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2019, pp. 157–158.

[5] Mohammed Alshahrani, Spyridon Samothrakis, and Maria Fasli. "Word mover's distance for affect detection". In: *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*. IEEE. 2017, pp. 18–23.

[6] Saima Aman and Stan Szpakowicz. "Identifying expressions of emotion in text". In: *Text, speech and dialogue*. Springer. 2007, pp. 196–205.

[7] Saima Aman and Stan Szpakowicz. "Using Roget's Thesaurus for Fine-grained Emotion Recognition." In: *IJCNLp*. 2008, pp. 312–318.

[8] Massa Baali and Nada Ghneim. "Emotion analysis of Arabic tweets using deep learning approach". In: *Journal of Big Data* 6.1 (2019), p. 89.

[9] Rakesh C Balabantaray, Mudasir Mohammad, and Nibha Sharma. "Multi-class twitter emotion classification: A new approach". In: *International Journal of Applied Information Systems* 4.1 (2012), pp. 48–53.

[10] Susana Bautista, Pablo Gervás, and Alberto Dıaz. "Adaptive text generation based on emotional lexical choice". In: *Proceedings of the XV International Conference on Human Computer Interaction*. ACM. 2014, p. 20.

[11] Hans-Georg Beyer and Hans-Paul Schwefel. "Evolution strategies–A comprehensive introduction". In: *Natural computing* 1.1 (2002), pp. 3–52.

[12] Chetashri Bhadane, Hardi Dalal, and Heenal Doshi. "Sentiment analysis: Measuring opinions". In: *Procedia Computer Science* 45 (2015), pp. 808–814.

[13] Haji Binali, Chen Wu, and Vidyasagar Potdar. "Computational approaches for emotion detection in text". In: *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*. IEEE. 2010, pp. 172–177.

[14] Margaret M Bradley and Peter J Lang. *Affective norms for English words (ANEW): Instruction manual and affective ratings*. Tech. rep. Citeseer, 1999.

[15] Ankush Chatterjee et al. "SemEval-2019 task 3: EmoContext contextual emotion detection in text". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 39–48.

[16] Despoina Chatzakou, Athena Vakali, and Konstantinos Kafetsios. "Detecting variation of emotions in online activities". In: *Expert Systems with Applications* 89 (2017), pp. 318–332.

[17] François-Régis Chaumartin. "UPAR7: A knowledge-based system for headline sentiment tagging". In: *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics. 2007, pp. 422–425.

[18] Ciprian Chelba et al. "One billion word benchmark for measuring progress in statistical language modeling". In: *arXiv preprint arXiv:1312.3005* (2013).

[19] Jacob Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[20] Munmun De Choudhury, Michael Gamon, and Scott Counts. "Happy, nervous or surprised? classification of human affective states in social media". In: *Sixth International AAAI Conference on Weblogs and Social Media*. 2012.

[21] Munmun De Choudhury et al. "Predicting depression via social media". In: *Seventh international AAAI conference on weblogs and social media*. 2013.

[22] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[23] Paul Ekman. "An argument for basic emotions". In: *Cognition & emotion* 6.3-4 (1992), pp. 169–200.

[24] Paul Ekman, Wallace V Friesen, and Phoebe Ellsworth. *Emotion in the Human Face: Guide-lines for Research and an Integration of Findings: Guidelines for Research and an Integration of Findings*. Pergamon, 1972.

[25] R Ezhilarasi and RI Minu. "Automatic emotion recognition and classification". In: *Procedia Engineering* 38 (2012), pp. 21–26.

[26] Manaal Faruqui and Chris Dyer. "Community evaluation and exchange of word vectors at wordvectors. org". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2014, pp. 19–24.

[27] Frauke Friedrichs and Christian Igel. "Evolutionary tuning of multiple SVM parameters". In: *Neurocomputing* 64 (2005), pp. 107–117.

[28] Bharat Gaind, Varun Syal, and Sneha Padgalwar. "Emotion detection and analysis on social media". In: *arXiv preprint arXiv:1901.08458* (2019).

[29] Maria Gendron and Lisa Feldman Barrett. "Reconstructing the past: A century of ideas about emotion in psychology". In: *Emotion review* 1.4 (2009), pp. 316–339.

[30] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. "Hierarchical versus flat classification of emotions in text". In: *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics. 2010, pp. 140–146.

[31] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. "Prior and contextual emotion of words in sentential context". In: *Computer Speech & Language* 28.1 (2014), pp. 76–92.

[32] Kristina Gligorić, Ashton Anderson, and Robert West. "How constraints affect content: The case of Twitter's switch from 140 to 280 characters". In: *arXiv preprint arXiv:1804.02318* (2018).

[33] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report, Stanford* 1.12 (2009), p. 2009.

[34] Fréderic Godin et al. "Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations". In: *Proceedings of the Workshop on Noisy User-generated Text*. 2015, pp. 146–153.

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

[36] Garrison W Greenwood, Ajay Gupta, and Kelly McSweeney. "Scheduling tasks in multiprocessor systems using evolutionary strategies". In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE. 1994, pp. 345–349.

[37] Deepak Kumar Gupta and Asif Ekbal. "IITP: supervised machine learning for aspect based sentiment analysis". In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. 2014, pp. 319–323.

[38] Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. "Emotion detection in email customer care". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics. 2010, pp. 10–16.

[39] Umang Gupta et al. "A sentiment-and-semantics-based approach for emotion detection in textual conversations". In: *arXiv preprint arXiv:1707.06996* (2017).

[40] Mark Hall et al. "The WEKA data mining software: an update". In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.

[41] Nikolaus Hansen, Dirk V Arnold, and Anne Auger. "Evolution strategies". In: *Springer handbook of computational intelligence*. Springer, 2015, pp. 871–898.

[42] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)". In: *Evolutionary computation* 11.1 (2003), pp. 1–18.

[43] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2 (2001), pp. 159–195.

[44] Maryam Hasan, Emmanuel Agu, and Elke Rundensteiner. "Using hashtags as labels for supervised learning of emotions in twitter messages". In: *ACM SIGKDD Workshop on Health Informatics, New York, USA*. 2014.

[45] Toshiharu Hatanaka et al. "System parameter estimation by evolutionary strategy". In: *Proceedings of the 35th SICE Annual Conference. International Session Papers*. IEEE. 1996, pp. 1045–1048.

[46] Carroll E Izard. "The face of emotion." In: (1971).

[47] Hanhoon Kang, Seong Joon Yoo, and Dongil Han. "Senti-lexicon and improved Naıve Bayes algorithms for sentiment analysis of restaurant reviews". In: *Expert Systems with Applications* 39.5 (2012), pp. 6000–6010.

[48] Phil Katz, Matthew Singleton, and Richard Wicentowski. "Swat-mp: the semeval-2007 systems for task 5 and task 14". In: *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics. 2007, pp. 308–313.

[49] Jasleen Kaur and Jatinderkumar R Saini. "An analysis of opinion mining research works based on language, writing style and feature selection parameters". In: *Int. J. Adv. Netw. Appl* (2013).

[50] Jasleen Kaur and Jatinderkumar R Saini. "Emotion detection and sentiment analysis in text corpus: a differential study with informal and formal writing styles". In: *International Journal of Computer Applications* 101.9 (2014).

[51] Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A Calvo. "Evaluation of unsupervised emotion models to textual affect recognition". In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics. 2010, pp. 62–70.

[52] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. "Sentiment analysis of short informal texts". In: *Journal of Artificial Intelligence Research* 50 (2014), pp. 723–762.

[53] Agata Kołakowska et al. "Modeling emotions for affect-aware applications". In: *Information Systems Development and Applications* (2015), pp. 55–69.

[54] Zornitsa Kozareva et al. "UA-ZBSA: a headline emotion classification through web information". In: *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics. 2007, pp. 334–337.

[55] Akshi Kumar, Prakhar Dogra, and Vikrant Dabas. "Emotion analysis of Twitter using opinion mining". In: *Contemporary Computing (IC3), 2015 Eighth International Conference on*. IEEE. 2015, pp. 285–290.

[56] Akshi Kumar and Teeja Mary Sebastian. "Sentiment analysis on twitter". In: *IJCSI International Journal of Computer Science Issues* 9.4 (2012), p. 372.

[57] Matt Kusner et al. "From word embeddings to document distances". In: *International conference on machine learning*. 2015, pp. 957–966.

[58] Lidija Lalicic et al. "Emotional brand communication on Facebook and Twitter: Are DMOs successful?" In: *Journal of Destination Marketing & Management* 16 (2020), p. 100350.

[59] Jingsheng Lei et al. "Towards building a social emotion detection system for online news". In: *Future Generation Computer Systems* 37 (2014), pp. 438–448.

[60] Jasy Suet Yan Liew. "Fine-grained emotion detection in microblog text". In: (2016).

[61] Vincenzo Loia and Sabrina Senatore. "A fuzzy-oriented sentic analysis to capture the human emotion in Web-based content". In: *Knowledge-based systems* 58 (2014), pp. 75–85.

[62] Jean Louchet. "Stereo analysis using individual evolution strategy". In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 1. IEEE. 2000, pp. 908–911.

[63] Kim Luyckx et al. "Fine-grained emotion detection in suicide notes: A thresholding approach to multi-label classification". In: *Biomedical informatics insights* 5 (2012), BII–S8966.

[64] Zongyang Ma et al. "Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 2014, pp. 999–1008.

[65] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 746–751.

[66] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[67] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[68] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[69] Saif M Mohammad. "# Emotional tweets". In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics. 2012, pp. 246–255.

[70] Saif M Mohammad and Peter D Turney. "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon". In: *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics. 2010, pp. 26–34.

[71] Kaitlyn Mulcrone. "Detecting emotion in text". In: *University of Minnesota–Morris CS Senior Semminar Paper* (2012).

[72] Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012.

[73] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. "EmoHeart: conveying emotions in second life based on affect sensing from text". In: *Advances in Human-Computer Interaction* 2010 (2010), p. 1.

[74] Andreas Ostermeier, Andreas Gawelczyk, and Nikolaus Hansen. "Step-size adaptation based on non-local use of selection information". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 1994, pp. 189–198.

[75] M Ostertag, E Nock, and U Kiencke. "Optimization of airbag release algorithms using evolutionary strategies". In: *Proceedings of International Conference on Control Applications*. IEEE. 1995, pp. 275–280.

[76] James W Pennebaker et al. *The development and psychometric properties of LIWC2015*. Tech. rep. 2015.

[77] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[78] Matthew E Peters et al. "Deep contextualized word representations". In: *arXiv preprint arXiv:1802.05365* (2018).

[79] Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. "Comparing the performance of different NLP toolkits in formal and social media text". In: *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2016.

[80] Robert Plutchik. "A general psychoevolutionary theory of emotion". In: *Theories of emotion*. Elsevier, 1980, pp. 3–33.

[81] Alberto Purpura et al. "Supervised lexicon extraction for emotion classification". In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 1071–1078.

[82] Mohammed Abdel Razek and Claude Frasson. "Text-based intelligent learning emotion system". In: *Journal of Intelligent Learning Systems and Applications* 9.1 (2017), pp. 17–20.

[83] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann–Holzboog*. 1973.

[84] Ingo Rechenberg. *Evolutionsstrategie'94*. frommann-holzboog, 1994.

[85] Fuji Ren and Ning Liu. "Emotion computing using Word Mover's Distance features based on Ren_CECps". In: *PloS one* 13.4 (2018), e0194136.

[86] Kirk Roberts et al. "EmpaTweet: Annotating and Detecting Emotions on Twitter." In: *LREC*. Vol. 12. Citeseer. 2012, pp. 3806–3813.

[87] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "A metric for distributions with applications to image databases". In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 59–66.

[88] James A Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[89] James A Russell and Albert Mehrabian. "Evidence for a three-factor theory of emotions". In: *Journal of research in Personality* 11.3 (1977), pp. 273–294.

[90] Robert E Schapire. "A brief introduction to boosting". In: *Ijcai*. Vol. 99. 1999, pp. 1401–1406.

[91] Tom Schaul, Tobias Glasmachers, and Jürgen Schmidhuber. "High dimensions and heavy tails for natural evolution strategies". In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011, pp. 845–852.

[92] Klaus R Scherer and Harald G Wallbott. "Evidence for universality and cultural variation of differential emotion response patterning." In: *Journal of personality and social psychology* 66.2 (1994), p. 310.

[93] H-P Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.(Teil 1, Kap. 1-5)*. Birkhäuser, 1977.

[94] HP Schwefel. *Evolution and optimum seeking. 1995*. 1995.

[95] Shadi Shaheen et al. "Emotion recognition from text based on automatically generated rules". In: *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE. 2014, pp. 383–392.

[96] Phillip Shaver et al. "Emotion knowledge: further exploration of a prototype approach." In: *Journal of personality and social psychology* 52.6 (1987), p. 1061.

[97] James E Shepherd, David L McDowell, and Karl I Jacob. "Modeling morphology evolution and mechanical behavior during thermo-mechanical processing of semi-crystalline polymers". In: *Journal of the Mechanics and Physics of Solids* 54.3 (2006), pp. 467–489.

[98] Rajdeep Singh, Roshan Bagla, and Harkiran Kaur. "Text analytics of web posts' comments using sentiment analysis". In: *Computing and Communication (IEMCON), 2015 International Conference and Workshop on*. IEEE. 2015, pp. 1–5.

[99] Marijn F Stollenga et al. "Deep networks with internal selective attention through feedback connections". In: *Advances in neural information processing systems*. 2014, pp. 3545–3553.

[100] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. "The general inquirer: A computer approach to content analysis." In: (1966).

[101] Carlo Strapparava and Rada Mihalcea. "Learning to identify emotions in text". In: *Proceedings of the 2008 ACM symposium on Applied computing*. ACM. 2008, pp. 1556–1560.

[102] Carlo Strapparava and Rada Mihalcea. "Semeval-2007 task 14: Affective text". In: *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics. 2007, pp. 70–74.

[103] Carlo Strapparava, Alessandro Valitutti, et al. "Wordnet affect: an affective extension of wordnet." In: *Lrec*. Vol. 4. 1083-1086. Citeseer. 2004, p. 40.

[104] Yu Sun, Zhi Ping Li, and Yao Wen Xia. "Emotional Interaction Agents in Intelligent Tutoring Systems". In: *Applied Mechanics and Materials*. Vol. 347. Trans Tech Publ. 2013, pp. 2682–2687.

[105] Martin D Sykora et al. "Emotive ontology: Extracting fine-grained emotions from terse, informal messages". In: (2013).

[106] Shabnam Tafreshi and Mona Diab. "GWU NLP Lab at SemEval-2019 Task 3: EmoContext: Effective Contextual Information in Models for Emotion Detection in Sentence-level in a Multigenre Corpus". In: *arXiv preprint arXiv:1905.09439* (2019).

[107] Jianhua Tao. "Context based emotion detection from text input". In: *Eighth International Conference on Spoken Language Processing*. 2004.

[108] Mike Thelwall et al. "Sentiment strength detection in short informal text". In: *Journal of the Association for Information Science and Technology* 61.12 (2010), pp. 2544–2558.

[109] Antoine J-P Tixier, Michalis Vazirgiannis, and Matthew R Hallowell. "Word embeddings for the construction domain". In: *arXiv preprint arXiv:1610.09333* (2016).

[110] Vikas Tripathi, Bhasker Pant, and Vijay Kumar. "CNN Based Framework for Sentiment Analysis of Tweets". In: ().

[111] S Voeffray. "Emotion-sensitive human-computer interaction (HCI): State of the art-Seminar paper". In: *Emotion Recognition* (2011), pp. 1–4.

[112] Wenbo Wang et al. "Harnessing twitter" big data" for automatic emotion identification". In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE. 2012, pp. 587–592.

[113] Keigo Watanabe et al. "Path planning for an omnidirectional mobile manipulator by evolutionary computation". In: *1999 Third International Conference*

on Knowledge-Based Intelligent Information Engineering Systems. Proceedings (Cat. No. 99TH8410). IEEE. 1999, pp. 135–140.

[114] Daan Wierstra et al. "Natural evolution strategies". In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 3381–3387.

[115] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. "Emotion classification using web blog corpora". In: *Web Intelligence, IEEE/WIC/ACM International Conference on*. IEEE. 2007, pp. 275–278.

[116] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *arXiv preprint arXiv:1906.08237* (2019).

[117] Peter Zachar and Ralph D Ellis. *Categorical versus dimensional models of affect: a seminar on the theories of Panksepp and Russell*. Vol. 7. John Benjamins Publishing, 2012.

[118] Yukun Zhu et al. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27.