DOCTOR OF PHILOSOPHY

Automatic Emotion Recognition in English and Arabic text

Al-Mahdawi, Amer

Award date:
2019

Awarding institution:
Bangor University

Automatic Emotion Recognition in English and Arabic text

by

Amer Al-Mahdawi

A thesis presented for the degree of
Doctor of Philosophy

School of Computer Science and Electronic Engineering

Bangor University

Supervisored by

William J. Teahan

April 2019

Acknowledgements

Sometimes it is hard to express the gratitude felt for another, especially for my supervisor Dr. William J. Teahan. I acknowledge, with gratitude, my debt of thanks to him for encouraging my research and allowing me to grow as a research scientist. His advice on both research as well as on my career have been invaluable and very much appreciated.

I would also like to thank my beloved wife Dr. Abeer Ali for her understanding and love during the past few years. Her support and encouragement were in the end what made this dissertation possible. Without her wholehearted support, I would not be able to finish my PhD studies.

My deepest emotions are for my children Fatima and Fahad, I know you missed me, as I do always. I have now finished my PhD degree, and I will never ever be away again.

Dedication

I would like to dedicate this thesis to the pure souls of my Mother and Father, my beloved wife Dr Abeer, my two eyes Fatima and Fahad, my brothers and my sister...

Abstract

This study investigated the automatic recognition of emotion in English and Arabic text. We perform experiments with a new method of classification for recognising emotions using the Prediction by Partial Matching (PPM) character-based text compression scheme. These experiments involve both document level classification (whether a text of document is emotional or not) and also fine-grained classification such as recognising Ekman's six basic emotions (Anger, Disgust, Fear, Happiness, Sadness, Surprise). Experimental results with three English datasets (the LiveJournal's blogs dataset, Aman's blogs dataset, and Alm's fairy tales dataset) show that the new method significantly outperforms the traditional word-based text classification methods. The results show that the PPM compression-based classification method is able to distinguish between emotional and non-emotional text with high accuracy, between texts involving Happiness and Sadness emotions (with 79.1% accuracy for Aman's dataset and 76.9% for Alm's datasets) and texts involving Ekman's six basic emotions for the LiveJournal dataset (87.4% accuracy). Results also show that the method outperforms traditional feature-based classifiers such as Naïve Bayes and SMO in most cases in terms of accuracy, precision, recall and F-measure.

In order to see how well the classifier performs on another language not related to English and also in order to create another Arabic benchmark corpus for future emotion classification experiments, we created a new Iraqi Arabic Emotion Corpus (IAEC) dataset annotated according to Ekman's basic emotions. This dataset is composed of Facebook posts written in the Iraqi dialect. We evaluated the quality of this dataset using four external judges which resulted in an average inter-annotation agreement of 0.751. We then explored six different supervised machine learning methods to test the new dataset. We used standard Weka classifiers ZeroR, J48, Naive Bayes, Multinomial Naive Bayes for Text and SMO. We compared these results with our compression-based classifier PPM. Our study reveals that the PPM

classifier significantly outperforms the other classifiers for the new dataset achieving the highest results in terms of accuracy, precision, recall, and F-measure.

We also designed and investigated another new classification technique motivated by information divergence to recognize Ekman's emotions in text. We used the three datasets written in the English Language and the one in the Arabic Language to evaluate the new method. The new method was able to achieve a better result for Alm's dataset in terms of accuracy, precision, recall and F-measure than PPM and standard Weka classifiers. The new method also outperforms all standard Weka classifiers for all four datasets. Finally, these results show that our proposed technique is promising as an alternative technique for English and Arabic text categorization in general.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recognising a person's emotional state is possible by using such cues as their facial expressions, their voice, the language they use or their behaviour. Written texts such as emails, texts, blogs and tweets now make up a significant amount of communication between people because of the growth of social media. Therefore, being able to recognise the emotional state of the person or persons producing the text would be very beneficial in many situations. For example, recognising certain types of emotion might help to predict when someone might commit a crime, or a terrorist act (Yang et al., 2012) or provide early risk detection for the Internet, particularly in health and safety areas such as detecting early signs of depression, anorexia or suicidal inclinations (Ramiandrisoa et al., 2018). Another area where emotion recognition is useful is in helping to build more affective interfaces where identifying the emotion of the user can allow the computer to respond more effectively. In customer care service, emotion recognition helps advertisers to pick up data about how much satisfied their clients are, [what parts of their service should be enhanced or reconsidered, to thus make a solid association with their end clients (Gupta et al., 2013)]. In Human-Computer Interaction, the computer can monitor the user's emotions to recommend appropriate music or motion pictures (Voeffray, 2011). In e-learning applications, the tutoring system can

settle on teaching materials, based on the client's emotions and mental state.

1.2 Motivation

The motivation for this study is to design and develop novel approaches that are effective at automatically recognising emotions in text. As well, due to the lack of a publicly available dataset in Iraqi Arabic labelled in the six emotions of the Ekman's classification, a new corpus will be developed to help facilitate the study.

1.3 Research Questions

The research questions of this study are as follows:

1. Can new methods be developed that are more effective for emotion recognition than existing approaches?

2. Would the Prediction by Partial Matching (PPM) compression-based method perform better than other common methods for emotion recognition?

3. What is the most effective PPM model for emotion recognition? For example, does the order of the PPMD model affect the results of the emotion classification?

4. Can these methods also be applicable to a language non-related to English (Arabic) and how effective are these methods?

1.4 Aim and Objectives

In order to seek answers to the research questions, the general aim of this study is to develop novel methods for automatic emotion recognition of text. So, the specific study objectives are as follows:

1. To apply the Prediction by Partial Matching (PPM) compression-based classification method to the problem of automatically recognising emotions in text.

2. To evaluate and validate the PPM method using different standard datasets and compare with other results achieved by other traditional classifiers (see chapters 3 and 4).

3. To create a new Iraqi Arabic Emotion Recognition dataset (IAEC).

4. To evaluate and validate the adapted PPM method using the IAEC, and compare the result of PPM with other traditional classifiers.

5. To develop and design a further new method for automatically recognising emotions in text.

6. To evaluate the newly devised method on the English and Arabic datasets.

1.5 Contributions

The study conducted in this thesis has provided several contributions. Strong evidence is presented that compression-based methods can be used for effective emotion recognition in text. This proof includes the effectiveness of the new methods on different datasets and different types of text.

The significant contributions of this study are as follows:

1. The feasibility of using the Prediction by partial matching (PPM) approach for automatically recognising emotions in English and Arabic text has been investigated.

2. A new approach has been devised for automatically recognising emotions in text based on information divergence.

3. The new methods for emotion recognition have been applied to English and Arabic emotion datasets.

4. A new emotion recognition corpus has been composed for the Arabic language in the Iraqi dialect.

1.6 Publications

Two conference papers based on this study have already been published. Table 1.1 shows the specific papers which are relate to this study.

The first paper entitled "Emotion Recognition in Text using PPM", describes the new method of classification using PPM based compression. The paper investigates applying the PPM to classify emotions in text. The emotions used in this paper were defined by Ekman who used facial expression to classify emotions. The classification in this paper was at the document level. Experiments show that PPM achieved better results when balancing the size of the training files of emotion classes through the process of training the PPM classifier. This paper was the first to report that balancing the size of training file will improve the result of classification using PPM. The insights gained from this paper have provided an important basis for this thesis as discussed in chapter 3.

The second paper, entitled "Automatically Recognising Emotions in Text Using Prediction by Partial Matching (PPM) Text Compression Method" describes the new proposed method of classification (PPM) to classify Ekman's emotions in blogs for two datasets used, and in fairy tales for the third dataset. The experimental results show that PPM outperforms other classifiers in terms of accuracy, precision, recall and F-measure. Different types of classifications have been applied to test how effective PPM is. The first classification was to classify emotion text vs. non-emotion text, the second classification was to classify *Happiness* vs *Sadness* text, while the third classification was to classify Ekman's emotions. The insights gained from this paper has also provided an important basis for this thesis as discussed in chapter 3.

A third paper entitled "A New Arabic dataset for emotion recognition" describes the process of creation the Iraqi Arabic Emotion Corpus (IAEC). This corpus consists of six emotions based on Ekman's emotions. This paper

also describes the process of applying the new proposed method PPM to classify Ekman's emotions in Arabic text. It also provides a description of the annotation of this corpus and also provides the evaluation process for the IAEC. Experimental results show the results of applying different classifiers to classify Ekman's emotion in the Arabic Text. The experimental results show a comparison of the results achieved by all classifiers that used the IAEC. The insights gained from this paper has also provided an important basis for this thesis as discussed in chapter 5.

The fourth paper entitled "Emotion Recognition for Text Using Information" currently being drafted describes a method for emotion recognition based on information divergence. The paper also provides experimental results that compare the results of information divergence method with other classifiers. The insights gained from this paper again have provided an important basis for this thesis as discussed in chapter 4.

Table 1.1: Publications that relate to this study.

No.	**Publication name**
1	Almahdawi, A. and Teahan, W. J. (2017). Emotion recognition in text using ppm. In *SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 149-155, Cambridge, UK. Springer.
2	Almahdawi, A. and Teahan, W. J. (2018). Automatically recognizing emotions in text using prediction by partial matching (ppm) text compression method. In *International Conference on New Trends in Information and Communications Technology Applications*, pages 269-283, Baghdad, Iraq. Springer
3	Al-Mahdawi, A. and Teahan, W. J. (2019). Emotion recognition for text using information divergence. *Journal of IEEE Transactions on Affective Computing*. Pending.
4	Almahdawi, A. J. and Teahan, W. J. (2019). A new Arabic dataset for emotion recognition. In *Intelligent Computing-Proceedings of the Computing Conference*, pages 200-216. Springer. London.

1.7 Thesis Outline

This thesis in organized as follows. **Chapter 1** provides the motivation for emotion recognition, the research questions, the research aim and objectives, the research contributions and the publications.

Chapter 2 provides the background and related work for this thesis in two parts. Part one discusses the background to the work: an introduction to affective computing, followed by introduction to text categorization. This is followed by an introduction to the Arabic language. Arabic calligraphy is also introduced followed by different types of the Arabic language and how it is encoded. Part two reviews the general concepts of emotion recognition and data compression. In this part, the different theories of emotions are investigated. This is followed by an introduction to sentiment analysis. The computational methods for emotion recognition are also introduced in details. Text categorization using Prediction by Partial Matching (PPM) is described and relative entropy introduced. Finally the available English datasets for emotion recognition are also described.

Chapter 3 represents the first practical part of the thesis. This chapter explains how emotions can be automatically recognised in text using Prediction by Partial Matching (PPM). The following methods PPM, ZeroR, Naïve Bayes, SMO are used as classifiers to classify text using Ekman's emotion, and to perform Happiness vs Sadness classification and Emotional vs Non emotional classification. The comparison among the classification results of these classifiers are investigated. Various orders of PPM have also been used and investigated. The results show that PPM outperforms other traditional classifiers and also how it can be used effectively for automatically recognising emotions in text.

Chapter 4 represents a new classification method based on information divergence. In this chapter, Ekman's emotion classification has been applied on three English datasets using information divergence based on relative entropy. Different variations of the approach have been devised: ID1, ID2 and ID3. The results show the new classifier performs effectively and its results are very competitive when compared to other classifiers.

Chapter 5 describes the creation of the new Iraqi Arabic Emotion recognition dataset (IAEC). This chapter introduces the available platforms of social media that people use to express their emotions for different life events. This chapter also introduces the process of composing corpus posts, followed by the process of annotating the blogs of the new created corpus. Then the evaluation of the new corpus is described. Experimental results for automatically recognising Ekman's emotion from the new corpus using all the above classifiers has been described.

Chapter 6 provides the conclusions. It reconsiders the results with regards to the research questions and aim and objectives. The limitations and future work are also discussed.

Chapter 2

Background and Related Work

The purpose of this chapter is to give a contextual explanation of the background and related work. This includes affective computing, text categorization and the Arabic language in the background section. The related work section includes the following topics that address the research questions and aim and objectives: theories of emotions, sentiment analysis, levels of analysis, computational methods for emotion recognition, machine learning methods, entropy, text classification using prediction by partial matching (PPM), relative entropy and English datasets for emotion recognition.

2.1 Background

This section is organized as follows: subsection 2.1.1 provides an overview of Affective computing. Subsection 2.1.2 provides a brief explanation of text categorization. Subsection 2.1.3 provides an overview of the Arabic Language and the geographical spread of the Arabic Language; subsection 2.1.4 provides an overview of Arabic characters and how to they are written; subsection **??** provides an overview of the Arabic calligraphy styles; subsection 2.1.5 provides the types of Arabic Language and finally, subsection 2.1.6 provides an overview of the Arabic encoding methods.

2.1.1 Affective Computing

Affective computing is the study of the design of computer systems that can understand and respond appropriately to human affects (elKaliouby, 2017). Research on affect or emotion occurred as long ago as the 19th century (James, 1884). AI is increasingly being used in various devices such as smart phones, tablets and laptops. But we still need to develop these devices to interact with human emotions. Affective computing will often use an affect model based on the training data from different sensors that collect information and build a system capable of perception, understanding and interpretation of the feelings of a human (Tao and Tan, 2005). Affective computing and sentiment analysis are elements of advanced AI (Minsky, 2007). recognising a person's emotional state is possible using such cues as their facial expressions, their voice, the language they use or their behaviour.

Written text such as emails, texting, blogs and tweets now makes up a significant amount of the communication between people because of the growth of social media. Therefore, being able to recognise the emotional state of the person or persons producing the text would be very beneficial in many situations. For example, recognising certain types of emotion might help to predict when someone might commit a crime or a terrorist act (Yang et al., 2012) or provide early risk detection for the Internet, particularly in health and safety areas such as detecting early signs of depression, anorexia or suicidal inclinations (Ramiandrisoa et al., 2018). Another area where emotion recognition is useful is in helping to build more affective interfaces where identifying the emotion of the user can allow the computer to respond more effectively.

Computational linguistics is an interdisciplinary field that draws on different fields such as statistics, psychology, cognitive science and natural language from a computational view (Uszkoreit, 2000). Computational linguistics is concerned with the understanding of written text from a computational view, and to what extent the language being considered is the mirror of the mind of the speaker, so it provides insight into their thoughts. Since the language is the way of communication between people, so people can interact

with computers through text or voice, and much research has been applied in this area to make computers understand human language and respond accordingly (Schubert, 2015).

2.1.2 Text Categorization

Text categorization (also called text classification) is the process of assigning a label or labels for a new document automatically based on predefined categories as produced by training on manually labelled documents (Yang and Liu, 1999).

According to Sebastian (Sebastiani, 2005), text mining recently has received more importance due to the increasing in the number of available electronic documents and from different sources. The goal of text mining is to extract useful information from available text resources and is applicable to various applications such as information retrieval, classification, summarization, Natural Language Processing (NLP), data mining, and machine learning techniques to automatically discover or classify different patterns of different documents.

Text categorization has been applied in different fields, such as document indexing depending on a controlled vocabulary, document filtering, word sense disambiguation, automated meta-data generation, and any application requiring document organization through adaptive document dispatching (Sebastiani, 2002).

The most popular approach for text categorization in the 1980s involved knowledge engineering such as defining a set of rules to encode expert knowledge about how to classify documents under the specified classes. Research in the 1990s started adopting machine learning methods which involved the process of automatically composing an automatic classifier by learning from a set of pre-labelled documents, which belong to certain categories. This approach achieved better accuracy than human experts (Sebastiani, 2002).

Currently, text categorization lies between machine learning and information retrieval and it can share a number of characteristics with other tasks such as information extraction and text mining (Knight, 1999; Basili and

Pazienza, 1997). Text categorization can be considered as an instance of text mining (Sebastiani, 2002).

Emotion recognition (or emotion classification) in text is a specific type of text categorisation. Shaheen et al. (2014) state that there are two types of emotion classification: those that are coarse grained and those that are fine grained. Coarse grained classification tries to identify positive and negative emotions in the text as it occurs for sentiment analysis. Fine grained classification on the other hand tries to identify more than just the two positive and negative categories by identifying more specific emotions (such as *Happiness* and *Sadness*). Ekman has stated that there are six basic emotions—*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise*—as these emotions are common to all cultures (Ekman, 1992; Ekman, 1999). Evidence for this was obtained by examining brief facial expressions that occur when a person is trying to conceal an emotion either deliberately or unconsciously.

The following sections will now review the background to the Arabic language, as this is relevant for the experiments that were done in chapter 5.

2.1.3 Arabic Language

Arabic language (العربية) is one of the most widely spoken languages more than 290 million native Arabic people, whilst about one billion people have used Arabic language as a second language (Simons and Fennig, 2018). According to cuneiform references, the word Arab "A - ri - bi " was used to depict the people that were living from the Anti Lebanon mountains in the west to Mesopotamia in the east, from Sinai in the south to north west Arabia (Ephal, 1982). Later according to Greek and Persian references, Arab presence was mentioned across the area of north Arabia and the Fertile Crescent (Macdonald, 2009).

According to table 2.1, the most widely spoken language around the world is Mandarin Chinese, the second most widely spoken language is English, and Arabic comes at rank five of the most widely spoken languages around the world with 422 million people who speak the language.

The Arabic language is a central Semitic language. It first appeared in

Rank	Language	Speakers (Millions)
1	Mandarin Chinese	1090
2	English	983
3	Hindustani	544
4	Spanish	527
5	Arabic	422
6	Malay	281
7	Russian	267
8	Bengali	261
9	Portuguese	229
10	French	229

Table 2.1: Top ten spoken languages around the world in millions of speakers (Simons and Fennig, 2018).

the era of Iron Age north-western Arabia and is now the lingua-franca of the Arab world (Putten, 2017). Most Muslims speak the Arabic language because it is the language associated with the Qur'an which is the holy book of Islam, and therefore it is religious language for all Muslims. The form of the language found in the Qur'an is Classical Arabic which is uniform all over the Arab world. There are also numerous spoken dialects of colloquial Arabic. The main dialect groups are Iraqi, Syrian, Egyptian, and North African being all influenced by Classical Arabic (Simons and Fennig, 2018).

The Arabic language can be considered as a unifying force for 22 countries from Bahrain in the east to Mauritania in the west and from Iraq in the north to Somalia in the south (Rasheed, 2008). The Arabic language influences other languages directly or indirectly such as Malay, Urdu, Persian, and Kurdish. The Arabic Language has also influenced languages in the west such as Portuguese, Sicilian and Spanish. The Arabic language has borrowed some words from these languages (Weekley, 2012).

2.1.4 Arabic Characters

The Arabic Language has twenty eight letters and is read from right to left. The Arabic language developed a system for writing based on Aramic and

Nabataean scripts whose letters are shown below.

أ ب ت ث ج ح خ د ذ ر ز ش ص ض ط ظ ع غ ف ق م ن ه و ي

Arabic has three vowel letters "أ ي و" (Elbeheri et al., 2006), while the others are consonants. Words in Arabic are separated by spaces. In contemporary centuries, the system of Arabic writing has been developed to the "Thamodi" style following the "Musnad" style (Carter, 1998). The style of Arabic numbers also changed from 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0 to an Indian style ١، ٢، ٣، ٤، ٥، ٦، ٧، ٨، ٩، ٠. Arabic has two genders: Feminine and Masculine (Cherif et al., 2015). Arabic has singular form, dual form, and plural form as well. For example 'he' in Arabic is **هو** and 'she' in Arabic is **هي** . **هما** used for dual masculine and feminine, and uses **هم** for masculine plural and uses **هن** for feminine plural.

Arabic letters can be written fully vowelised, not vowelised, and partially vowelised. For example, the sentence "In the Name of God, the Most Gracious, the Most Merciful... " is vowelised "بِسْمِ آللهِ آلرَحمٰنِ آلرَحيمِ" or it can be written as not vowelised as "بسم الله الرحمن الرحيم".

2.1.5 Types of Arabic language

The modal written system of Arabic text can be divided into three categories as Arabic is a "tri-glossic" language. The first type is Classical Arabic (CA). The second type is Modern Standard Arabic (MSA) (Najeeb et al., 2014), and the third is Dialectal Arabic (Arabic, 2015).

2.1.5.1 Classical Arabic

As mentioned before, this is the old style of Arabic which is the language of the holy Qur'an and classical literature. The difference between CA and MSA is by the vocabulary and style. Many people rely on translation to understand the meaning of the different vocabulary used for CA.

2.1.5.2 Modern Standard Arabic

MSA is the universal Arabic language that is understood by all Arabic speakers. The other name of MSA in Arabic is اللغة العربية الفصحى that is used by most TV shows, historical series, lectures and so on.

2.1.5.3 Dialectal Arabic

Spoken Arabic is called "Colloquial Arabic" or "Arabic dialects" as well. Spoken Arabic differs from Modern Standard Arabic in the following ways:

- It has a simple grammatical structure.
- The pronunciation of some letters are different from Modern Standard Arabic, and are also pronounced differently from one dialect to another.
- Depending on the Arabic dialect, some words or expressions have the same or different meaning, when they express their sense of humour or use a common expression.
- Dialects contain many words and expressions that do not exist or even have equivalent words or expressions in Modern Standard Arabic.

Arabic dialects can be categorized into the following: Iraqi Arabic; North African Arabic including Morocco, Algeria, Libya, and Tunisia; Yemeni Arabic including Yemen and south western Saudi Arabia; Hassaniya Arabic including Mauritania; Najdi Arabic including central Saudi Arabia; Egyptian Arabic; Hejazi Arabic including Western Saudi Arabia; Levantine Arabic including Palestine, Jordan, Lebanon, and Syria; Gulf Arabic including Oman, the U.A.E, Bahrain, Kuwait, and Qatar (Arabic, 2015).

2.1.6 Arabic Encoding Methods

As stated, the Arabic language is unlike the English language as it does not have upper and lower case characters and it is written from right to left.

There are three methods used to encode the Arabic language. These methods are shown in Figure 2.1 and discussed in the following three sections.

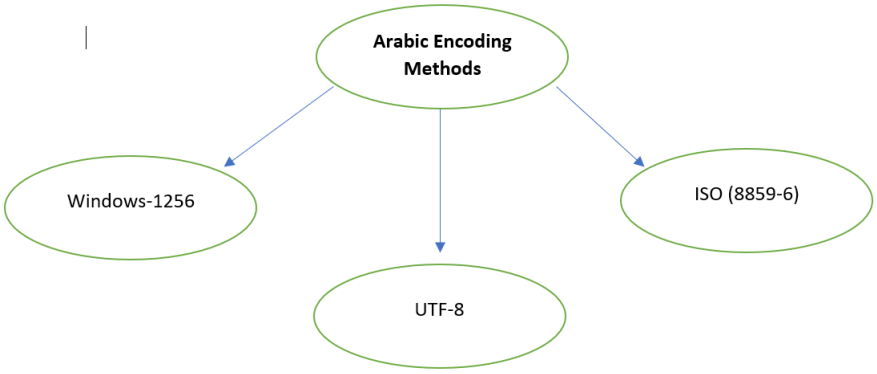


Figure 2.1: The Arabic Encoding Methods.

2.1.6.1 ISO (8859-6) Arabic Coding Standard

The International Organization for Standardization (ISO) is a body run by the European Computer Manufacture Association. The ISO 8859-6 character encoding was designed for the Arabic Language. It uses an 8-bit character scheme to encode Arabic characters. This scheme does not support all the supplementary Arabic characters and also does not support other languages based on Arabic characters such as Kurdish, Persian, and Urdu. Figure 2.2 illustrates the ISO 8859-6 Arabic encoding standard (International, 2000).

2.1.6.2 Windows-1256 Encoding

Windows-1256 encoding was developed by Microsoft. It uses an 8-bit character encoding system to encode Arabic characters and other languages based on Arabic characters such as Kurdish, Persian, and Urdu. Windows-1256 encodes more forms of Arabic characters, but it still does not support all supplementary characters, and besides it is not compatible with ISO 8859-6 Arabic encoding system. Table 2.3 illustrates the scheme (Microsoft, 2018).

2.1.6.3 UTF-8 Encoding

The UTF-8 encoding scheme is the most popular encoding method used for internet websites and applications including Facebook, Twitter, Youtube and Google. Over 115 million documents on the Internet use UTF-8 encod-

Figure 2.2: The ISO 8859-6 Arabic Encoding Scheme (International, 2000).

ing, which represents 84.6% of internet websites (BuiltWith, 2009). Google recorded the encodings of websites from 2001 and 2012. Figure 2.4 shows the usage of the main encoding schemes on the web back in 2012 (Davis, 2012). Figure 2.5 illustrates how the UTF-8 encoding scheme dominates

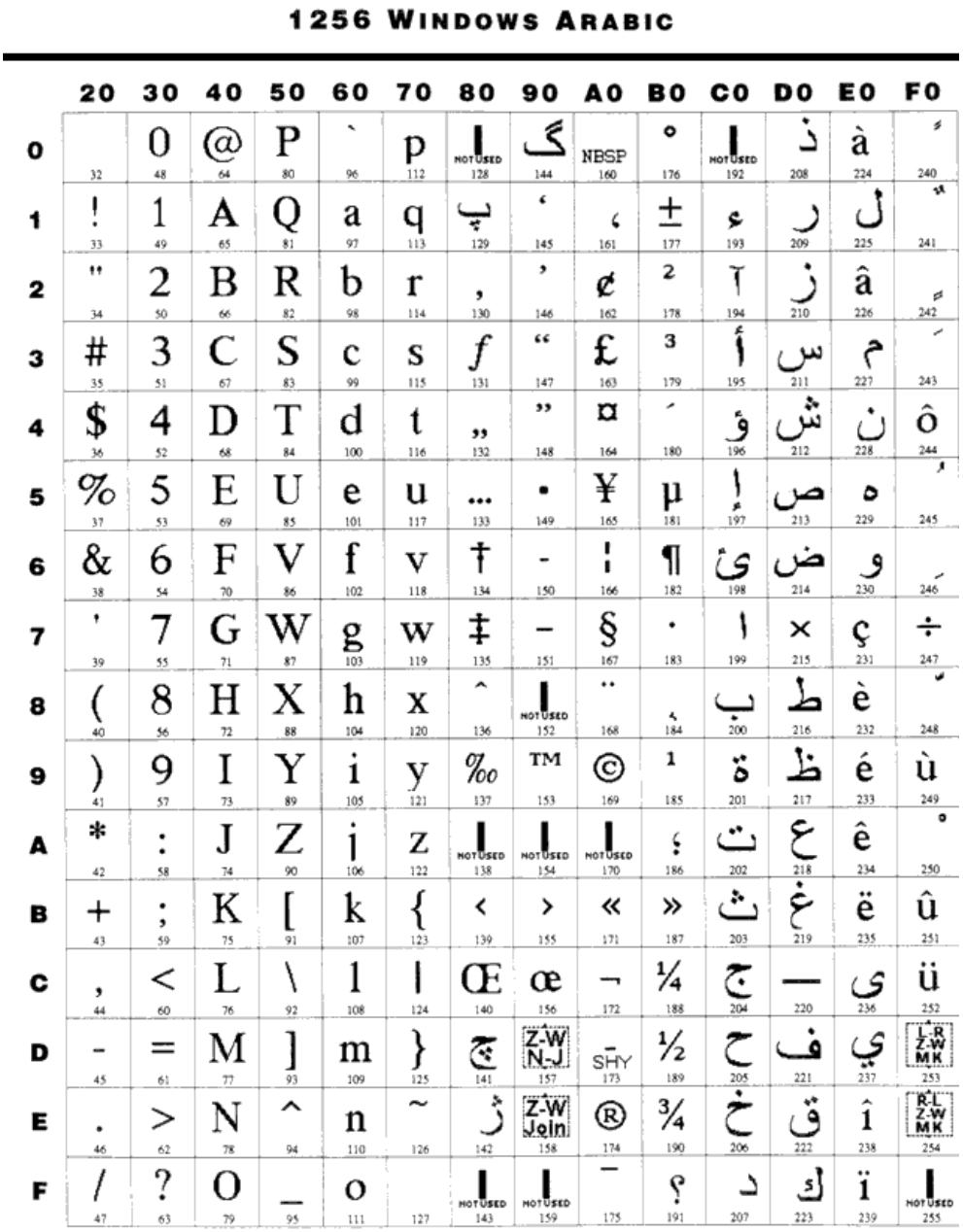


Figure 2.3: The Windows 1256 Arabic Encoding Scheme (Microsoft, 2018).

other encoding schemes for encoding internet websites.

The UTF-8 encoding scheme is a multi-byte encoding system. It uses the ASCII character encoding (0-127) to encode English characters using one byte. The UTF-8 uses up to four bytes to represent characters of other

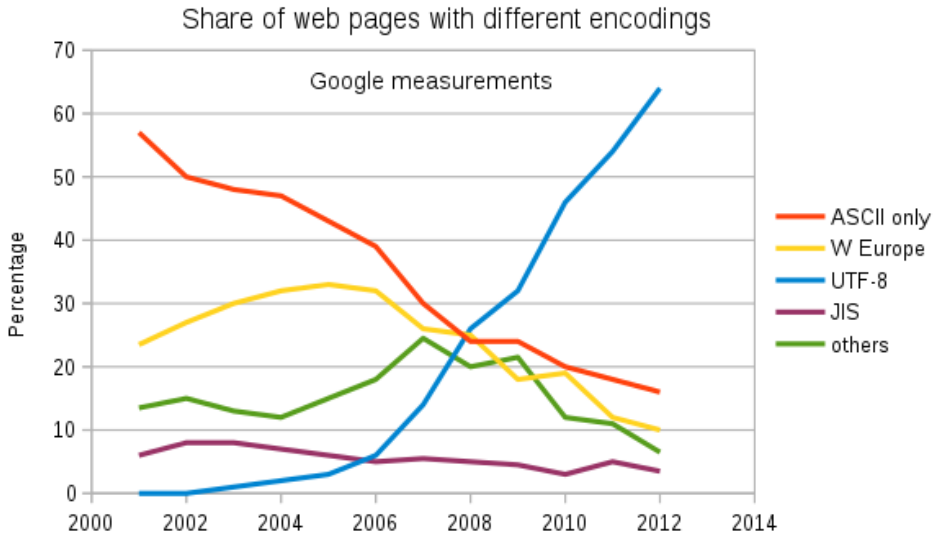


Figure 2.4: The usage of the main Encoding Schemes (Davis, 2012).

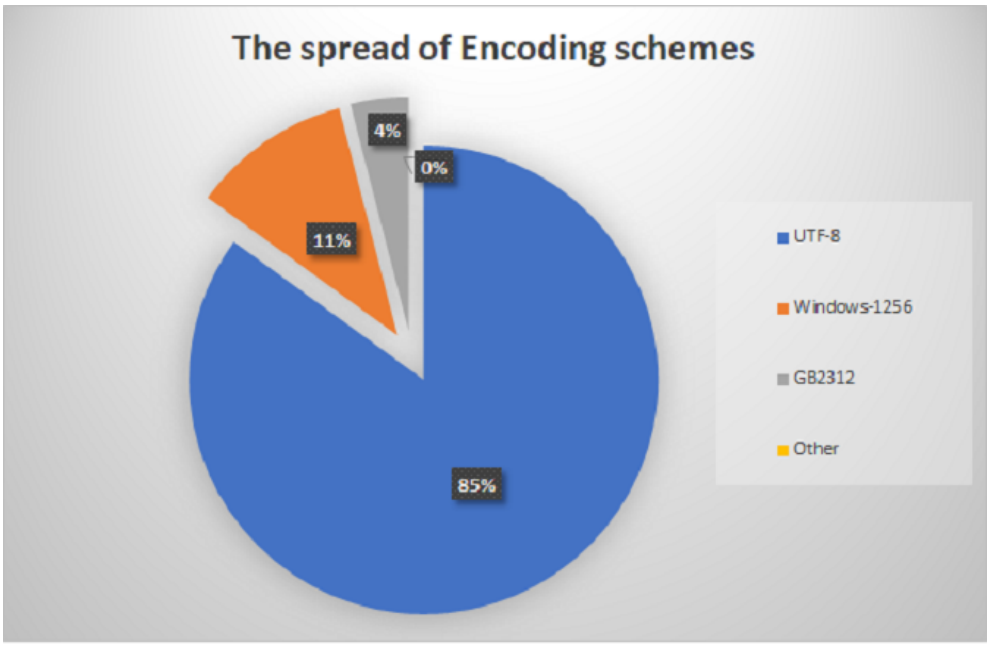


Figure 2.5: The spread of the UTF-8, windows 1256 encoding schemes (BuiltWith, 2009).

languages that need more bytes to represent their characters such as Arabic, Chinese, and Japanese. In the Arabic language, the UTF-8 scheme encodes

each Arabic character and the supplementary characters by using two bytes.

2.1.7 Buckwalter Arabic Transliteration

The Buckwalter Arabic transliteration is a transliteration system that uses standard Latin encoding to represent Arabic characters for computers (Buckwalter, 1990). Many publications have used the Buckwalter transliteration for natural language processing. The Buckwalter is a strict transliteration character to character system and the resulting text is not easy to read.

2.2 Related Work

In this section, prior research that is related to this study will be described, focusing on sentiment analysis, emotion recognition, and moods analysis.

Natural language processing (NLP) is a domain of study into the processing of human natural language for different applications such as machine translation, speech recognition, text processing, and artificial intelligence (Chowdhury, 2003).

Statistical NLP uses probabilistic, statistical and stochastic methods to solve problems in NLP such as statistical parsing, using various methods such as stochastic context-free grammar and hidden Markov models. Difficulties became apparent when processing long sentences that were highly ambiguous and required realistic grammars that could offer many possible analyses. Markov models and corpus-based methods were investigated for statistical NLP (Manning and Schutze, 1999).

This section is organized as follows: subsection 2.2.1 provides an overview of the theories of emotions, subsection 2.2.2 describes sentiment analysis, subsection 1.1 provides the motivation for emotion recognition in text, subsection 2.2.3 provides the levels of specifying the sentiment in the text, subsection 2.2.5 describes the computational methods used in extracting emotions in text, subsection 2.2.6 describes the related classifiers used in emotion recognition in text, subsection 2.2.7 describes the theory of entropy and how it is used in text categorization, subsection 5.5.2 explains how the PPM algo-

rithm works in text categorization based on entropy, subsection 2.2.9 explains how relative entropy works in text categorization, subsection 2.1.7 provides a brief description of Buckwalter translitration and how it used to change Arabic character to Latin characters and section 2.3 describes the available English emotion datasets.

2.2.1 Theories of Emotions

It is important to mention and describe how emotion recognition started and what are the basics of the approach. There have been much recent research in different fields such as psychology, social science, and others that have investigated emotions. This includes emotions used by humans in communications through facial expression, gesture, the volume of speech and so on. There have also been many investigations in order to categorise these emotions correctly (Picard, 1997).

Much research has been carried out on human emotions (Plutchik and Kellerman, 1980; Ekman, 1992; Frijda, 1986; Izard, 1971; James, 1884; Parrott, 2001) and others. Table 2.2 lists these researchers and how they categorised basic emotions. These are now used for most of the research into emotion recognition in text.

Table 2.2: Emotions as they were categorised by researchers often used as the basis for emotion recognition research.

(Plutchik, 1980)	**(Ekman, 1992)**	**(Parrott, 2001)**
Acceptance	Anger	Anger
Anger	Disgust	Fear
Anticipation	Fear	Hate
Disgust	Happiness	Joy
Fear	Sadness	Sadness
Joy	Surprise	Surprise
Sorrow		
Surprise		

In light of the above, Cabanac defines emotions as "any mental experience with a high intensity and high hedonic content (*pleasure/displeasure*)" (Ca-

banac, 2002). Shelke confirms emotion is important to human life especially in decision making and it can be considered as a fundamental aspect of human lives (Shelke, 2014). There are many definitions by different psychologists depending on their opinion about basic emotions as shown in table 2.2. The purpose of this table is to highlight the wide variations that exist among researchers. The categorisation of emotions depends on the expression used for discriminating emotions and how these map to emotions.

As shown in table 2.3, Mowrer proposed two basic states of emotions which are: *pleasure* and *pain*. According to Mowrer, these two emotions can be further extended to include *fear*, *relief*, *disappointment*, and *hope* (Mowrer, 1960). Watson defines three basic emotional states, *fear*, *love*, and *rage* by considering emotion as hardwired (Watson, 1930). Four basic emotional states were proposed by Panksepp, which are: *expectancy*, *fear*, *rage*, and *panic* (Panksepp, 1982); and Kemper also proposed *anger*, *depression*, *fear*, and *satisfaction* as emotional states (Kemper, 1987). James defines *fear*, *grief*, *love*, and *rage* as an emotional state by involving movement to determine emotions (James, 1884), while Gray proposes four emotional states which are: *anxiety*, *joy*, *rage* and *terror* by considering emotions are hardwired (Gray, 1982).

Oatley and Johnson-Laird build their theory on the importance of *anger*, *anxiety*, *disgust*, *happiness*, and *sadness* (Oatley and Johnson-Laird, 1987). Ekman et al. suggest six emotional states, which are: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise* by examining facial expression (Ekman et al., 1972). Frijda defines *desire*, *happiness*, *interest*, *sorrow*, *surprise*, and *wonder* based on action readiness (Frijda, 1986). McDougall suggests seven states of emotion: *anger*, *disgust*, *elation*, *fear*, *subjection*, *tender-emotion*, and *wonder* by employing instincts (McDougall, 1926). More recently, Plutchik defines *acceptance*, *anger*, *anticipation*, *disgust*, *joy*, *fear*, *sadness*, and *surprise* as eight emotional states by adopting biological processes (Plutchik, 1980). Tomkins defines nine emotional states: *anger*, *interest*, *contempt*, *disgust*, *distress*, *fear*, *joy*, *shame*, and *surprise* based on the density of neural firing (Tomkins, 1984). Izard proposes ten emotional states which are: *anger*, *contempt*, *disgust*, *distress*, *fear*, *guilt*, *interest*, *joy*, *shame*, and *surprise* by

Table 2.3: Emotions categorised by researchers based on (Ortony and Turner, 1990).

Reference	**Fundamental Emotion**	**Basis for inclusion**
(Arnold, 1960)	Anger, aversion, courage, dejection,desire, despair, hate, fear, hope, love, sadness	Emotion is related to action tendencies
(Ekman et al., 1972)	Anger, disgust, fear, joy, sadness, surprise	Emotion is related to universal facial expressions
(Frijda, 1986)	Desire, happiness, interest, surprise, wonder, sorrow	Emotion is related to forms of action readiness
(Gray, 1982)	Anxiety, joy, rage and terror	Emotions are hardwired
(Izard, 1971)	Anger, contempt, disgust, distress, fear, guilt, interest, joy, shame,surprise	Ditto
(James, 1884)	Fear, grief, love, rage	Emotions involve movement
(McDougall, 1926)	Anger, disgust, elation, fear, subjection, tender-emotion, wonder	Emotion is related to instincts
(Mowrer, 1960)	Pain, pleasure	These are unlearned emotional states
(Oatley and Johnson-Laird, 1987)	Anger, disgust, anxiety, happiness, sadness	Do not demand Propositional content.
(Panksepp, 1982)	Expectancy, fear, rage, panic	Emotions are hardwired
(Plutchik, 1980)	Acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise	Relation to adaptive biological processes
(Tomkins, 1984)	Anger, interest, contempt, disgust, distress, fear, sadness, surprise	Relation to the density of neural firing.
(Watson, 1930)	Fear, love, rage	Emotions are hardwired
(Weiner and Graham, 1984)	Happiness, sadness	Emotions are attribution independent

regarding emotions as hardwired (Izard, 1971). Finally, Arnold determines eleven emotional states: *anger*, *aversion*, *courage*, *dejection*, *desire*, *despair*,

fear, *hate*, *hope*, *love*, and *sadness* by considering action tendencies (Arnold, 1960).

As shown in tables 2.2 and 2.3, Ekman made a change to one of his basic emotions. In 1982 he used *joy*, but in the year 1992, he changed *joy* to *happiness*. According to a recent survey in 2016, there has not been any major differences in the emotions as shown in Tables 2.2 and 2.3 (Bruna et al., 2016)

Parrott classifies the emotions of human beings via an emotion hierarchy. He uses six types of emotions at the base level and these emotions are *joy*, *sadness*, *surprise*, *love*, *anger* and *fear* as shown in figure 2.6. The other words are sorted to the second and third levels (Parrott, 2001).

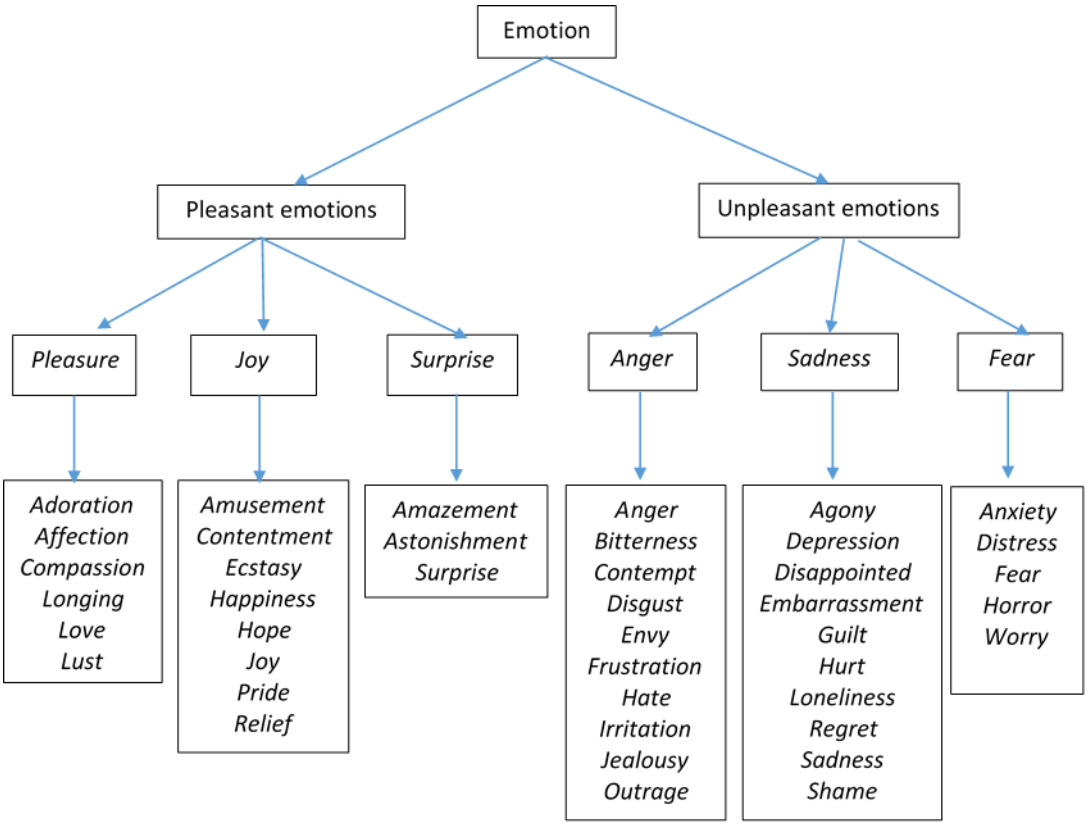


Figure 2.6: Different hierarchical structure of emotions based on (Parrott, 2001).

The left side of the figure describes "pleasant emotions" which are a mix

of high positive emotions and low negative affect such as *Adoration*, *Amusement*, *Amazement*, *Ecstasy* and *Love*. The right side describes "unpleasant emotions" and is a mixture of high negative and low positive affect such as *Anger*, *Depression*, *Anxiety*, *Disappointment*, *Hate*. The "strong engagement" and "disengagement" axis is the opposite of "pleasant emotions" and "unpleasant emotions". The map of the circumplex theory of affect is completed as shown in figure 2.7.

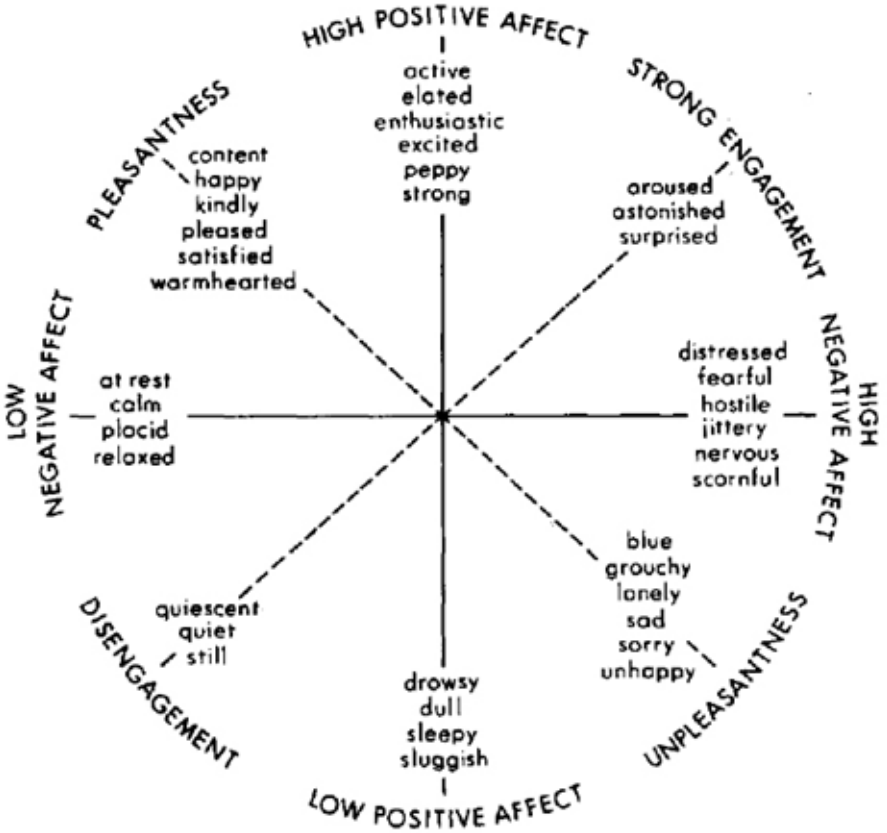


Figure 2.7: Circumplex theory of affect (Watson and Tellegen, 1985).

2.2.2 Sentiment Analysis

The process of identifying the polarity of the sentence, paragraph and document is useful and valuable. This process is useful specially in marketing, when users are trying to buy a product online and they do not know if this product is good or satisfies their needs. Or the reviews of each product could give an idea about this product and change the customer opinion toward this product.

Liu (2012) declared that sentiment analysis is an investigation of human attitudes, evaluation, and estimation towards entities for example products, company, car and so on. The terms "sentiment analysis", "opinion mining", "review mining", "subjectivity analysis", "opinion extraction", "sentiment mining", "affect analysis", "emotion analysis" are all under the umbrella of sentiment analysis (Liu, 2012).

The terms sentiment, opinion, attitude, orientation, and emotion are all related to human subjectivity. Subjectivity reflects the deep emotion of a human, his/her ideas and attitudes. It is not easy to find out the subjectivity in the text, and various methods have been devised such as sentiment analysis, opinion mining, emotion detection, and emotion recognition.

Most sentiment analysis research has focused on computing the sentiment in a given text, and finding whether it is positive, negative, or neutral. Karlgren and Cutting was one of the earliest research in this field (Karlgren and Cutting, 1994). They used a statistical computation to figure out the text genre and its categorization from the topic. But after that the idea of determining sentiment of a whole document not by topic was suggested by Pang et al. (2002). They used movie reviews as data from internet movie database IMDb archive and applied machine learning (Naïve Bayes, Maximum entropy, Support vector machine) to find out whether the reviews were positive or negative. In the same year Turney reported that an unsupervised technique can be used to predict the sentiment orientation of text as being either recommended and not recommended. He used reviews from different sites such as movie reviews, banks, cars, and travel destination (Turney, 2002). Other research by Pang and Lee (2004) concerning sentiment analysis on

movie reviews to find the polarity of reviews used Naïve Bayes, SVM and the minimum-cut framework. The latter is a statistical framework based on two different types of information: *Individual scores* and *Association scores*. The result achieved by Naïve Bayes with the help of the minimum cut framework is slightly better than using Naïve Bayes only. The accuracy achieved with the addition of the minimum cut framework was 86.4%, while 85.2% accuracy was achieved by Naïve Bayes without using the minimum cut framework. SVM achieved 86.15% accuracy when using the minimum cut framework and 85.45% without using the minimum cut framework. They took data from the website `www.rottentomatoes.com` and the Internet Movie Database IMDb.

Whitelaw et al. (2005) present a state of the art method for discerning sentiments based on appraisal theory and support vector machines. This was applied to movie reviews and achieved an accuracy of 90.2%. Kennedy and Inkpen (2006) reported a new method that focused on detecting the polarity of the reviews. They used a general inquirer to determine positive and negative sentiment and using a term-counting method with unigrams and bigrams with support vector machines to increase the accuracy.

Sentiment analysis has been applied on different areas not only on the movie reviews, but also applied for banks, company products, destination tourism, business and so on (Dave et al., 2003; Yu and Hatzivassiloglou, 2003; Popescu et al., 2005).

Most sentiment analysis research has focused on computing the sentiment in a given text, besides finding whether the overall sentiment is positive, negative, and neutral. The Merriam-Webster dictionary defines sentiment as an attitude, thought, or judgement prompted by feeling. Yadollahi et al. suggested that sentiment is an idea or opinion coloured by an emotion (Yadollahi et al., 2017). So, the process of analysing sentiment in text involves the analysis of both the opinion and the emotion behind the sentiment.

Emotion and opinion have a strong correlation. For example, sometimes emotions motivate a person to come to some conclusion about a product and build an opinion about it. Similarly, the opinion of a person affects emotions in others. Yadollahi et al. gives the following example “My family thinks it’s a good decision to continue my education overseas, though they feel sad to

miss me". This example represents a positive opinion and a negative emotion towards the same topic (Yadollahi et al., 2017).

Figure 2.8 provides a classification of different types of sentiment analysis. This figure classifies sentiment analysis under two main types: opinion mining and emotion mining. The opinion mining category is concerned with an expression of opinion such as positive, negative, and neutral sentiment but emotion mining is concerned with the expression of emotions such as *Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise* and so on (Yadollahi et al., 2017).

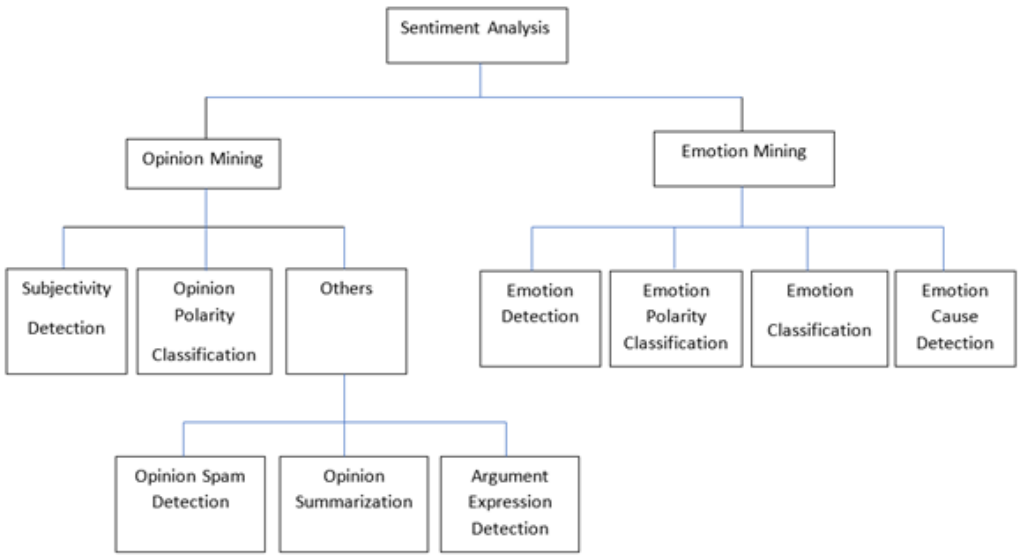


Figure 2.8: Taxonomy of sentiment analysis tasks (Yadollahi et al., 2017).

The various types of sentiment analysis can be defined as follows:

- *Subjectivity detection*: Liu defines subjectivity detection as the process of detecting whether a text is subjective or objective. A subjective text is a text that contains a personal opinion such as "I like the colour blue", while an objective text contains factual information such as "The sky is blue" (Liu, 2012).

- *Opinion polarity classification*: This is the process of calculating whether

the text contains "positive", "negative" or sometimes "neutral" opinion (Yadollahi et al., 2017).

- *Opinion Spam Detection*: Jindal and Liu work on detecting the fake opinion (which is divided into three types: *untruthful opinions* that involve the reviews that deliberately mislead readers, *reviews on brands only* concerning the reviews that comment on brands and manufactures or the seller, and *non-reviews* involving advertisements and other irrelevant reviews that contain no opinion) that could be written intentionally by malicious users in order to raise the positivity or negativity of opinion for or against a product or service in order to make this product or service popular or unpopular (Jindal and Liu, 2008).

- *Opinion Summarization*: This is the process of summarizing a large set of opinions toward a point or topic, incorporating alternate points of view, perspectives, and polarities. This is vital specially when somebody needs to settle on a choice, in light of the fact that a solitary sentiment can not be dependable. The work of Hu and Liu was the earliest work involving sentiment summarisation on item reviews (Hu and Liu, 2004).

- *Argument Expression Detection*: This is the process of distinguishing argumentative structures and the connection between various arguments inside a document, for example, one being against the other. Lin et al. provided one of the earliest works in this area (Lin et al., 2006).

- *Emotion Detection*: This is the process of identifying whether the content of the text contains emotions or not. This is like subjectivity detection for opinion mining and was first investigated by Gupta et al. (Gupta et al., 2013).

- *Emotion Polarity Classification*: This is the process of deciding the polarity of current emotion in text, expecting it has some. This is similar to opinion polarity classification. Instances of this work can be

found in Handcock et al. and Alm et al. (Hancock et al., 2007; Alm et al., 2005).

- *Emotion Classification*: This is the process of fine-grained classification of emotion in text which involves labelling the content into a pre-defined set of defined emotions. A large portion of the literature that we expand on later in this thesis falls into this category (Yadollahi et al., 2017).

- *Emotion Cause Detection*: This is the process of factors related to the causes of emotions. Early work in this category was by Lee et al. (2010) with more recent work by Gao et al. (2015).

In this thesis, our work focuses specifically on emotion classification (also called emotion recognition).

2.2.3 Levels of Analysis

Generally, sentiment analysis has been explored mainly at three levels:

Document-level: This is the task that focuses on sentiment classification of documents. It is the process of classifying a document according to a positive or negative opinion, sentiment or emotion. It is also known as "document level classification" due to it considering the entire document as one entity (Pang et al., 2002; Turney, 2002). In this level of analysis, the document should express only one opinion, sentiment or emotion, rather than more than one entity (opinion or sentiment).

The work in this study involves data collected from web blogs and sentences from fairy tales. The training texts are combined together and dealt with as one entity by combining the texts for each Ekman's emotion (*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise*) as one entity. But instead of opinion mining using the polarity of text (positive, negative, and neutral), we will classify each document as *Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise*. Our experimental results on document sentiment classification will be discussed in more detail in the next chapter.

Sentence level: The task at this level works at the sentence level, and decides if each sentence is positive, negative, neutral or contains a sentiment (Liu, 2012). According to Neviarouskaya et al. the surrounding context could influence the sentence and this is a challenge (Neviarouskaya et al., 2010).

Entity and Aspect level: The aspect level implements finer-grained analysis. It is also known as feature level (feature-based) opinion mining and summarization (Hu and Liu, 2004). It includes determining opinion polarity about a specific part of a product or service (Wang et al., 2016; Lin and He, 2009; Jo and Oh, 2011; Mukherjee and Liu, 2012).

2.2.4 Measures for Evaluating the Classification

In the field of the text classification and machine learning, a confusion matrix also known as an error matrix is used to describe the performance of the classification algorithm (Stehman, 1997). Each row in the confusion matrix represents the instances of the predicted class while each column represents the actual instances of the class (Powers, 2011).

Table 2.4: Example of confusion matrix for two classes.

		Actual class	
		Class 1	Class 2
Predicted class	Class 1	5 (True Positives)	2 (False Positives)
	Class 2	3 (False Negatives)	17 (True Negatives)

We use TP as an abbreviation for the number of the True Positive, FP is the number of the False Positives, FN is the number of the False Negatives and finally the TN is the number of the True Negatives. To calculate the *Accuracy* for each classification (Olson and Delen, 2008), we used macro-averaging of the class accuracies:

$$Accuracy = \frac{1}{N} \sum_{C \in Classes} \frac{TP_C + TN_C}{TP_C + TN_C + FP_C + FN_C}$$

where $Classes$ is the set of classes, and N is the number of classes ($N = 5$ for Alm's dataset, and $N = 6$ for Aman's and LiveJournal datasets). To calculate the *Precision* and *Recall*, we also used macro-averaging:

$$Precision = \frac{1}{N} \sum_{C \in Classes} \frac{TP_C}{TP_C + FP_C}.$$

$$Recall = \frac{1}{N} \sum_{C \in Classes} \frac{TP_C}{TP_C + FN_C}.$$

Finally, in order to further evaluate the performance of each classifier, the F-measure was calculated as follows:

$$F{-}measure = 2 \times (\frac{Precision \times Recall}{Precision + Recall}).$$

2.2.5 Computational Methods for Emotion Recognition

Three main methods have been used for emotion recognition in text. These methods are keyword based, learning based, and hybrid-based methods. These methods use n-grams, part of speech tags, phrase patterns, and synonyms as features for emotion recognition.

1- Keyword-based method: This method depends on the presence of emotion keywords and it may include pre-processing of a text by using a parser and emotion dictionary . Strapparave et. al used linguistic resources for lexical representation of affective knowledge called WordNet–Affect (Strapparava et al., 2004). A subset of synset that represented affective concepts related to affective words was included in WordNet-Affect. This method simply identified the emotion words in the text that are found in WordNet-Affect. Other publications used this method in online chat systems (Ma et al., 2005; Hancock et al., 2007; Zhang et al., 2005; Li et al., 2007).

2- Learning-based method: This method uses a trained classifier to classify text into emotion classes using keywords as features. This method can easily adapt to domain changes as it quickly learns new keywords from datasets by using a large training dataset as an input to the machine learning algorithm to produce a classification model. Research by Strapparave and Mihalcea (2008) used this type of method by developing a system based on a variation of Latent Semantic Analysis to classify emotions in text, with the text having no affective words. Unfortunately, this method achieved low accuracy due to a lack of semantic analysis and it is not context sensitive (Strapparava and Mihalcea, 2008).

3- Hybrid-based method: This method consists of the combination of the previous two methods in addition to some extra information added to the classification from various sciences such as information from psychology (Wu et al., 2006). The advantage of this method is that it outperforms the previous two methods in terms of accuracy by combining learning of the classifier from the training dataset and adding knowledge-rich linguistic information from dictionaries (Binali et al., 2010). Wu et al. (2006) suggested a novel approach working on sentence level emotion mining based on detecting predefined Semantic Labels (SLs) and Attributes of the sentence (ATTs), afterwards classifying based on psychological patterns of human emotions called Emotion Generation Rule (EGR). However, this method was limited to only to these emotions (*Happiness*, *Unhappiness* and *Neutral*) as this method suffered from ambiguity if the EGR generated more than current emotions (Wu et al., 2006).

Yang et al. suggested a hybrid model for emotion recognition that encompasses lexicon keyword spotting, CRF based (conditional random field) emotion cue identification, and machine learning based on emotion classification by using the SVM, Naïve Bayes, and Max entropy classifiers. The results are generated by different techniques and integrated by different vote-based merging strategies. The method performed well with the manually annotated gold stand suicide notes with precision of 58%, recall 64%, and F-measure of 61% (Yang et al., 2012).

Ghazi et al. proposed a multiple hierarchal method to classify Ekman's

emotion. In the first classification level, they find whether a sentence contains an emotion or not, then the classification examines the polarity of the sentence (positive or negative), and finally a fine-grained classification for the actual emotion is applied to the sentence. For every classification stage, they used various features, and they obtained (+7%) better accuracy than flat emotion classification on fined-grained classification. The disadvantage of this method is that it is not context sensitive (Ghazi et al., 2010).

Neviarouskaya suggested EmoHeart, a lexical rule-based system to classify emotion in text that depicts the emotion expressions in a 3D virtual world called "Second Life" (Neviarouskaya et al., 2010). Their system looked first for emoticons and emotional abbreviations, This stage called *Symbolic Cue Analysis*, then if the system could not find any, the system goes to the *Syntactical Structure analysis* stage which is devoted to the analysis of syntactical structure of the sentence. The sentence is processed into different levels (word, phrase, and sentence) called *Word Level Analysis* to create an emotional vector that represents the sentence. At the word level, each word is mapped to the emotional vector, that contains values for the emotions included in the system (anger, disgust, fear, guilt, interest, joy, sadness, shame, surprise), a dataset of emotional vector is being built from these used. At phrase level, and sentence level they merge the emotional vectors of the words that composed the phrase or the sentence by either summation or maximizing among these vectors see Figure 2.9 . Finally, the emotion of the sentence is calculated by the maximum intensity of the vector.

2.2.6 Machine Learning Methods

This section provides a brief summary of the machine learning classifiers used in this thesis. Also, a comparison among the following classifiers to classify Ekman's emotion in text is provided in subsequent chapters: ZeroR, Naïve Bayes, J48, Support Vector Machine (SVM) and Prediction by Partial Matching (PPM).

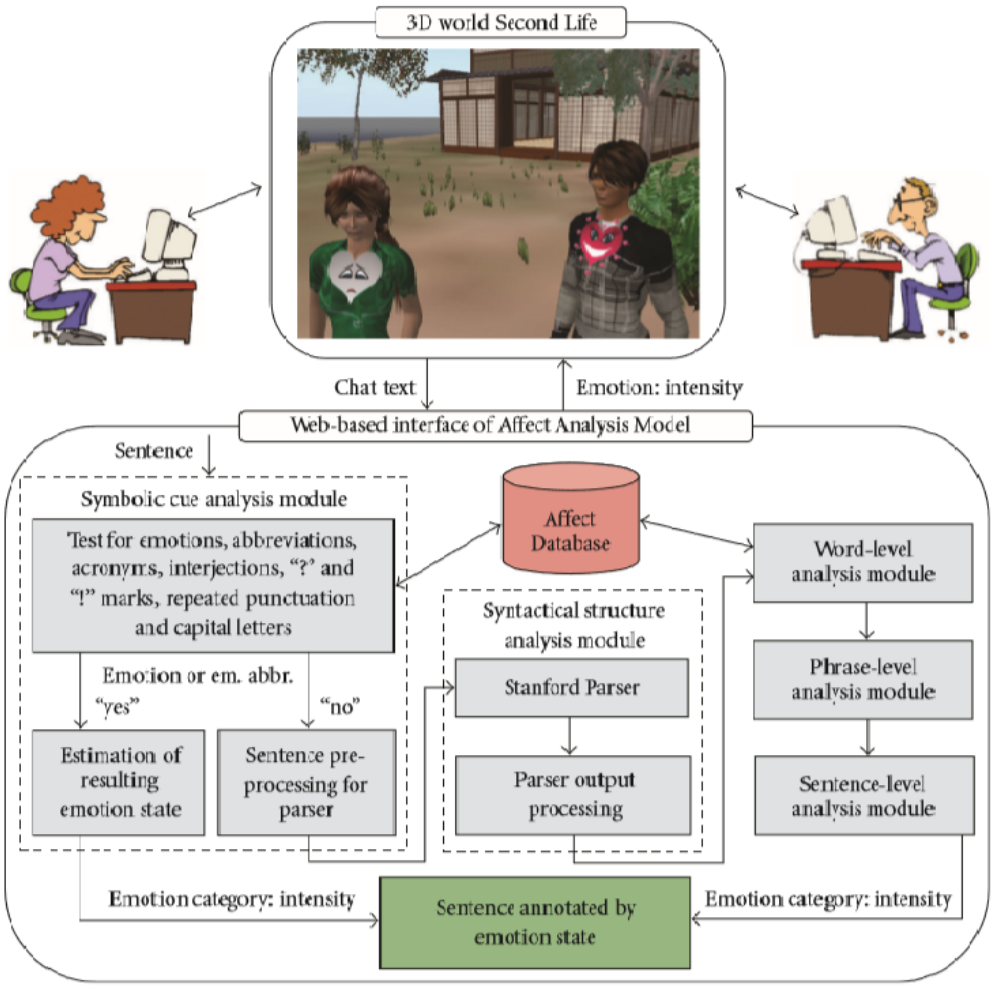


Figure 2.9: Architecture of the EmoHeart system (Neviarouskaya et al., 2010).

2.2.6.1 ZeroR

ZeroR classifier represents the simplest classification, it predicts the most common class in the training data for all test data. It is used to compare its result with other classifiers with the presence of the dominant class (Fernández-Delgado et al., 2014).

2.2.6.2 Naïve Bayes

Naïve Bayes is a classifier based on the calculation of well-known prior and conditional probabilities (Yong-feng and Yan-ping, 2004). This approach calculates the probability that document D belongs to class C. "Naïve" refers to the assumption that all the attributes of the classification are independent of each other when predicting the context of the class. According to this assumption, each attribute can be learned separately, and this simplifies learning when we have a large number of attributes (Kim et al., 2006). There are two main types of models for this classifier, multivariate Bernoulli, and multinomial models (Lewis, 1998; Agrawal and An, 2012; McCallum et al., 1998).

Troussas et al. (2013) uses the Naïve Bayes classifier in sentiment analysis of Facebook status. They compared it with the Rocchio classifier which is a classic algorithm for processing relevance feedback that stemmed from the SMART Information Retrieval System. It is based on the assumption that nearly all users have a common conception of which documents can be labelled as relevant or irrelevant (Manning et al., 2010). Troussas et al. (2013) also compared the Naïve results with Perception classifiers which is a linear binary classifier using a supervised learning approach (Freund and Schapire, 1999). They found that Naïve Bayes achieved better results than the Rocchio and perception classifiers. Pratama and Sarno uses the Naïve Bayes, K-Nearest Neighbours and support vector machine classifiers for personality classification for Twitter text. They found out that the Naïve Bayes classifier results are competitive with other classifiers (Pratama and Sarno, 2015).

2.2.6.3 J48 Decision Tree

Decision tree classification is one of the possible ways towards multistage decision making (Haralick, 1976). The main idea of the decision tree is to break down a complex decision-making process into a group of simpler decisions, therefore providing a simple solution to interpret (Safavian and Landgrebe, 1991). Decision trees support a very efficient way of obtaining a decision and

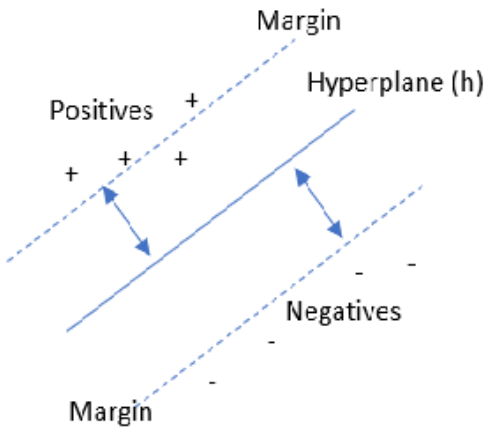
this makes the decision tree an important classifier in different applications such as pattern recognition where the efficiency is important (Nagel, 1987)

Decision tree classification builds a decision tree model from training data that contains labels for classes. The decision tree algorithm attempts to discover the way that attribute vectors behave for a number of training samples classes in order to predict new samples (Korting, 2006). The decision tree classification algorithm helps in understanding the critical distribution in data (Nadali et al., 2011). Gokulakrishnan et al. used the J48 classifier with other classifiers for emotion detection on Twitter text (Wang et al., 2012). Munezero et al. used the J48 for detecting antisocial behaviour in text (Munezero et al., 2014).

2.2.6.4 SVM

The Support Vector Machine (SVM) proposed by Cortes and Vapnik (1995) uses a supervised learning approach. The SVM assumes a hyperplane that classifies the training data, the hyperplane classifies the positive and negative training data. Classification is based on the distance of the margin from the decision hyperplane to positive and negative training examples (Binali et al., 2010) as shown in Figure 2.10.

Figure 2.10: Distribution of even margins from svm hyperplane (h) (Binali et al., 2010).



Lin combined Naïve Bayes with SVM to reduce the number of features in the feature vector (Lin, 2002). SVM is also very effective at multi-label text classification (Qin and Wang, 2009). Mishne presents preliminary experiments on mood classification for blog posts using SVM (Mishne et al., 2005). Aman and Szpakowicz used Naïve Bayes and SVM classifiers in binary classification to classify the emotional vs. non emotional blogs, and found that the SVM outperforms Naïve Bayes in terms of accuracy (Aman and Szpakowicz, 2007).

2.2.7 Entropy

Entropy is defined as a measure of the predictability of the content of a message. When the upcoming character is highly predictable then the entropy is low; however when the upcoming character is difficult to predict then the entropy is high. Shannon presented the idea of entropy for a language. Entropy has became an essential idea for information theory (Shannon, 1948).

Let S be a finite sequence with possible values: $s_1, s_2, s_3, ...s_n$, $P = p(s_1)$, $p(s_2), ..., ps_n$ where the probabilities are independent and sum to 1.

The entropy H of S is :

$$H(S) = -\sum_{i=1}^{n} P(s_i) \log_2 P(s_i). \tag{2.1}$$

If the base of the logarithm is 2, then the measurement of the unit is given by bits. For example, assume we have a text containing four characters *d,e,f,g* with the following probabilities $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{2}$ respectively. Now we can determine the average number of bits (codelength) for this text, or the minimum number of bits needed to represent each symbol is as follows:

$$H = -P(d) \log_2 P(d) - P(e) \log_2 P(e) - P(f) \log_2 P(f) - P(g) \log_2 P(g)$$

$$H = -\tfrac{1}{2} \log_2 \tfrac{1}{2} - \tfrac{1}{2} \log_2 \tfrac{1}{2} - \tfrac{1}{3} \log_2 \tfrac{1}{3} - \tfrac{1}{2} \log_2 \tfrac{1}{2}$$

$H = 1.28$ bits.

As Teahan states entropy is "a measure of how much uncertainty is involved in the selection of a symbol — the greater the entropy, the greater the uncertainty. It can also be considered a measure of the information content of the message — more probable messages convey less information than less probable ones" (Teahan, 1998). Further discussion of entropy are given in (Charniak, 1993), (Brown et al., 1992), (Jelinek, 1990), (Shannon, 1948) and they all have contributed to the following discussion.

Assume $S = s_1, s_2, s_3, ..., s_n$ is a stream of symbols of a language L, then equation 2.1 can be reformulated as:

$$H(S) = -\sum P(s_1, s_2, s_3, ..., s_n) \log_2 P(s_1, s_2, s_3, ..., s_n). \tag{2.2}$$

The sum in equation 2.2 and in subsequent formulas are assumed to be made over all possible sequences. The value of $H(S)$ reflects the difficulty of the language modelling task. So the higher value of $H(S)$, the more difficult to predict the next symbol. The difficulty of the language modelling task can be measured by the following equation:

$$\frac{1}{n}H(S) = -\frac{1}{n}\sum P(s_1, s_2, s_3, ..., s_n) \log_2 P(s_1, s_2, s_3, ..., s_n). \tag{2.3}$$

Equation 2.1 can be expanded to be more general for a language with probability distribution L:

$$H(L) = \lim_{n\to\infty} -\frac{1}{n}\sum P(s_1, s_2, s_3, ..., s_n) \log_2 P(s_1, s_2, s_3, ..., s_n). \tag{2.4}$$

This is named the *entropy of the language* and can be considered to be the limit of the entropy when the length of the message gets very large. For independent sources,

$$p(x_1, x_2, ..., x_n) = p(x_1)p(x_2)...p(x_n) \tag{2.5}$$

and equation 2.4 reduces to equation 2.1. According to Brown et al. (1992) if the process producing the language is *ergodic* (which is that sufficiently long sequences of symbols are typical of it, and can be used to deduce its statical structure), then this formula reduces to the Shannon McMillian Breiman theorem.

$$H(L) = \lim_{n\to\infty} -\frac{1}{n} \log_2 p(x_1, x_2, ..., x_n). \tag{2.6}$$

Normally, the probability distribution of the language L is not known. However, the upper bound to $H(L)$ can be estimated by applying a model M for the language L as an approximation:

$$H(L, M) = -\sum P_M(s_1, s_2, s_3, ..., s_n) \log_2 P_M(s_1, s_2, s_3, ..., s_n). \tag{2.7}$$

To estimate the probabilities, the model $P_M(s_1, s_2, s_3, ..., s_n)$ is applied for the probabilities $p(x_1, x_2, ..., x_n)$. Always, the entropy H(L) is less or equal to the *cross-entropy* $H(L, M)$ as it is dependent on the best possible source for the language model (Teahan, 2000):

$$H(L) <= H(L, M). \tag{2.8}$$

Teahan estimated the entropy of the English language by using the PPM text compression scheme and the entropy was 1.48 bpc (approximately 1.5 bits per character as a measure of the compression-rate) (Teahan, 1998). Here, the bits per character (or "bpc") can be calculated as follows:

$$bpc = \frac{\text{Encoded file size (Bytes)} \times 8}{\text{Original file size (Bytes)}}. \tag{2.9}$$

Brown concluded that to estimate an upper bound to the entropy, text-compression can be applied (Brown et al., 1992). We can define the number of bits $b_m(s_1, s_2, s_3, ...s_n)$ for a stream of symbols in the text $(s_1, s_2, s_3, ...s_n)$

by applying a model M as follows:

$$H(L, M) = \lim_{s \to \infty} \frac{1}{n} b_M(s_1, s_2, s_3, ...s_n). \tag{2.10}$$

$H(L, M)$ represents the number of bits needed to encode a long series of text taken from the language L.

According to Brown, the cross entropy represents a measure of how successful the model of the language is at predicting the test text. The closer the cross entropy $H(L, M)$ is to $H(L)$, the less imprecise the model of the language is. So the lower the cross entropy, the better performance of the model (Brown et al., 1992).

In this thesis, the Prediction by Partial Matching (PPM) text compression scheme has been used to perform entropy-based text categorisation. PPM will be discussed in the next section.

2.2.8 Text categorisation using Prediction by Partial Matching (PPM)

Prediction of Partial Matching (PPM) is an adaptive lossless text compression method first published in 1984 (Cleary and Witten, 1984) that processes characters in the text in a sequential manner. A variable order Markov-based model is updated dynamically as the text is processed with both the encoder and decoder maintaining the same model at each stage of the encoding and decoding processes. A finite context is used to predict the upcoming character in the text. Standard PPM will use a fixed maximum context to try to make its initial prediction. (This defines the "order" of the model). Text compression experiments with English and some other natural language texts have shown that a fixed maximum context length of 5 (i.e. an order 5 model) usually works best. The method essentially estimates probabilities for the upcoming character.

The model uses an "escape" mechanism that smoothes the probability estimates by backing-off to a shorter context when novel characters are encountered (i.e. those with zero probabilities). This backing-off process may

need to be undertaken multiple times until a context is found where the character can be predicted. For characters that have not been seen anywhere previously in the text, a default order -1 context is used where every character is predicted with equal probability. Various escape methods (such as methods A, B, C and D) have been devised over the years to define how the escape probability is estimated. (These are described in the literature as variants PPMA, PPMB and so on).

The PPMC variant was developed by Moffat (Moffat, 1990) and has become the benchmark version. The probability estimates for this method is based on using the number of characters that have occurred before, called the number of types:

$$e(X) = \frac{t(X)}{f(X) + t(X)} \quad \text{and} \quad p(x_i|X) = \frac{c(x_i|X)}{n(X) + t(X)} \tag{2.11}$$

where $e(X)$ represents the probability of the escape symbol for context X, $p(x_i|X)$ denotes the probability for character x_i given context X, $c(x_i|X)$ is the number of times the context X was followed by the character x_i, $f(X)$ is the total number of times that the context X has occurred and $t(X)$ denotes the total number of types of the predictions in that context.

The PPMD variant was first developed by Howard in 1993 (Howard, 1993). In most cases, experiments show that PPMD performs better than the other variants. This variant is similar to the PPMC variant with the exception that each count is incremented by a 1/2:

$$e(X) = \frac{t(X)}{2f(X)} \quad \text{and} \quad p(x_i|X) = \frac{2c(x_i|X) - 1}{2f(X)}. \tag{2.12}$$

The performance of the PPM models has improved using 'full exclusion' mechanism. The 'full exclusion' mechanism involves excluding counts from lower order calculations during escaping for symbols already predicted by a higher order (since they would have been encoded already so can be excluded). This mechanism has been found to improve compression by a few percent in most experiments.

The PPM model can be represented by the following formula:

$$H_M(T) = \sum_{1}^{n} -log_2\, p(x_i|x_{i-m}, \ldots, x_{i-1})$$

where H_M is the compression codelength given model M of order m for the probability distribution for the characters x_i over the text sequence $T = x_1, x_2 \ldots x_n$ of length n. Each character will be predicted based on the prior context x_{i-m}, ..., x_{i-1} of length m.

To simplify how PPMD works, Table 2.5 illustrates how the PPMD model works orders 2, 1, 0 and -1 where k = 2,1,0 and -1 have been processed for the input string 'passionless'. For clarification purpose for this example, the highest context order is $k = 2$. If the upcoming character is predicted successfully by the modelling context then the probability p will be used to encode this character, whereas c signifies the frequency count of that character. In the example, if the input string 'passionless' is followed by the character 'i', the probability of prediction of this character using the context 'ss $\rightarrow$ i' in order 2 is $\frac{1}{2}$. This probability will be used to encode the character 'i' and it only requires only one bit to encode since (-$\log_2 \frac{1}{2} = 1$).

Presume that another upcoming character 's' follows the string 'passionless'. Since the order 2 model does not predict this character, as a consequence the escape probability ($\frac{1}{2}$) of the context 'ss $\rightarrow$ i' will be encoded for model 2. In this situation, the encoder will move down from the order 2 model ($k = 2$) to the order 1 model ($k = 1$). For order 1, the context 's $\rightarrow$ s' predicts the character 's' with a probability of $\frac{3}{6}$. So the total probability needed to encode the character 's' is $\frac{1}{2} \times \frac{3}{6}$ or 2 bits. In this context, a further more precise probability prediction is obtained by observing that the character 'a' can not be encoded using this context. So, we can exclude the characters that are already predicted by higher orders. This is what is called the full exclusion technique which corrects the probability of this character in order 1 yielding the total probability of $\frac{1}{2} \times \frac{3}{5}$ or 1.7 bits required to encode this character.

In contrast, if the upcoming character is '.' which has never been seen before, the escape probability will be repeated down from order 2 through all models to order -1 (k = -1) where all characters are encoded with equal prob-

abilities of $\frac{1}{|A|}$ where $|A|$ is the alphabet's size. Assuming that the English language is encoded using 8 bit American Standard Code for Information Interchange (ASCII) so therefore the alphabet size is 256. So the total probability to encode the character '.' will be $\frac{1}{2} \times \frac{2}{6} \times \frac{8}{22} \times \frac{1}{|A|}$ or 12.1 bits. The full exclusion technique can be used again to obtain more a precise probability by excluding the characters that are predicted in higher orders, as follows: $\frac{1}{2} \times \frac{2}{5} \times \frac{8}{17} \times \frac{1}{|A|-8}$ or 11.4 bits.

Text categorisation using PPM is performed by training N different models $M_1, M_2, ...M_N$ where N is the number of classes and the training text used to train each model is representative of the class being modelled. The main idea is to guess the correct class of the text T using the following formula:

$$\hat{\theta}(T) = \arg\min_i H_{M_i}(T) \tag{2.13}$$

for each class i. Essentially, one constructs a PPM model for each class, and the text is compressed using each model with the class being chosen from the model that compresses the text best.

Prediction by Partial Mapping (PPM) has been used in various applications for text categorization and natural language processing. Khmelev and Teahan used PPM in natural language processing to accurately recognize the source of written text (Khmelev and Teahan, 2003). Teahan and Harper used PPM to recognize the most relevant author of the text (Teahan and Harper, 2001). Al-Kazaz and Teahan used PPM to perform the automatic cryptanalysis of ciphers, and word segmentation in order to make the decoded text more readable (Al-Kazaz and Teahan, 2016).

Recently, Altamimi and Teahan used PPM to classify gender (Altamimi and Teahan, 2017). Alamri and Teahan used PPM for automatic correction of Arabic dyslexic text (Alamri and Teahan, 2019). As far as we know, no research prior to this study has used PPM for emotion recognition.

2.2.9 Relative-Entropy

In this section, we will describe how we can use relative-entropy (also called information divergence) for text categorization. This is related to the new

Table 2.5: Processing the string "passionless" using PPMD models with maximum order of 2.

Order k=2			**Order k=1**			**Order k=0**			**Order k=-1**		
Prediction	c	p	Prediction	c	p	Prediction	c	p	Prediction	c	p
PPMD											
pa → s	1	$\frac{1}{2}$	p → a	1	$\frac{1}{2}$	→ p	1	$\frac{1}{22}$	→ A	1	$\frac{1}{\|A\|}$
→ Esc	1	$\frac{1}{2}$	→ Esc	1	$\frac{1}{2}$	→ a	1	$\frac{1}{22}$			
						→ s	4	$\frac{7}{22}$			
as → s	1	$\frac{1}{2}$	a → s	1	$\frac{1}{2}$	→ i	1	$\frac{1}{22}$			
→ Esc	1	$\frac{1}{2}$	→ Esc	1	$\frac{1}{2}$	→ o	1	$\frac{1}{22}$			
						→ n	1	$\frac{1}{22}$			
ss → i	1	$\frac{1}{2}$	s → s	2	$\frac{3}{6}$	→ l	1	$\frac{1}{22}$			
→ Esc	1	$\frac{1}{2}$	→ i	1	$\frac{1}{6}$	→ e	1	$\frac{1}{22}$			
			→ Esc	2	$\frac{2}{6}$	→Esc	8	$\frac{8}{22}$			
si → o	1	$\frac{1}{2}$									
→ Esc	1	$\frac{1}{2}$	i → o	1	$\frac{1}{2}$						
			→ Esc	1	$\frac{1}{2}$						
io → n	1	$\frac{1}{2}$									
→ Esc	1	$\frac{1}{2}$	o → n	1	$\frac{1}{2}$						
			→ Esc	1	$\frac{1}{2}$						
on → l	1	$\frac{1}{2}$									
→ Esc	1	$\frac{1}{2}$	n → l	1	$\frac{1}{2}$						
			→ Esc	1	$\frac{1}{2}$						
nl → e	1	$\frac{1}{2}$									
→ Esc	1	$\frac{1}{2}$	l → e	1	$\frac{1}{2}$						
			→ Esc	1	$\frac{1}{2}$						
le → s	1	$\frac{1}{2}$									
→ Esc	1	$\frac{1}{2}$	e → s	1	$\frac{1}{2}$						
			→ Esc	1	$\frac{1}{2}$						

method devised in chapter 4.

Let us assume we have documents that are given labels independently to one or more classes. Let us suppose we have a training group of the classified

documents (Teahan and Harper, 2003). For any given class C, we can create a model for that class P_C, and a model for its complement $P_{\overline{C}}$. Relative entropy gives a technique to decide whether a particular document D is associated with a particular class C. We will use the notation $P(D|C) = P_C(D)$ and $P(D|P\overline{C}) = P_{\overline{C}}(D)$ to decide D is associated with C if:

$$P(D|C)P(C) > P(D|\overline{C})P(\overline{C}) \tag{2.14}$$

where $P(C)$ is the previous probability of the document that is associated with class C. The following expression will be used to classify the document with respect to class C:

$$-log_2 \frac{P(D|C)}{P(D|\overline{C})} \tag{2.15}$$

A cut-off is picked to provide optimal classification, depending on some measure of classification goodness. Equation 2.15 has dual formulation in the compression field with cross-entropy as follows:

$$= -\log_2 \frac{P(D|C)}{P(D|\overline{C})}$$

$$\equiv -log_2 P(D|C) - log_2 P(D|\overline{C})$$

$$\equiv -log_2 P_(D|C) + log_2 P(D|\overline{C}))$$

$$\equiv H(C, P_C, D) - H(\overline{C}, P_{\overline{C}}, D). \tag{2.16}$$

Consequently, maximising the code-length differences between document model and its complement model is equivalent to creating an independent Bayes classifier for each class (Teahan and Harper, 2003). Teahan and Harper (2003) and Frank et al. (2000) provided a similar analysis (Teahan and Harper, 2003). In using text classification based compression, the optimal cut-off for code-length differences will be computed using equation 2.16 for each class and its complement and this is called the class cut-off.

Frank et al. highlighted the need for feature selection in the process of text classification, but they did not mention using feature selection in text classification based compression (Frank et al., 2000). Teahan and Harper suggested selecting features that exceeded certain threshold on code-length differences (Teahan and Harper, 2003). The code-length differences were sorted based on the potential contribution to the overall code-length differences as in equation 2.16. They used feature selection for each class individually and the resulting cut-off was refered to as feature cut-off.

The advantage of using thresholding (class and feature selection) mentioned by Frank et al. (2000) is that this will automatically adjust to different amounts of training data from the model P_M and its complement $P_{\overline{M}}$.

2.3 English Datasets for Emotion Recognition

Various datasets have been used by other researchers for their experiments on emotion recognition. A full description of three of the most common datasets that have been used for the experiments in this study is provided below.

Some samples from each of the datasets have also been provided in Table 2.6. They show the diverse nature of the texts included in the datasets and that for the LiveJournal blogs especially, the text contains many non-standard features including spelling mistakes, grammatical errors and colloquialisms potentially making the classification task more difficult. Table 2.7 shows the number of texts classified in each class for the three datasets.

LiveJournal Dataset

The LiveJournal dataset is a large dataset composed of 815,494 web blog posts (whose total size is 1.6 GB in the original XML format). LiveJournal is a free weblog service available at `http://www.LiveJournal.com/` used by millions of users. It is classified into 132 moods such as *happy*, *cheerful* and *sad* where the author of the blog has chosen to describe their mood while

Dataset	Emotion	Sample
LiveJournal	*Anger*	.... i have nothing positive to say right now. at all.
LiveJournal	*Fear*	i feel like the world is talking behind my back.... i feel like the person who is the but of the joke but doesnt know it....
LiveJournal	*Happiness*	Well kids, I had an awesome birthday thanks to you. = D Just wanted to so thank you for coming and thanks for the gifts and junk. =) I have many pictures and I will post them later. 'hearts;
Aman	*Disgust*	I think the most important thing I can say about this city, is that all of the rumors: That the city's dirty, the people are rude, the language barrier is insurmountable, the metro is incomprehensible, you things will be stolen, that they hate Americans.
Aman	*Happiness*	the trip was fantastic, plimoth was old and there were real live pilgrims everywhere, salem was cute and adorable but very touristy, ithaca was gorgeous and adorable and joel was there so the whole thing made me very happy and clearly that was the highlight of the trip, niagara was fantastically beautiful but boring once you've seen the falls, buffalo was sketchy but had great wings and now im home, and i think it's the most wonderful place ive ever been and i dont ever want to leave!
Alm	*Anger-Disgust*	Then he was very angry, and went without his supper to bed; but when he laid his head on the pillow, the pin ran into his cheek: at this he became quite furious, and, jumping up, would have run out of the house; but when he came to the door, the millstone fell down on his head, and killed him on the spot.
Alm	*Sadness*	And now the sister wept over her poor bewitched brother, and the little roe wept also, and sat sorrowfully near to her.
Alm	*Surprise*	But–seated upon the stump, she was startled to find an elegantly dressed gentleman reading a newspaper.

Table 2.6: Samples from the datasets used for emotion recognition experiments.

Table 2.7: Number of texts classified in each of the emotion classes for the three datasets.

Dataset	*Anger-Disgust*	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Alm	218			166	445	264	114
Aman		179	172	115	536	173	115
LiveJournal		562	140	277	3601	1164	520

writing his/her post (Mishne et al., 2005). One issue with this dataset is that the consistency of the moods found in the dataset is highly variable because they are individually assigned. On the other hand, it helps us to access the writer's emotion directly without using an external annotator.

Aman's Dataset

This dataset is composed of emotion-rich sentences taken from web blogs. The author of this dataset searched for web blogs that contained various seed words representing the emotion. This dataset consists of 1290 emotional sentences of six types of emotions (179 *Anger* sentences, 172 *Disgust* sentences, 115 *Fear* sentences, 536 *Happiness* sentences, 173 *Sadness* sentences, and 115 *Surprise* sentences) based on Ekman's definition of emotions and also comprises 2800 non-emotional sentences in total. A collection of seed words was used for each emotion; for instance, the *Happiness* emotion encompasses the following seed words ('awesome', 'happy', 'amused', 'fantastic', 'excited', 'pleased', 'cheerful', 'love', 'great', 'amazing'). It also uses a neutral label for sentences that do not contain emotions. Four annotators were used to manually label sentences in the resulting dataset (Aman and Szpakowicz, 2007; Aman and Szpakowicz, 2008).

Alm's Dataset

This dataset consists of annotated sentences taken from fairy tales. The sentences in this dataset are labelled with five types of emotions (*Anger-Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise*) based on Ekman's definition of emotions. Since *Anger* and *Disgust* emotions are similar, Alm decided to merge them into one class (Alm et al., 2005).

2.4 Summary and Discussion

We have divided this chapter into two parts. The first part reviewed the background subjects to this thesis. We reviewed affective computing, and text categorization as it related to this thesis. We also reviewed Arabic language fundamentals, including its characters, Arabic calligraphic styles, and the types of Arabic language. We discussed the Arabic encoding methods with particular attention to those used for Arabic language. We found there are three encoding methods used (ISO 8859-6, windows 1256, and UTF-8). We also found that the UTF-8 encoding method is the most popular encoding method used and requires more than one byte to represent Arabic characters.

In part two, we have reviewed the different theories of emotions, and discussed sentiment analysis. We discussed the levels of analysis for sentiment analysis. We also discussed the computational methods for emotion recognition. We gave a brief description of the relevant machine learning methods previously used for emotion recognition such as Naïve Bayes, J48, and the SVM classifier. We also explained entropy and how it can be used for classification. An explanation of the PPM compression scheme and how it works has been provided. The idea of relative entropy in text classification has been discussed. An overview of the available English emotions datasets has also been provided.

We found that no prior research has used the PPM approach for emotion recognition. This is the main reason for choosing the Prediction by Partial Matching (PPM) for emotion recognition. Later chapters examine how well PPM performs at this task.

Chapter 3

Emotion Recognition in text using PPM

3.1 Introduction

In this chapter, we investigate the effectiveness of using PPM for automatic recognition of emotion in English text. This chapter is based on two conference papers (Almahdawi and Teahan, 2017; Almahdawi and Teahan, 2018).

Granularity is one large system or entity broken down into small pieces or parts. For instance, a meter is broken into centimetres. Coarse-grained systems consist of a small number and large ingredients than fine-grained systems. A coarse-grained description of a system considers large subcomponents, on the other hand, a fine-grained system considers smaller components of the large ones are composed (Fink et al., 2011).

The purpose of this chapter is to determine the effectiveness of one possible method for recognising emotions in text using text compression. We are interested in both coarse grained emotion recognition (such as whether the text is emotional or not) and fine-grained emotion recognition such as distinguishing between *Happiness* and *Sadness* emotions or distinguishing texts according to Ekman's six basic emotions. Text compression can be used to classify texts by emotion using a two stage supervised learning process: the first stage builds models by training on texts that are representative of each

type of emotion being classified; and the second stage uses the training models to compress each testing text, and then assign the class using the label associated with the model that compresses the testing text best.

This chapter is organised as follows. The next section discusses related work. The PPM-character based text compression method for classifying texts is then discussed followed by a description of how the datasets used in our experiments were processed along with the experimental results. The chapter completes with the conclusion and future work in the final section.

3.2 PPM-based Text Categorisation

Prediction of Partial Matching (PPM) is an adaptive lossless text compression method first published in 1984 (Cleary and Witten, 1984) that processes characters in the text in a sequential manner. A variable order Markov-based model is updated dynamically as the text is processed with both the encoder an decoder maintaining the same model at each stage of the encoding and decoding processes. A finite context is used to predict the upcoming character in the text. Standard PPM will use a fixed maximum context to try to make its initial prediction (This defines the "order" of the model). Text compression experiments with English and other natural language texts have shown that a fixed maximum context length of 5 (i.e. an order 5 model) usually works best. The method essentially estimates probabilities for the upcoming character. The method was discussed in more detail in Chapter 2, section 2.2.8.

3.3 Experimental Results

Our experimental results are presented in this section.Three datasets have been used for the experiments: the LiveJournal dataset (Mishne et al., 2005), Alm's dataset (Alm et al., 2005) and Aman's dataset (Aman and Szpakowicz, 2007).

The experimentation plan is as follows:

Table 3.1: Experiments types and purpose.

No.	Chapter	Type of experiment	Purpose of experiment
1	3	Ekman's emotion classification at document level	To see how PPM performs at document level
2	3	Ekman's emotion classification at document level using equal sizes of training data files	To see how PPM performs at document level using equal sizes of training data files
3	3	emotional vs non-emotional fine-grained classification on the blog and sentence level	To see how PPM performs at fine-grained level compared to traditional classifiers
4	3	Happiness vs Sadness fine-grained classification on the blog and sentence level	To see how PPM performs at fine-grained level compared to traditional classifiers
5	3	Ekman's emotion fine-grained classification on the blog and sentence level using punctuated vs non-punctuated text	To see how PPM performs at fine-grained level using the punctuated vs non-punctuated text
6	3	Ekman's emotion fine-grained classification on the blog and sentence level	To see how PPM performs at fine-grained level compared to other research results
7	3	Ekman's emotion fine-grained classification on the blog and sentence level in different PPM order models	To see how PPM performs at fine-grained level with different order models
8	4	Ekman's emotion fine-grained classification on the blog and sentence level using the ID classifier	To see how ID classifier performs at fine-grained level using ID1, ID2 and ID3 and find which on is with the best result

9	4	Ekman's emotion fine-grained classification on the blog and sentence level using the ID classifier	To see how ID classifier performs at fine-grained level compared to other classifiers results
10	5	Ekman's emotion fine-grained classification on the blog and sentence level using the traditional word-based classifiers	To see how the traditional word-based classifiers perform on the new Arabic dataset (IAEC)
11	5	Ekman's emotion fine-grained classification on the blog and sentence level using the traditional character-based classifiers	To see how the traditional character-based classifiers perform on the new Arabic dataset (IAEC)
12	5	Ekman's emotion fine-grained classification on the blog and sentence level using PPM	To see how PPM performs on the new Arabic dataset (IAEC)
13	5	Ekman's emotion fine-grained classification on the blog and sentence level using different PPM order models	To see how PPM order models perform on the new Arabic dataset (IAEC)
14	5	Ekman's emotion fine-grained classification on the blog and sentence level using the ID classifier	To see how ID classifier performs at fine-grained level on the new Arabic dataset (IAEC)
15	5	Ekman's emotion fine-grained classification on the blog and sentence level using the ID classifier	To see how ID classifier performs at fine-grained level compared to other classifiers results

3.3.1 Experimental Methodology

The TAWA Toolkit developed by Teahan was used to obtain the PPM compression codelength estimates. This toolkit allows PPM models to be constructed from training text. Although standard dynamic PPM models are possible using the toolkit, we chose to use a static variation where once the models have been trained, they are not updated subsequently as the testing text is being processed since previous text categorisation experiments (Teahan and Harper, 2003) have shown these models to be just as effective as dynamic models.

For our experiments, we used order 5 PPM character-based models with escape method D as this usually leads to the best compression for English text(Cleary and Witten, 1984). Static models were created while training, and then used to classify the separate testing data. Ten-fold cross validation was then applied to evaluate the classification of the text according to Ekman's six basic emotions for all three datasets using the static models. Prior to each of the experiments described below, we applied the following pre-processing steps to the various datasets. Aman and Alm's datasets are directly labelled by Ekman's basic emotions. An extra label is used for blogs that do not contain emotions in Aman's dataset. For Alm's dataset, each sentence is classified with its equivalent emotion label. However, for the LiveJournal dataset, the following pre-processing is required. Firstly, the XML tags were removed; secondly, all punctuation and URLs were removed; and thirdly, blogs labelled in the same class were extracted from the LiveJournal data and concatenated together to form six separate text files for Ekman's classes. Aman and Szpakowicz (Aman and Szpakowicz, 2007) describe how to collect blogs from the web by using seed words. Each one of these seed words have its equivalent mood in the LiveJournal weblogs so we use these to map the 132 moods to the six Ekman emotions using synonyms (for example, *awesome* and *fantastic* are some synonyms for the *Happiness* emotion).

In order to improve the effectiveness of the PPM classifier, we have found that balancing the training data for the different PPM models works best. Where there is an imbalance in training data for a particular class or classes,

then the performance of the PPM classifier can be improved by essentially truncating the amount of training data used for all classes to the smallest class training size with any unused training data discarded. The rationale behind using balanced sizes for class training data is as follows. If the training size for one class were to predominate, for example, then that class model will often become a better predictor of the general language (e.g. English) compared to the individual class models at predicting the language specific to each class. (Just as idiolects are associated with the speech peculiarities of an individual person, the language of each class will also have its own peculiarities, but this can be dominated if one has a better model of the standard variety of the language). In order to overcome this issue, a simple expedient is to truncate training sizes for each class so that they are the same even if this means truncating training text in most classes down to the smallest class training size. As far as we are aware, we are the first to report this result that balancing of class training size often leads to noticeably improved classification performance for emotion recognition for the PPM classifier.

The text files for each dataset were first split into ten partitions in order to perform ten-fold cross-validation where different folds were used for training and testing ensuring that text was split on a blog boundary rather than in the middle of the blog. For the LiveJournal dataset, in order to obtain roughly equal sized training text for each class, the files were reduced in size to just over 2.4MB each using the text from the beginning of each file. For Aman's dataset, text related to each of the six Ekman emotions was extracted directly according to the blog annotations. This resulted in just 10KB being available when using balanced text sizes as Aman's dataset is much smaller than the LiveJournal dataset. Similarly, for Alm's dataset, the text for each emotion could be extracted directly, with 12KB of text available for each emotion.

We have performed both document level and fine-grained experiments as discussed below.

3.4 Experiments with document level classification

For document level experiments, we used PPMD5 on the three datasets for classifying Ekman's emotions. Table 3.2 summarises the best results obtained for the three datasets in terms of accuracy, precision, recall and F-measure.

Table 3.2: PPMD5 classification results for Ekman's emotions for the three datasets.

Dataset	**Accuracy**	**Precision**	**Recall**	**F-measure**
LiveJournal	96.1 %	0.90	0.88	0.89
Aman	95.6 %	0.89	0.87	0.88
Alm	88.0 %	0.71	0.70	0.70

Further experiments on the LiveJournal, Aman, and Alm datasets to classify Ekman's emotions have been performed to determine what effect the training text size has and how using balanced or unbalanced training sizes for classes affects the classification accuracy, precision, recall, and F-measure (see Table 3.3). First, we used equal-sized texts for training from the LiveJournal dataset for each of the six classes that ranged in size from 100KB to 2.4MB. We compared this to several cases where the text sizes used were not balanced—one where 3MB was used for all classes except for *Disgust* which used 2.4MB which was the maximum available for that class; and one where the full text that was available in the dataset was used for each class (so the training text sizes across the six classes was very unbalanced ranging from 2.4MB for *Disgust* up to 78MB for *Happiness*, with the latter class having a size greater than the combined sizes for the other five classes).

The results listed in Table 3.3 for the LiveJournal dataset are striking and show two noticeable trends. Firstly, the results consistently improve with training size, rising from an accuracy of 78.9% when as little as 100KB is used up to 96.1% when 2.4MB is used instead. And secondly, if unbalanced sizes are used between classes, this leads to a noticeable drop in performance, dropping down to an accuracy of 76.6% when training size is maximised across the classes but is very unbalanced as a result. Even when only one

class is unbalanced (as when 3MB is used for all classes except *Disgust*), the negative effect of unbalancing the class sizes outweighs the positive effect of using a larger size for the other classes.

Table 3.3: PPMD5 classification results for Ekman's emotions for the three datasets.

Datas.	**Training size**	**Acc.**	**Prec.**	**Rec.**	**F-m.**
LiveJ.	100KB *for each class*	78.9 %	0.35	0.37	0.36
	500KB *for each class*	87.8 %	0.73	0.63	0.68
	1MB *for each class*	91.7 %	0.83	0.75	0.79
	2MB *for each class*	95.6 %	0.87	0.87	0.87
	2.4MB *for each class*	96.1 %	0.90	0.88	0.89
	3MB *except class for Disgust* (2.4MB)	88.3 %	0.74	0.65	0.69
	Anger (22.1MB), *Disgust* (2.4MB), *Fear* (11.9MB), *Happi.* (78MB), *Sadness* (24.7MB), *Surpr.* (8.8MB)	76.6 %	0.53	0.30	0.38
Aman	10KB *for each class except Surprise* (9.8KB)	95.6 %	0.89	0.87	0.88
	Anger (17.5KB), *Disgust* (13.5KB), *Fear* (12.4KB), *Happi.* (38.8KB), *Sad.* (13.5KB), *Surpr.* (9.8KB)	78.9 %	0.70	0.37	0.48
Alm	12KB *for each class*	88.0 %	0.71	0.70	0.70
	Anger-Disgust (23.7KB), *Fear* (20KB), *Happiness* (59.9KB), *Sadness* (36.4), *Surprise* (12.3KB)	75.6 %	0.65	0.42	0.51

We also explored the effect of training size for both Aman's dataset and Alm's dataset. These datasets have smaller size compared to the LiveJournal dataset so it is not possible to explore what effect increasing the size of the text has in any depth. However, we did compare the two cases when the class training text sizes were mostly balanced (approx. 10KB for Aman's dataset and 12KB for Alm's dataset), and when the sizes were unbalanced using the full text that was available in each class. In this case, the *Happiness* class again had the largest size compared with the other five classes, and was just over 4 times the size for the smallest class *Surprise*. The results in Table 3.3 clearly shows that using balanced class sizes compared to using unbalanced sizes for these datasets has a significant impact on classification performance, dropping from 95.6% to 78.9% accuracy for Aman's dataset and from 88.0% to 75.6% accuracy for Alm's dataset.

3.4.1 Experiments with fine-grained classification

For fine-grained experiments, we used also PPMD5 in Ekman's emotions classification on the three datasets. Subsection 3.4.1.1 shows the binary classification results between emotional versus non-emotional text. Subsection 3.4.1.2 shows the binary classification results between happiness versus sadness text. Subsection 3.4.2 shows the classification results of Ekman's emotions in the three datasets using PPM. Subsection 3.4.3 shows the results of Ekman's emotions classification in Alm dataset using different orders of PPM. Subsection 3.4.4 shows the Ekman's emotions classification on Aman dataset using different orders of PPM. Moreover, subsection 3.4.5 shows Ekman's emotions classification on LiveJournal dataset using different orders of PPM.

3.4.1.1 Experiments with *emotional* versus *non-emotional* sentences for Aman's dataset

In this experiment, Aman's dataset has been used for the training and testing data. The purpose of this experiment was to evaluate the effectiveness of the PPM classifier at distinguishing between emotional and non-emotional content. Based on the available training data in Aman's dataset, we used text to train both the emotional and non-emotional PPM models. Two text files were extracted from Aman's dataset. One of these files contained 1290 blogs deemed to be emotional, while the other file contained 2800 blogs deemed to be non-emotional. These text files were used to evaluate the PPM classifier using a ten-fold cross-validation process with 9/10 of the text being used to train two static order 5 PPMD models which were then used to predict the appropriate class on the remaining test data. In this classification, the classification level is on blog level.

The ZeroR, J48, Naïve Bayes and SMO classifiers implemented in Weka were applied to the same dataset. For all classifiers, the StringToWordVector filter has been used with the NGramTokenizer to select NGrams as features to compare with PPM since the latter implicitly works with n-grams. The results that were obtained are shown in Table 3.4.

Table 3.4: Classification results on emotional versus non-emotional sentences for different classifiers on Aman's dataset.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
PPMD5	63.3%	**0.59**	**0.60**	**0.59**
ZeroR	68.5%	0.50	0.50	0.50
Naïve Bayes	**69.4%**	0.50	0.51	0.50
J48	68.5%	0.50	0.50	0.50
SMO	**69.4%**	0.50	0.51	0.51

As shown in Table 3.4, the highest accuracy was achieved by the Naïve Bayes and SMO classifiers with 69.4%, while the PPM classifier achieved the lowest accuracy among these classifiers with 63.3%. However, in terms of precision, recall and F-measure, the PPM classifier clearly outperforms the feature-based classifiers that we experimented with such as ZeroR, Naïve Bayes, J48, and SMO. The precision of the feature-based classifiers were as follows: ZeroR 0.50, Naïve Bayes 0.50, J48 0.50, and SMO 0.50. The PPMD5 classifier achieved a significantly higher precision of 0.59. In addition, PPM achieved the best results for recall and F-measure compared to other classifiers with 0.60 recall and 0.59 F-measure. While ZeroR and J48 achieved 0.50 for both recall and F-measure, Naïve Bayes and SMO achieved recall results of 0.51 and F-measure of 0.50 for Naïve Bayes and 0.51 for SMO. All measures were computed as macro-averages of precision, recall, and F-measure for the emotional and non-emotional classes for the ten folds used during the ten-fold cross-validation evaluation process.

3.4.1.2 Binary classification: *Happiness* versus *Sadness*

The second set of experiments investigated the binary classification problem of distinguishing between texts classed by the two Ekman emotions *Happiness* and *Sadness*. These emotions are the only pair of Ekman's emotions that are antonyms of each other and therefore they should be easier to distinguish.

The PPM method was first applied to the data extracted from the LiveJournal data for just the two classes. The two text files used for our experiments contained 3601 blogs in the *Happiness* class, and 1164 blogs in the

Sadness class. The results of the experiment are presented in Table 3.5 where we compared PPM results with other classifiers such as Naive bayes, ZeroR, J48, and SMO on the same tested blogs.

In comparison, Mihalcea and Liu also used LiveJournal blogs to classify only the *Happiness* and *Sadness* blogs by using a Naïve Bayes classifier and their method achieved 79.13% accuracy (Mihalcea and Liu, 2006) but Mihalcea used five-fold cross validation.

Our experimental results in distinguishing *Happiness* versus *Sadness* are shown in Table 3.5. They show that PPM outperforms the other classifiers in terms of accuracy, recall and F-measure, although Naïve Bayes and SMO have better precision.

Table 3.5: Classification results on Happiness versus Sadness sentences for different classifiers on LiveJournal's dataset.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
PPMD5	**80.0%**	0.78	**0.62**	**0.69**
ZeroR	71.1%	0.36	0.50	0.42
Naïve Bayes	71.3%	**0.86**	0.50	0.63
J48	71.1%	0.36	0.50	0.42
SMO	72.2%	**0.86**	0.51	0.64

Our experimental results in distinguishing Happiness versus Sadness are shown in Table 3.5. They show that PPM outperforms the other classifiers in terms of accuracy, recall and F-measure, although Naïve Bayes and SMO have better precision.

The next experiments investigated the *Happiness* versus *Sadness* binary classification for Aman's and Alm's datasets. For Aman's dataset, text was extracted for the two different classes. The number of happiness sentences to be tested was 536, and the number of sadness sentences was 173 sentence. On the other hand, Alm's dataset consisted of 445 sentences for the happiness emotion, while there were 264 sentences for the sadness emotion. These texts were used to train PPM models and classify test data separately. PPM was used to produce models of each text. Ten-fold cross validation was used to evaluate the classification of test data according to the *Happiness* versus

Sadness emotions.

Table 3.6 summarises the PPM classifier results for the three datasets. 80.0% accuracy was obtained for the Livejournal dataset, 79.1% accuracy for Aman's dataset, whereas 76.9% accuracy was obtained for Alm's dataset. We have not applied Weka classifiers on the Alm and Aman's datasets to classify between *Happiness* verses *Sadness* since no-one has yet used this type of classification to compare with.

Table 3.6: PPM classification results for *Happiness* versus *Sadness* emotions produced by the PPM classifier for the LiveJournal, Aman, and Alm datasets.

Dataset	**Accuracy (%)**	**Precision**	**Recall**	**F-measure**
LiveJournal	80.0 %	0.78	0.62	0.69
Aman	79.1 %	0.72	0.66	0.69
Alm	76.9 %	0.76	0.73	0.75

3.4.2 Experiments with Ekman's emotion classes

In these experiments, the PPM method was applied to the three datasets in order to classify Ekman's basic emotions on blog level for Aman and LiveJournal's dataset, and sentence level for Alm's dataset.

Previous compression experiments with English text (Teahan, 1998) have shown that a PPM order 5 model with escape method D is effective and therefore this was used for the classification experiments. Ten-fold cross validation was applied to evaluate the classification of the text according to Ekman's six basic emotions for all three datasets and for each fold a static model was built from the training data of that fold for each class.

Table 3.7 summarises the results that were obtained for the three datasets in terms of accuracy, precision, recall and F-measure. For example, the average accuracy for the classification of Ekman's six basic emotions for the LiveJournal dataset was 87.4%, for Aman's dataset was 85.0%, whereas for Alm's dataset was 69.2%.

A comparison was also made between using the PPM classifier for Ekman's classes on the LiveJournal dataset using text with and without basic

Table 3.7: PPM classification results for Ekman's emotions for the LiveJournal, Aman, and Alm datasets.

Dataset	**Accuracy**	**Precision**	**Recall**	**F-measure**
LiveJournal	87.4 %	0.69	0.26	0.38
Aman	85.0 %	0.50	0.42	0.46
Alm	69.2 %	0.26	0.24	0.25

pre-processing steps applied to it for blogs. The purpose of this experiment was to determine if the presence of the punctuation and digits were important for the classification or not. The pre-processing involved removing all punctuation and digits from the dataset. Table 3.8 shows the comparison of the classification results for the two texts of the LiveJournal dataset. The results for the PPM classifier changed noticeably when using the raw LiveJournal text compared to when the text was pre-processed first by removing the punctuation and digits. Obviously, PPM achieved better accuracy, precision, recall and F-measure with the LiveJournal's text without punctuations as the punctuations in the LiveJournal's text has negatively affected the classification result of PPM.

Table 3.8: PPM classification results for Ekman's emotions for the two versions of the LiveJournal dataset with and without punctuation and digits.

Dataset	**Accuracy**	**Precision**	**Recall**	**F-measure**
LiveJournal text with punctuation and digits (i.e. raw text)	87.4 %	0.69	0.26	0.38
LiveJournal text without punctuation and digits (pre-processed text)	90.9 %	0.77	0.37	0.50

The confusion matrix that resulted from applying PPM to classify emotions for the text that contains punctuations is presented in Table 3.9. The training class is shown in the leftmost column with the testing class shown in the topmost row. The number of correct classifications made are shown in bold font. Table 3.10 presents the confusion matrix that resulted when

applying PPM to classify emotions for the pre-processed text (without punctuations).

Table 3.9: Confusion matrix for the PPM classification of the six basic emotions for the LiveJournal blogs with punctuation.

Training	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Anger	**46**	0	1	484	30	1
Disgust	0	**18**	0	112	9	1
Fear	0	0	**17**	234	19	7
Happiness	13	2	4	**3464**	105	13
Sadness	2	0	1	842	**317**	2
Surprise	2	1	1	427	50	**39**

Table 3.10: Confusion matrix for the PPM classification of the six basic emotions for the LiveJournal blogs without punctuation.

Training	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Anger	**53**	0	2	493	12	2
Disgust	0	**18**	1	115	5	1
Fear	2	0	**17**	249	9	0
Happiness	9	3	5	**3524**	55	5
Sadness	1	1	1	384	**769**	8
Surprise	2	1	0	209	141	**167**

Table 3.10 shows the confusion of the *Happiness* class with both *Sadness* and *Surprise* classes has been reduced leading to an increase in the number of the correctly classified items of the *Sadness* and *Surprise* classes.

Further PPM classification experiments were conducted using two versions of the text (with and without punctuation and digits) for both Aman's and Alm's datasets. The results are shown in Tables 3.11 and 3.14. The results show that unlike the LiveJournal dataset, the removal of the punctuation and digits from Aman's dataset reduces the classification slightly, with accuracy decreasing from 84.9% to 84.4%, precision from 0.50 to 0.47, recall increasing from 0.42 to 0.43 and F-measure decreasing from 0.46 to 0.45. For Alm's dataset, there is also some decrease in these measures with accuracy

decreasing from 69.2% to 69.0%, precision decreasing from 0.26 to 0.25, recall decreasing from 0.24 to 0.23 and F-measure decreasing from 0.25 to 0.24.

Table 3.11: PPM classification results on Ekman's emotions for the two versions of Aman's Dataset with and without punctuation and digits.

Dataset	**Accuracy**	**Precision**	**Recall**	**F-measure**
Aman's dataset with punctuation and digits (i.e. raw text)	84.9 %	0.50	0.42	0.46
Aman's text without punctuation and digits (pre-processed text)	84.4 %	0.47	0.43	0.45

Table 3.12 shows the confusion matrix that resulted from applying PPM to classify Ekman's emotions of Aman dataset. The text used in this experiment contains punctuations and numbers, while table 3.13 shows the confusion matrix that resulted from applying PPM to classfy Ekman's emotions in Aman dataset that contains text without punctuation and numbers.

Table 3.12: Confusion matrix for the PPM classification of the six basic emotions for the Aman blogs with punctuations.

Training	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Anger	**66**	23	11	56	17	6
Disgust	26	**55**	11	67	11	2
Fear	11	13	**45**	35	7	4
Happiness	11	17	9	**460**	29	10
Sadness	12	16	16	79	**48**	2
Surprise	10	3	3	59	5	**35**

Table 3.17 compares the PPM results with previously published results for the classification of Ekman's emotions on individual blogs for the three datasets. Chaffer and Inkpen (Chaffar and Inkpen, 2011) used various traditional classifiers (ZeroR, Naïve Bayes, J48, SMO) implemented using Weka on both Aman's and Alm's datasets. Ghazi et al. (Ghazi et al., 2010) used both a flat SMO classifier and a two-level SMO classifier on the same two datasets. However, it is important to note that direct comparison between

Table 3.13: Confusion matrix for the PPM classification of the six basic emotions for the Aman blogs without punctuations.

Training	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Anger	**70**	18	13	56	14	8
Disgust	39	**54**	14	53	9	3
Fear	21	6	**50**	27	6	5
Happiness	33	19	11	**420**	34	19
Sadness	16	15	19	62	**53**	8
Surprise	15	6	4	43	7	**40**

Table 3.14: PPM classification results on Ekman's emotions for the two versions of Alm's dataset with and without punctuation and digits.

Dataset	**Accuracy**	**Precision**	**Recall**	**F-measure**
Alm's dataset with punctuation and digits (i.e. raw text)	69.2 %	0.26	0.24	0.25
Alm's dataset without punctuation and digits (pre-processed text)	69.0 %	0.25	0.23	0.24

Table 3.15: Confusion matrix for the PPM classification of the six basic emotions for the Alm blogs with punctuations.

Training	*A/D*	*Fear*	*Happiness*	*Sadness*	*Surprise*
A/D	**53**	44	47	53	21
Fear	14	**41**	63	44	4
Happiness	63	49	**71**	253	9
Sadness	21	29	119	**93**	2
Surprise	21	13	34	26	**20**

these studies is difficult due to the different processing and data selection methods used in each case.

Table 3.16: Confusion matrix for the PPM classification of the six basic emotions for the Alm blogs without punctuations.

Training	*A/D*	*Fear*	*Happiness*	*Sadness*	*Surprise*
A/D	**42**	50	54	57	15
Fear	16	**44**	48	54	4
Happiness	67	60	**65**	241	12
Sadness	22	32	104	**104**	2
Surprise	24	16	23	34	**17**

Table 3.17: Comparing accuracy results for Ekman's emotions for the three datasets.

	Accuracy		
Method (and reference)	LiveJournal	Aman	Alm
PPM [*This thesis*]	87.4 %	**84.9 %**	**69.2** %
ZeroR (Chaffar and Inkpen, 2011)		68.5 %	36.9 %
Naïve Bayes (Chaffar and Inkpen, 2011)		73.0 %	54.9 %
J48 (Chaffar and Inkpen, 2011)		71.4 %	47.5 %
SMO (Chaffar and Inkpen, 2011)		81.2 %	61.9 %
Flat SMO (Ghazi et al., 2010)		61.7 %	57.4 %
Two-level SMO (Ghazi et al., 2010)		65.5 %	56.6 %

3.4.3 Experiments with Ekman's emotion classes with different orders of PPM for the Alm dataset

In these experiments, the PPM method was applied to the Alm dataset in different orders to classify Ekman's emotions due to find the best order of PPM that could help improving the result of classification. These experiments started from PPMD2 to PPMD10. Table 3.18 summarises the results.

Table 3.18 shows the highest accuracy was achieved by PPMD4 with value of 69.4%, and the lowest accuracy was achieved by PPMD2 with a value of 68.2%. The variation in accuracy between the highest and lowest values is only 1.2%. In comparison with PPMD5, the variation in F-measure is only 0.4 compared to PPMD2 which is a slight decrease in performance but not significant.

Based on the values of the highest and lowest F-measure, the difference

Table 3.18: PPM classification results for Alm's dataset using different PPMD model orders.

Order	Accuracy (%)	Precision	Recall	F-measure
PPMD2	68.2 %	0.20	0.21	0.21
PPMD3	69.0 %	0.24	0.23	0.24
PPMD4	**69.4 %**	0.26	**0.24**	**0.25**
PPMD5	69.2 %	0.26	**0.24**	**0.25**
PPMD6	69.3 %	**0.27**	**0.24**	**0.25**
PPMD7	69.0 %	0.25	0.23	0.24
PPMD8	69.1 %	0.25	0.23	0.24
PPMD9	69.3 %	0.25	0.23	0.24
PPMD10	69.1 %	0.25	0.23	0.24

is not significant. In addition, the PPMD4 achieved the best performance for classifying Ekman's emotion for Alm's dataset.

3.4.4 Experiments with Ekman's emotion classes with different orders of PPM for the Aman's dataset

In these experiments, the PPM method was applied to Aman's dataset using different orders to classify Ekman's emotions due to find the best order of PPM that could help improve the result of classification. Table 3.19 summaries the results that were obtained.

Table 3.19 shows that PPMD3 outperforms other orders pf PPM. PPMD3 achieved the highest values of accuracy, precision, recall and F-measure. This is due to the type of text that included in Aman dataset.

3.4.5 Experiments with Ekman's emotion classes with different orders of PPM for the LiveJournal's dataset

In these experiments, the PPM method was applied to the LiveJournal dataset in different orders and find the best order of PPM to classify Ekman's emotions. Table 3.20 summaries the results were obtained by PPMD

Table 3.19: PPM classification results for Aman dataset using different PPMD orders.

Order	Accuracy (%)	Precision	Recall	F-measure
PPMD2	85.1 %	0.48	**0.45**	**0.47**
PPMD3	**85.2 %**	**0.50**	**0.45**	**0.47**
PPMD4	84.7 %	0.49	0.43	0.46
PPMD5	85.0 %	**0.50**	0.42	0.46
PPMD6	84.7 %	0.49	0.43	0.46
PPMD7	84.7 %	0.48	0.43	0.45
PPMD8	84.7 %	0.48	0.43	0.45
PPMD9	84.7 %	0.48	0.43	0.45
PPMD10	84.6 %	0.48	0.43	0.45

on different orders.

Table 3.20: PPM classification results for the LiveJournal's dataset using different PPMD orders.

Order	Accuracy (%)	Precision	Recall	F-measure
PPMD2	83.3 %	0.32	**0.32**	0.32
PPMD3	86.6 %	0.45	**0.32**	0.37
PPMD4	**87.7 %**	0.59	0.29	**0.39**
PPMD5	87.4 %	0.69	0.26	0.38
PPMD6	87.1 %	0.72	0.25	0.37
PPMD7	87.0 %	0.72	0.25	0.37
PPMD8	86.9 %	0.73	0.24	0.36
PPMD9	87.0 %	0.73	0.24	0.37
PPMD10	87.1 %	**0.74**	0.25	0.37

Table 3.20 shows the highest accuracy achieved by PPMD4 with a value of 87.7%, and the lowest accuracy achieved by PPMD2 with a value of 83.3%. The variation between the highest and lowest values is significant being 5.1%.

3.5 Conclusion

This chapter has described how the Prediction by Partial Matching (PPM) text compression scheme can be applied to the problem of emotion recog-

nition in text. Experimental results show that the PPM is very effective when compared with other traditional data mining methods at recognising emotions in text. The proposed method processes all characters in the text without explicit feature extraction while the other research methods have relied on processing words as features.

One important issue with our proposed method should be considered. It is often the case that the amount of text available for training purposes in different classes is mis-matched document classification. In performing the experiments reported in this chapter, we have found that our method works best if the sizes of text in all classes are the same, even if this means truncating text in most classes down to the smallest class size. This is because if the size for one class is much larger than for other classes, for example, then that class model will often become a better predictor of the general language (e.g. English) compared to the individual class models at predicting the language specific to each class.

For fine-grain classification, experiments with the PPM-based classifier were performed on three datasets: the LiveJournal dataset, Alm's dataset and Aman's dataset. Binary classification to recognise either the *Happiness* or *Sadness* emotions was applied on the three datasets. The experiments on the LiveJournal dataset achieved 76.0% and on Aman's dataset achieved 79.01% accuracy, whereas our method achieved 76.9% on Alm's dataset.

Another binary classification experiment on emotional versus non-emotional sentences was applied to Aman's dataset using the PPM, Naïve Bayes, and SMO classifiers. Although the accuracy result for PPM (63.3%) was less than for two other classifiers (Naïve Bayes 69.4%, and SMO 69.4%), the PPM method achieved the best results in terms of precision, recall and F-measure for all classifiers that were compared.

Experiments at recognising Ekman's basic emotions using the PPM-based classifier were also performed on the three datasets. Our experiment on the LiveJournal's dataset achieved 87.4% accuracy, on Aman's dataset 85.0% accuracy and on Alm's dataset 69.2% accuracy. This is a significant improvement over previously published results that relied on traditional word-based data mining methods on the same datasets.

We also found variations on accuracy, precision, recall, and F-measure when these texts were pre-processed to remove punctuation characters and digits. For the LiveJournal's datasets, all measures were increased when we removed punctuation and digits from text prior to classification, although for Aman's and Alm's datasets, there was very little variation in performance.

As well, experiments on different order models were performed on the three datasets. For Alm's dataset, accuracy of 69.4% was highest for order 4 models (PPMD4), and F-measure was highest (0.25), for order 4, 5 and 6 models. For Aman's dataset, accuracy of 85.2% was highest for order 3 models, and F-measure was highest (0.47) for order 2 and 3 models. For the LiveJournal dataset, accuracy of 78.7% was the highest for order 4 models, and F-measure was highest (0.39) also for order 4 models.

In future, it would be interesting to explore this method on different languages such as Arabic (see Chapter 5) rather than English by performing additional experiments.

Chapter 4

A New Text Classifier using Information Divergence

This chapter provides a description of the components of a new text classification method. The last chapter explored the effectiveness of a compression-based classifier for emotion recognition. This chapter is based on the paper (Al-Mahdawi and Teahan, 2019). This chapter explores whether a new divergence-based classifier based on relative entropy instead that processes words can also be effective. The chapter is organised as follows: in section 4.1 describes the mechanism of work for the information divergence classifier. Section 4.2 contains subsections that illustrate the results and issues with the new classifier, these subsections are as follow: subsection 4.2.1 describes the datasets used in testing the information divergence classifier, subsection 4.2.2 describes the implementation details to evaluate the new classifier, subsection 4.2.3 lists some example outputs produced by the ID classifier, subsection 4.2.4 discusses the confusion matrices produced by ID classifier, subsection 4.2.5 discusses the issue of unclassified text produced by the ID classifier, subsection 4.2.6 displays the overall results of the experiments by ID classifier, subsection 4.2.7 compares the results of ID classifier with other published results using the same datasets, section 4.3 summarises the ID classifier.

The new method first creates some training data from two overall texts

for each class. For example, for emotion recognition, the first contains a concatenation of all the texts that are classified in each of Ekman's emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise*); and the complement containing a concatenation of all the texts that are *not* classified in each of the classes. (The term 'complement' here as commonly used in text categorisation publication refers to all the text that is included in all the other texts except for the current emotion, and does not relate to the mathematical terminology). The text complement for a particular emotion is effectively all the text except the text for the current emotion. For example, the complement of the *Happiness* emotion, $\overline{Happiness}$, is the text created from the concatenation of the *Anger*, *Disgust*, *Fear*, *Sadness*, and *Surprise* training texts together.

Figure 4.1 uses a Venn digram to illustrate some sample bigram phrases taken from both the *Happiness* and $\overline{Happiness}$ training texts (represented in the figure as E and $\overline{E}$ respectively) from the LiveJournal dataset. Referring to Figure 4.1, the E region contains samples of text found in the training documents that have been classified with the *Happiness* emotion such as "Very cute" and "Very nice". In contrast, the $\overline{E}$ region contains samples of text found in the training documents that have *not* been classified with the *Happiness* emotion such as "Zodiac sign" and "YOU THINK". Between the two regions, there are phrases that are common to both texts: "were amazed" and "loves that" are shown as two examples.

The next section will describe the new Information Divergence type classifier.

4.1 Information Divergence Classifier

The Information Divergence method uses a metric called *divergence* to measure the difference in probabilities between common n-grams found in the emotion training text and that found in its complement (We classify blogs for Aman and LiveJournal's datasets, whereas we classify sentences for Alm's dataset). Formally, let T_E be defined as the concatenation of training text documents that have been classified with a particular emotion E and $T_{\overline{E}}$ be

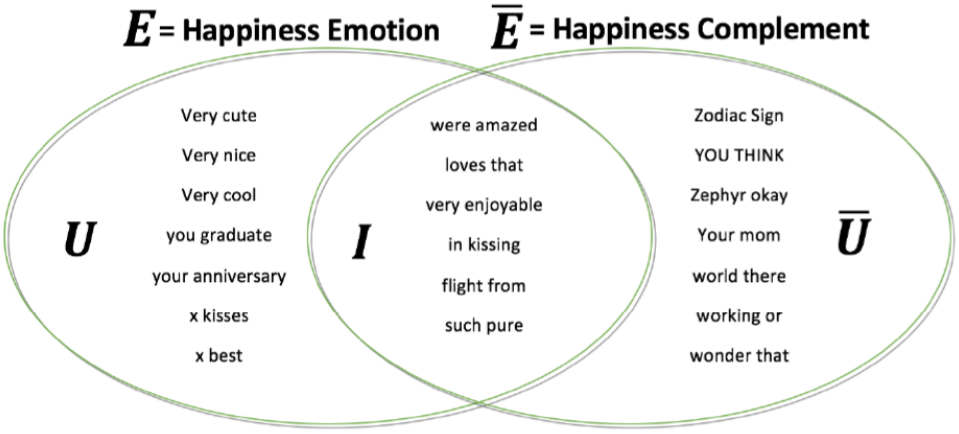


Figure 4.1: Sample bigrams taken from the *Happiness* training text from the LiveJournal dataset and its complement.

defined as the concatenation of training text documents that have *not* been classified with emotion E. Let S_E be the set of n-grams to be found in T_E (i.e. unigrams, bigrams *or* trigrams depending on our n-gram unit of analysis). Also let $S_{\overline{E}}$ be the set of n-grams to be found in $T_{\overline{E}}$. Essentially, T_E and $T_{\overline{E}}$ are the list of n-gram tokens from the respective concatenated training texts, and S_E and $S_{\overline{E}}$ are the set of n-gram types. Referring to the Venn digram in Figure 4.1, when we consider which of the n-grams are contained (or not) in these two sets, we have three regions in the figure (U, I or $\overline{U}$) whose sets are as follows:

$$S_U := S_E \cup S_{\overline{E}} - S_{\overline{E}}$$

$$S_I := S_E \cap S_{\overline{E}}$$

$$S_{\overline{U}} := S_E \cup S_{\overline{E}} - S_E.$$

These three sets are: U, the set of unique n-grams found in T_E but not in $T_{\overline{E}}$; I, the set of n-grams found in both T_E and $T_{\overline{E}}$; and $\overline{U}$, the set of unique n-grams found in $T_{\overline{E}}$ but not in T_E. The set I is used for calculating the *divergence* described below. The other two sets represent the sets of unique n-grams which are ignored for the classifier described in this chapter.

The classifier in this chapter only focuses on the area of intersection text between emotion and its complement texts in order to examine the effect of the information divergence between these. Although it would seem that using data from U and $\overline{U}$ would be helpful, preliminary experiments have shown this not to be the case, and this is left for future work.

We can compute the divergence between the two texts T_E and $T_{\overline{E}}$ as follows. Let $H(T, g)$ be the *codelength*, the cost in bits of encoding an n-gram g according to some language model. Using a naïve estimate for the probability P:

$$P(T, g) = c(T, g)/N(T)$$

where $c(T, g)$ is the number of times an n-gram g occurs in text T and $N(T)$ is the total number of times any n-gram occurs in text T, we can estimate the codelength H is defined as follows:

$$H(T, g) = -\log_2 P(T, g). \tag{4.1}$$

The codelength difference d computes the difference between the codelengths from two texts (T_E and $T_{\overline{E}}$, say) for n-gram g :

$$d(T_E, T_{\overline{E}}, g) = H(T_{\overline{E}}, g) - H(T_E, g). \tag{4.2}$$

This value will be high when there is a notable divergence in the n-gram probability between the two texts, and zero when the probability is similar. Most common n-grams (i.e. those comprised of function words and common content words) will have a codelength difference close to zero. In contrast, where probabilities differ significantly (usually for specific content words), the codelength difference will be 3 to 5 or more bits.

We can define a related value by using a threshold—this is non-zero when the value for d defined above exceeds a threshold θ and zero otherwise:

$$d'(T_E, T_{\overline{E}}, g, \theta) = \begin{cases} d(T_E, T_{\overline{E}}, g) & \text{if} \geq \theta \\ 0 & \text{otherwise.} \end{cases}$$

The idea is that the n-grams with the highest codelength difference values (greater than the threshold) should be the most useful in distinguishing between texts and the rest can be filtered out.

We can calculate the overall divergence D between the two texts T_E and $T_{\overline{E}}$ by summing the codelength differences for each n-gram g found in S_I for some codelength threshold as follows:

$$D(T_E, T_{\overline{E}}, S_I, \theta) = \sum_{g \in S_I} d'(T_E, T_{\overline{E}}, g, \theta).$$

Divergence defined in this way is similar to *codelength difference* defined by Walker et al. (2015) but without the use of the absolute function to ensure the difference values are always positive (here the difference values can go negative for n-grams that have a much higher probability in the complement set rather than the other way around). Also, the emphasis here is on texts for emotion recognition and a threshold mechanism has been added to filter out the low codelength values for n-grams that have similar probabilities between the texts being compared. The method here also sums the divergence values into a single overall total rather than analyse the values separately for information visualisation purposes.

Alternatively, we can also calculate the divergence for a separate testing text t with a set of n-grams S_t. We simply sum the codelength differences for n-grams from S_I that also occur in the testing text:

$$D(T_E, T_{\overline{E}}, S_I, S_t, \theta, L) = \sum_{g \in (S_I \cap S_t(L))} d'(T_E, T_{\overline{E}}, g, \theta) \tag{4.3}$$

where $S_t(L)$ is the set of n-grams in the testing set S_t where $n = L$ (for example, $S_t(1)$, $S_t(2)$ and $S_t(3)$ are the sets of unigrams, bigrams and trigrams in the testing set S_t respectively).

An alternative metric which we have found more effective for emotion recognition for the classifier defined above is based simply on comparing the number of common n-grams that have been filtered by the divergence metric according to some codelength threshold but resorts to using the divergence values to break ties. Using this approach, we define a new type of classifier

using the pseudo-code shown in Algorithm 1.

Algorithm 1: Pseudo-code for the ID classifier.

```
1  ID Classifier (Classes, TrainingTexts, TrainingNgrams,
   TestNgrams, θ, L)
2  bestClasses ← ∅
3  maxCount ← 0
4  for each class E in Classes do
5  |  CommonNgrams[E] ←
   |   (TrainingNgrams[E] ∩ TrainingNgrams[Ē]) with divergence
   |   values d' ≥ θ and n-gram length n = L
6  |  Divergences[E] ←
   |   D(TrainingTexts[E], TrainingTexts[Ē], CommonNgrams[E],
   |   TestNgrams, θ, L)
7  |  thisCount ← |CommonNgrams[E] ∩ TestNgrams|
8  |  if (thisCount > maxCount) then
9  |  |  bestClasses ← {E}
10 |  |  maxCount ← thisCount
11 |  else if (thisCount = maxCount) then
12 |  |  bestClasses ← bestClasses ∪ {E}
13 if (bestClasses = ∅) then
14 |  return "Unclassified"
15 else if (|bestClasses|= 1) then
16 |  return Class ∈ bestClasses
17 else
18 |  return Class with the smallest Divergence value in Divergences
```

The Algorithm has six input parameters: $Classes$, which is the set of class labels being used for the classification (e.g. *Anger* and *Happiness* for emotion recognition); $TrainingTexts$, which is an array of concatenated texts and their complements used for training, indexed by class (e.g. $TrainingTexts[E]$ and $TrainingTexts[\overline{E}]$ specifies the concatenated training text and its complement for emotion E, respectively); $TrainingNgrams$ is the array of n-grams in those texts for each class; $TestNgrams$ is the set of n-grams in the testing text being classified; θ is the codelength filter being used to filter the divergence values as described above; and L is the length of the n-grams being used in the analysis, where $n = L$. The Algorithm returns a single

class label as the output of the classification.

The algorithm works first (on Lines 4 to 12) by finding which classes ($bestClasses$) have the maximum number of common n-grams ($maxCount$) found in the intersection of the test n-grams with the n-grams common to the emotion and its complement S_I (Line 7), which is filtered by the codelength threshold θ and n-gram length filter L (lines 5 and 6). If there are ties, then this is broken by selecting (on Line 18) the class with the smallest divergence value for the list of Divergence values that were calculated on Line 6.

By varying the length of the n-grams used in the analysis (parameter L on Line 1 for Algorithm 1), the following classifiers can be defined: ID1, ID2 and ID3 (using $L = 1, 2, 3$ respectively). These are used in the experimental evaluation for Ekman six emotions classes below.

One problem with the Information Divergence type of classifier is that it requires common n-grams between the testing text and the training texts. In many cases, especially when using trigrams and when the testing text is relatively small, there can be few common n-grams. Also, there was not any problem of equal codelength between two classes, so the classifier needs to be applied to different problems and type of texts to find out how well the ID classifier performs. For equation 4.2 we tried the absolute value of difference rather than the difference in codelengths, but the experimental results showed that the classification using the non-absolute value of difference is more accurate. We tried to use the sum or the average value of all occurred ngrams in the tested text, but the experimental results showed that the way we used in our algorithm 1 is more effective. Further work is also needed to determine how well the classifier works with different datasets.

4.2 Experimental results

Our experimental results are presented in this section. The purpose of the experiments was to evaluate the effectiveness of the new classifier. The next subsection discusses the datasets that we used in the evaluation. This is followed by a discussion of the implementation details relating to the evaluation, some example outputs that were produced by the classifier, an analysis

of the confusion matrices produced during the classification experiments and a discussion of the unclassified texts that resulted. Finally, overall results for the new classifier are then discussed and compared to other classifiers.

4.2.1 Datasets for Emotion Recognition using Ekman's Emotions

As before, three datasets have been used for the experiments. Further details of these datasets are summarised in Table 4.1 and the number of texts that have been classified in each of the emotion classes is shown in Tables 4.2 and 4.3 along with the total number of texts and the total overall number of characters in the dataset. An obvious feature of these datasets is that the classes are clearly not balanced, especially the *Happiness* class, most notably for the LiveJournal dataset. The *Happiness* texts make up 36.9% of Alm's dataset, almost 41.5% of Aman's dataset and 57.5% of the LiveJournal dataset. Instances for just two of the classes (*Happiness* and *Sadness*) make up over three quarters of the LiveJournal dataset. Another obvious feature is the large difference in average size of the texts in the LiveJournal dataset (16,549.4 chars. on average) compared to the other two datasets (126.3 chars. for Alm and 79.7 chars. for Aman).

4.2.2 Implementation details used for the experimental evaluation

This section provides a brief description of the implementation details for the evaluation of the new classifier whose results are described in the next section.

- **Stage 1** (Processing training and testing data): When processing the training and testing data, ten-fold cross-validation is used. This stage uses the following steps. Firstly, for each fold, all the training texts for each emotion are concatenated together into a single text. This results in six emotion text files according to Ekman's basic emotion (or five files in the case for Alm's dataset). Secondly, for each emotion, we concatenate

Table 4.1: Details of the three datasets used in the experimental evaluation.

Dataset	**Description**
Alm (Alm et al., 2005)	This small dataset comprises 1207 annotated sentences taken from fairy tales labelled with five types of emotions based on Ekman's definition: (*Anger-Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise*). Alm decided to merge the *Anger* and *Disgust* emotions into one class since the emotions are similar.
Aman (Aman and Szpakowicz, 2007; Aman and Szpakowicz, 2008)	This dataset consists of 1291 emotion-rich sentences taken from web blogs labelled according to Ekman's six emotions. Seed words (such as 'awesome' and 'amused' for the *Happiness* emotion) were used to search for the web blogs.
LiveJournal (Mishne et al., 2005)	This large dataset, collected from LiveJournal (`http://www.LiveJournal.com/`), is composed of 815,494 web blogs also labelled by Ekman's six emotions according to 132 moods such as *happy*, *cheerful* and *sad* that the author of the blog chose to describe their mood when writing their post resulting in 6264 texts.

Table 4.2: Number of texts classified in each of the emotion classes for the three datasets.

Dataset	*Anger-Disgust*	*Anger*	*Disgust*	*Fear*	*Happiness*	*Sadness*	*Surprise*
Alm	218			166	445	264	114
Aman		179	172	115	536	173	115
LiveJournal		562	140	277	3601	1164	520

Table 4.3: Total number of texts classified for the three datasets.

Dataset	**Total**	**Characters**
Alm	1207	152,390
Aman	1290	102,943
LiveJournal	6264	103,665,221

the text from the other five emotions to create the complement text, i.e. concatenate *Disgust*, *Fear*, *Happiness*, *Sadness*, and *Surprise* text together to produce $\overline{Anger}$ and so on. Thirdly, n-grams (unigrams,

bigrams, and trigrams) are obtained for each of the emotions and emotion complement texts. Fourthly, all the common n-grams are obtained between these, then codelengths for each n-gram are computed for each using equation 4.1. Fifthly: codelength differences between codelengths of the common n-grams for each are computed using equation 4.2. Three dictionaries for each emotion are created (i.e. implemented using the dictionary data structure in Python), containing the unigrams, bigrams and trigrams produced whose codelength differences between a specific emotion and its complement are greater than or equal to the selected threshold.

The process requires pre-selecting the threshold (θ) for the values of the code length differences that produce the best classification results on the held-out testing data using Algorithm 1. In the experimental results reported below, a threshold of zero ($\theta = 0$) was found to produce a good trade-off between accuracy, precision, recall and F-measure, while at the same time minimising the number of unclassifieds, and therefore this can be applied when testing below we list results for a range of thresholds in order to determine which provides a good choice).

- **Stage 2** (Classifying using Information Divergence): For the test data in each fold, the method classifies using Algorithm 1 each testing text individually. First, the n-gram dictionaries of the six emotions will be available as a result of Stage 1. Then, depending on the classifier being adopted (ID1, ID2, ID3), the necessary dictionary or dictionaries (i.e. unigrams, bigrams or trigrams) will be consulted. This involves first computing the number of n-grams in the testing text that also appear in each emotion dictionary. The emotion which has the highest n-gram count will be selected as the emotion of that text as the result of the classification. Sometimes, two or more emotions have the same number of n-grams appearing in the testing text. To solve this case, the summation of codelengths for all n-grams that appear in the testing text and the emotion dictionary will be used, and the emotion with the smallest summation will be selected as the emotion of the text. The reason for

choosing the smallest summation is that the cost of compressing this text should be less for this emotion compared to other emotions.

4.2.3 Example outputs produced by the ID1 classifier

This section provides some example outputs produced by the ID1 classifier in order to illustrate important aspects of how it works.

Table 4.4 to Table 4.7 list the top 30 unigrams for the ID1 classification for some emotions such as the *Happiness*, *Sadness* and *Anger* classes for Alm's and Aman's datasets respectively. These were produced using training data taken from one of the folds during the ten-fold cross-validation process during Stage 1 described above and are provided to illustrate sample output for the ID1 classifier. The unigrams in the tables (shown in the second column) are arranged in descending order according to the codelength difference values (shown in the last column) as computed using Equation 4.2. The ranking number according to the ordering is shown in the first column. Unigrams that closely reflect the particular emotion are highlighted in bold font in order to illustrate the effectiveness of the ranking. Unigrams that end up with a similar codelength difference are ranked in this table in reverse alphabetical order. (This is a common occurrence—e.g. ranks 5 to 10 and ranks 16 to 24 for Table 4.4.) The frequency counts $c(T_E, g)$ and $c(T_{\overline{E}}, g)$ of the number of times each unigram g for emotion E occurs in the emotion training data and in its complement $\overline{E}$ respectively are shown in columns three and four. The codelengths $H(T_E, g)$ and $H(T_{\overline{E}}, g)$ are shown in columns five and six and are calculated according to Equation 4.1.

As Alm's and Aman's datasets are not large, the frequency counts are often consequently relatively low, except for more common words such as "wept" which appears 20 times in the *Sadness* training data for Alm's dataset compared to only twice in its complement and "poor" which appears 27 times in the same training data compared to 13 times in its complement. In the *Happiness* training data for Aman's dataset, the word "happy" appears 18 times compared to only once in its complement, and "love" appears 38 times compared to just three times in its complement. In the *Anger* train-

Table 4.4: The unigram codelength differences for the ID1 classification in the Alm's dataset for the *Sadness* class for one of the folds.

Rk.	Unigram (g)	$c(T_S, g)$	$c(T_{\overline{S}}, g)$	$H(T_S, g)$	$H(T_{\overline{S}}, g)$	$d(T_S, T_{\overline{S}}, g)$
1	**wept**	20	2	8.267	13.422	5.155
2	**grief**	8	1	9.589	14.422	4.833
3	**alone**	6	1	10.004	14.422	4.418
4	**wept.**	5	1	10.267	14.422	4.155
5	**wiped**	4	1	10.589	14.422	3.833
6	**weep**	4	1	10.589	14.422	3.833
7	wandered	4	1	10.589	14.422	3.833
8	**tears,**	4	1	10.589	14.422	3.833
9	longer	4	1	10.589	14.422	3.833
10	**died.**	4	1	10.589	14.422	3.833
11	**cry**	7	2	9.782	13.422	3.640
12	**tears**	10	3	9.267	12.837	3.570
13	**sunk**	3	1	11.004	14.422	3.418
14	speak	3	1	11.004	14.422	3.418
15	**sorrow**	6	2	10.004	13.422	3.418
16	save	3	1	11.004	14.422	3.418
17	hard	3	1	11.004	14.422	3.418
18	cock	3	1	11.004	14.422	3.418
19	child,"	3	1	11.004	14.422	3.418
20	begged	3	1	11.004	14.422	3.418
21	bed,	3	1	11.004	14.422	3.418
22	although	3	1	11.004	14.422	3.418
23	against	3	1	11.004	14.422	3.418
24	Thus	3	1	11.004	14.422	3.418
25	forest,	5	2	10.267	13.422	3.155
26	poor	27	13	7.834	10.721	2.887
27	youth	2	1	11.589	14.422	2.833
28	way,	2	1	11.589	14.422	2.833
29	watched	2	1	11.589	14.422	2.833
30	upstairs	2	1	11.589	14.422	2.833

ing data for Aman's dataset, the word "pissed" appears 4 times compared to only once in its complement, and "annoy" appears 3 times compared to only once in its complement. The difference in codelengths then determines how divergent each unigram is compared to other unigrams, and how high up this ranking table it appears as a consequence.

Table 4.5: The unigram codelength differences for the ID1 classification in the Aman's dataset for the *Anger* class for one of the folds.

Rk.	Unigram (g)	$c(T_A, g)$	$c(T_{\overline{A}}, g)$	$H(T_A, g)$	$H(T_{\overline{A}}, g)$	$d(T_S, T_{\overline{A}}, g)$
1	off.	7	1	8.560	13.962	5.401
2	**pissed**	4	1	9.367	13.962	4.594
3	**annoy**	3	1	9.783	13.962	4.179
4	you,	2	1	10.367	13.962	3.594
5	**worse**	2	1	10.367	13.962	3.594
6	**terrorists**	2	1	10.367	13.962	3.594
7	taken	2	1	10.367	13.962	3.594
8	spread	2	1	10.367	13.962	3.594
9	safety	2	1	10.367	13.962	3.594
10	putting	2	1	10.367	13.962	3.594
11	push	2	1	10.367	13.962	3.594
12	post	2	1	10.367	13.962	3.594
13	plot	2	1	10.367	13.962	3.594
14	off,	2	1	10.367	13.962	3.594
15	**killing**	2	1	10.367	13.962	3.594
16	**issues**	2	1	10.367	13.962	3.594
17	half	2	1	10.367	13.962	3.594
18	full	2	1	10.367	13.962	3.594
19	**fuck**	6	3	8.783	12.377	3.594
20	ex	2	1	10.367	13.962	3.594
21	cut	2	1	10.367	13.962	3.594
22	classes	2	1	10.367	13.962	3.594
23	busy	2	1	10.367	13.962	3.594
24	bad.	2	1	10.367	13.962	3.594
25	ass	2	1	10.367	13.962	3.594
26	**argument**	2	1	10.367	13.962	3.594
27	**annoying**	2	1	10.367	13.962	3.594
28	above	2	1	10.367	13.962	3.594
29	10	2	1	10.367	13.962	3.594
30	**fucking**	7	4	8.560	11.962	3.401

Examining the unigrams in the four tables, we can see some unigrams that reflect the different emotions appearing in the tables in the top 30 unigrams that are listed (i.e. the unigrams that are shown in bold font). Some examples are the unigrams "wept", "grief", "alone", "wiped" and "tears" for the *Sadness* class, the unigrams "happy","love" and "good" for the *Happiness*

Table 4.6: The unigram codelength differences for the ID1 classification in the Alm's dataset for the *Happiness* class for one of the folds.

Rk.	Unigram(g)	$c(T_H, g)$	$c(T_{\overline{H}}, g)$	$H(T_H, g)$	$H(T_{\overline{H}}, g)$	$d(T_S, T_{\overline{S}}, g)$
1	**merry**	14	1	9.492	14.103	4.612
2	**glad**	13	1	9.598	14.103	4.505
3	**pretty**	7	1	10.491	14.103	3.612
4	other,	7	1	10.491	14.103	3.612
5	**lucky**	7	1	10.491	14.103	3.612
6	**danced**	14	2	9.491	13.103	3.612
7	song	6	1	10.714	14.103	3.389
8	light	6	1	10.714	14.103	3.389
9	**feast**	6	1	10.714	14.103	3.389
10	lived	16	3	9.299	12.518	3.220
11	world,	5	1	10.977	14.103	3.126
12	**lovely**	5	1	10.977	14.103	3.126
13	hand	10	2	9.977	13.103	3.126
14	everything	5	1	10.977	14.103	3.126
15	everyone	5	1	10.977	14.103	3.126
16	piece	4	1	11.299	14.103	2.805
17	money	4	1	11.299	14.103	2.805
18	horses,	4	1	11.299	14.103	2.805
19	**dancing**	4	1	11.299	14.103	2.805
20	**love**	7	2	10.491	13.103	2.612
21	**happy**	21	6	8.907	11.518	2.612
22	**bright**	7	2	10.491	13.103	2.612
23	**laughed**	10	3	9.977	12.518	2.541
24	while,	3	1	11.714	14.103	2.389
25	was!	3	1	11.714	14.103	2.389
26	used	3	1	11.714	14.103	2.389
27	touched	3	1	11.714	14.103	2.389
28	sunshine	3	1	11.714	14.103	2.389
29	son,	3	1	11.714	14.103	2.389
30	pig	3	1	11.714	14.103	2.389

class, and the unigrams "pissed", "annoy", "worse", "killing" and "issues" for the *Anger* class. This is also despite unigrams being used in the analysis as opposed to bigrams and trigrams which perhaps should provide a more effective means for distinguishing texts as they often are when used for language modelling, for example (although we have found that unigrams are

Table 4.7: The unigram codelength differences for the ID1 classification in the Aman's dataset for the *Happiness* class for one of the folds.

Rk.	Unigram(g)	$c(T_H, g)$	$c(T_{\overline{H}}, g)$	$H(T_H, g)$	$H(T_{\overline{H}}, g)$	$d(T_S, T_{\overline{S}}, g)$
1	**happy**	18	1	8.435	13.563	5.128
2	**love**	38	3	7.357	11.978	4.621
3	year	11	1	9.145	13.563	4.417
4	**good**	23	3	8.081	11.978	3.896
5	**well,**	6	1	10.020	13.563	3.543
6	**great**	12	2	9.020	12.563	3.543
7	staying	5	1	10.283	13.563	3.280
8	thing.	4	1	10.605	13.563	2.958
9	seeing	4	1	10.605	13.563	2.958
10	myself.	4	1	10.605	13.563	2.958
11	**hope**	4	1	10.605	13.563	2.958
12	close	4	1	10.605	13.563	2.958
13	baby	7	2	9.797	12.563	2.765
14	working	3	1	11.020	13.563	2.543
15	went	12	4	9.020	11.563	2.543
16	well.	3	1	11.020	13.563	2.543
17	way,	3	1	11.020	13.563	2.543
18	w/	3	1	11.020	13.563	2.543
19	single	3	1	11.020	13.563	2.543
20	**pride**	3	1	11.020	13.563	2.543
21	ones	3	1	11.020	13.563	2.543
22	nice	9	3	9.435	11.978	2.543
23	music	3	1	11.020	13.563	2.543
24	is,	3	1	11.020	13.563	2.543
25	everyday	3	1	11.020	13.563	2.543
26	brought	3	1	11.020	13.563	2.543
27	**better**.	3	1	11.020	13.563	2.543
28	ball	3	1	11.020	13.563	2.543
29	We	18	6	8.435	10.978	2.543
30	I'll	3	1	11.020	13.563	2.543

most effective for our Information Divergence classifier as discussed below).

Note the absence (as expected) of most of the function words in these tables (such as "the", "a" and "and") despite these all appearing with high frequency in the training and complement texts. This is because the probability of these words do not diverge between the training texts and their

complements. Note also the absence in these four examples of content words that are proper names (such as "Hans," and "Snowdrop"), unless in the training text they are associated with a particular emotion.

4.2.4 Confusion matrix results for each classifier

The results from compiling the confusion matrices for each dataset for the ID1 classifier are discussed in this section.

In order to gain insight into and illustrate how well the ID1 classifier performs, and provide a more detailed analysis which reports important aspects of the classification not revealed by the overall results listed below, Tables 4.8 to 4.16 list the confusion matrices that were produced for the three different datasets and various thresholds (θ).

Table 4.8: Confusion matrix for Ekman's emotions classification for Alm's Dataset using the ID1 classifier for threshold $\theta = -2$.

	Threshold $\theta = -2$					
	A/D	F	H	$Sd.$	$Su.$	U
A/D	**65**	14	87	42	5	5
F	18	**12**	91	38	2	5
H	47	15	**216**	152	8	7
$Sd.$	12	12	177	**59**	3	1
$Su.$	19	7	54	20	**3**	11

Table 4.9: Confusion matrix for Ekman's emotions classification for Alm's Dataset using the ID1 classifier for threshold $\theta = 0$.

	Threshold $\theta = -2$					
	A/D	F	H	$Sd.$	$Su.$	U
A/D	**58**	36	58	35	30	1
F	17	**61**	49	31	8	0
H	35	65	**235**	75	35	0
$Sd.$	27	60	80	**77**	20	0
$Su.$	20	19	32	19	**24**	0

Each row of the matrix provides the number of instances in the actual (ground-truth) classes that were assigned the predicted class shown in the respective columns. These are labelled as follows: A/D for the $Anger/Disgust$

Table 4.10: Confusion matrix for Ekman's emotions classification for Alm's Dataset using the ID1 classifier for threshold $\theta = 2$.

	Threshold $\theta = -2$					
	A/D	F	H	$Sd.$	$Su.$	U
A/D	**72**	36	14	11	73	12
F	34	**69**	5	10	36	12
H	124	72	**90**	31	111	17
$Sd.$	81	41	15	**56**	65	6
$Su.$	36	21	2	2	**46**	7

Table 4.11: Confusion matrix for Ekman's emotions classification for Aman's Dataset using the ID1 classifier for threshold $\theta = -2$.

	Threshold $\theta = -2$						
	A	D	F	H	$Sd.$	$Su.$	U
A	**55**	13	4	101	3	2	1
D	48	**16**	6	85	9	1	7
F	23	5	**9**	69	6	3	0
H	85	29	25	**320**	39	16	22
$Sd.$	40	8	10	98	**16**	1	0
$Su.$	18	2	8	74	4	**8**	1

Table 4.12: Confusion matrix for Ekman's emotions classification for Aman's Dataset using the ID1 classifier for threshold $\theta = 0$.

	Threshold $\theta = 0$						
	A	D	F	H	$Sd.$	$Su.$	U
A	**57**	20	23	15	53	10	1
D	36	**16**	17	31	55	10	7
F	16	10	**15**	20	45	9	0
H	66	56	48	**142**	150	52	22
$Sd.$	22	11	12	28	**92**	8	0
$Su.$	16	7	17	21	35	**18**	1

class; A for the *Anger* class; D for the *Disgust* class; F for the $Fear$ class; H for the *Happiness* class; $Sd.$ for the *Sadness* class; and $Su.$ for the *Surprise* class. The column labelled U lists the number of unclassified texts for each class (i.e. the classifier returned the $Unclassified$ class).

The numbers in the diagonal written in bold font represent the correctly classified texts for each class. The numbers off-diagonal represent the mis-

Table 4.13: Confusion matrix for Ekman's emotions classification for Aman's Dataset using the ID1 classifier for threshold $\theta = 2$.

	Threshold $\theta = 2$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**72**	17	28	11	20	16	15
D	46	**35**	24	4	20	7	36
F	27	7	**38**	3	19	9	12
H	103	42	69	**95**	64	45	118
Sd.	41	14	35	10	**45**	13	14
Su.	22	3	22	7	13	**36**	12

Table 4.14: Confusion matrix for Ekman's emotions classification for LiveJournal's Dataset using the ID1 classifier for threshold $\theta = -2$.

	Threshold $\theta = -2$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**0**	0	0	562	0	0	0
D	0	**0**	0	140	0	0	0
F	0	0	**0**	277	0	0	0
H	0	0	0	**3600**	1	0	0
Sd.	0	0	0	1163	**1**	0	0
Su.	0	0	0	520	0	**0**	0

Table 4.15: Confusion matrix for Ekman's emotions classification for LiveJournal's Dataset using the ID1 classifier for threshold $\theta = 0$.

	Threshold $\theta = 0$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**118**	0	0	217	157	70	0
D	22	**3**	0	58	39	18	0
F	10	0	**4**	141	77	45	0
H	80	0	2	**2773**	499	247	0
Sd.	42	0	0	334	**673**	115	0
Su.	15	0	0	174	176	**155**	0

classified texts. Formally, for a specific class C:

- The True Positives (TP_C) is the number of the correctly classified instances for a class C. This is the number that appears in the main diagonal of the confusion matrix for the row labelled C.

Table 4.16: Confusion matrix for Ekman's emotions classification for LiveJournal's Dataset using the ID1 classifier for threshold $\theta = 2$.

	Threshold $\theta = 2$						
	A	D	F	H	$Sd.$	$Su.$	U
A	**302**	0	2	0	2	256	0
D	57	**14**	4	0	0	65	0
F	90	6	**12**	0	0	169	0
H	1490	10	21	**6**	6	2068	0
$Sd.$	417	8	4	1	**3**	731	0
$Su.$	139	1	2	0	0	**378**	0

- The True Negatives (TN_C) is the number of all instances that do not belong to a certain class C and are classified as not belonging to that class. This is the sum of all the values in the confusion matrix excluding the row and column that are labelled C.

- The False Positives (FP_C) is the number of all instances that are classified as a certain class C but they actually do not belong to that class. This is the sum of all the values in the column labelled C excluding the diagonal value TP_C.

- The False Negatives (FN_C) is the number of all instances that are classified as not belonging to a certain class C but they actually belong to that class. This is the sum of all the values in the row labelled C excluding the diagonal value TP_C.

Analysing each of the confusion matrices tables in turn, we can see a number of noticeable results for the Information Divergence based classifier (ID1) on the three datasets. For Alm's dataset (Table 4.8 to Table 4.10), the least number of unclassified texts as listed in the column labelled U occur when the threshold $\theta = 0$ (Table 4.9).

For the confusion matrices produced for Aman's dataset (Table 4.11 to Table 4.13), there are also a large number of mis-classifications off-diagonal.

For the confusion matrices produced for the Livejournal dataset (Tables 4.14 to Table 4.16) which have a significantly larger amount of training texts than the other two datasets, there are no unclassified texts.

In summary, the threshold $\theta = 0$ seems to be a good choice across all three datasets when weighing up issues to do with the number of unclassified texts and issues to do with the number of true positives, false negatives and false positives.

4.2.5 Unclassified texts

One issue with the ID1 classifier is that Information Divergence works only with common unigrams that occur between the emotion and its complement. (This is the area labelled I shown in Figure 4.1). It is clear from the figure that the number of the common n-grams considered by the Information Divergence classification method may potentially not be enough to train a classifier to be effective at classifying all the texts resulting in unclassified documents. If the tested text contains unigrams that do not exist in the set of common unigrams between the emotion training text and its complement, this text will not be classified, which is evidenced by the number of unclassifieds for Aman's dataset already discussed. (This issue is even more prevalent for the ID2 and ID3 classifiers as well which is a major contributor to these classifiers not performing as well as ID1, as shown by the experimental results below).

Examining the texts that were unclassified for the three datasets, only a single text out of 1207 texts is unclassified for Alm's dataset. (Recall that this dataset comprises single sentence texts from fairy tales). This unclassified text in the *Anger* class consists of a single unigram, *"Humph!*, which does not appear in the training data.

Table 4.17 shows all the unclassified blogs for the ID1 classifier for Aman's dataset. Note that although this dataset comprises web blogs, the examples in the table are all very short. Most of these blogs contain a single word, the exceptions being the blogs numbered 1, 15 and 23 in the table (such as "Poopy poopy poopy" for blog 1). Note that some of the unclassified blogs occur multiple times in the dataset (for example, "ugh." and "lol."). The reason why these texts are not classified is because as stated none of the unigrams that compose these texts appear in the common unigrams between

the class training text and its complement for the current fold being processed during the cross-validation process.

Table 4.17: All the unclassified blogs for Aman's dataset after applying Ekman's emotions classification using the ID1 classifier.

No.	**Blog**	**Class**	**No.**	**Blog**	**Class**
1	Poopy poopy poopy	*Anger*	17	lol.	*Happiness*
2	WTF)	*Disgust*	18	LOL.	*Happiness*
3	UGH!	*Disgust*	19	Wow.	*Happiness*
4	Ew.	*Disgust*	20	Woot.	*Happiness*
5	Urgh.	*Disgust*	21	LOL.	*Happiness*
6	ugh.	*Disgust*	22	lol jk)	*Happiness*
7	ugh.	*Disgust*	23	Fun fun fun.	*Happiness*
8	ugh.	*Disgust*	24	Wooo!	*Happiness*
9	lol.	*Happiness*	25	lol).	*Happiness*
10	lol.	*Happiness*	26	Wow.	*Happiness*
11	lol.	*Happiness*	27	LOL.	*Happiness*
12	lol.	*Happiness*	28	:D!	*Happiness*
13	lol.	*Happiness*	29	Lovely.	*Happiness*
14	lol).	*Happiness*	30	wowo.	*Happiness*
15	:D lol wow.	*Happiness*	31	Eh.	*Surprise*
16	lol.	*Happiness*			

As stated above, there were no unclassified texts for the ID1 classifier for the LiveJournal dataset during the ten-fold cross-validation process due to the large amount of training text that is available when this dataset is processed.

4.2.6 Overall Results

The overall results of the experiments with the ID1 classifier are shown in Table 4.18.

Table 4.18 lists the datasets that was experimented with in the first column, the threshold θ in the second column, and then in the remaining columns the Accuracy, Precision, Recall and F-measure along with the number of texts that were assigned the $Unclassified$ class. The highest column value for each dataset is shown in bold font.

Table 4.18: Ekman's emotions classification for the three datasets using the new ID1 classifier.

Dataset	**Thres.** θ	**Acc.**	**Prec.**	**Rec.**	**F-measure.**	**Unclass.**
Alm	2.0	71.9	**0.37**	0.31	0.34	54
	1.5	72.8	0.35	**0.35**	**0.35**	13
	1.0	70.8	0.34	0.32	0.33	5
	0.5	71.2	0.33	0.32	0.33	1
	0.0	**75.1**	0.33	0.33	0.33	1
	−0.5	74.7	0.32	0.28	0.30	1
	−1.0	74.4	0.31	0.28	0.29	1
	−1.5	72.9	0.24	0.25	0.25	1
	−2.0	71.9	0.23	0.22	0.22	1
Aman	2.0	77.6	**0.33**	**0.28**	**0.30**	207
	1.5	76.1	0.28	0.27	0.28	82
	1.0	75.0	0.28	0.27	0.27	54
	0.5	75.2	0.28	0.27	0.27	47
	0.0	75.9	0.24	0.25	0.25	31
	−0.5	77.3	0.23	0.23	0.23	31
	−1.0	**78.6**	0.25	0.23	0.24	31
	−1.5	78.0	0.24	0.22	0.23	31
	−2.0	78.0	0.24	0.21	0.22	31
LiveJournal	2.0	70.5	0.33	0.24	0.27	0
	1.5	70.7	0.32	0.25	0.28	0
	1.0	73.4	0.35	0.29	0.32	0
	0.5	86.1	0.48	**0.35**	0.40	0
	0.0	**86.5**	**0.58**	0.32	**0.41**	0
	−0.5	86.0	0.31	0.17	0.22	0
	−1.0	85.8	0.18	0.17	0.17	0
	−1.5	85.8	0.15	0.17	0.16	0
	−2.0	85.8	0.18	0.17	0.17	0

Overall, several noticeable trends in the results can be discerned from Table 4.18. The number of unclassifieds for the smaller datasets (Alm's and Aman's) significantly decreases as the threshold θ gets smaller, and reaches a minimum when $\theta = 0$ or $\theta = 0.5$ and any reduction in the value of θ has no further effect. For the large LiveJournal dataset, the number of unclassifieds is zero for all thresholds due to the much greater amount of training data available.

Accuracy peaks for Alm's and the LiveJournal datasets when $\theta = 0$, and for Aman's dataset when $\theta = -1$. The best values for precision, recall and F-measure are highest for the smaller datasets when θ is higher (0.5 or 2.0). However, the results for Alm's dataset are not significantly worse when $\theta \geq 0$. In contrast, for the LiveJournal dataset, the best values for precision, recall and F-measure occur when $\theta = 0, 0.5$. However, the highest values for precision, recall and F-measure for Aman's dataset occur when $\theta = 2$, but this results in a significantly higher number of classifieds (207) compared to the other threshold values. Choosing $\theta = 0$ for this dataset results in a notable drop in precision, recall and F-measure unlike for the other two datasets, but this threshold value minimises the number of unclassifieds.

In summary, setting the threshold θ to a value of zero provides a good trade-off between accuracy, precision, recall and F-measure, while at the same time minimising the number of unclassifieds.

We also performed some experiments with the ID2 and ID3 classifiers which used bigrams and trigrams respectively (rather than unigrams as for ID1). Some of the results for these experiments are shown in Table 4.19 which also includes the ID1 results from above for comparison. In the table, the first column lists the dataset experimented with, the second column the classifer used (ID1, ID2 or ID3), the third column the threshold θ used (in this case, only results for $\theta = 0$ have been shown), and subsequent columns list the accuracy, precision, recall, F-measure and the number of unclassified documents. The results shown in the table, and for the different threshold θ values not shown, indicate similar trends to above for different thresholds, and that ID1 clearly outperforms both the ID2 and ID3 classifiers for all the datasets. The number of unclassified documents also significantly increases for ID2 and ID3 for the smaller datasets due to the lack of training data when using longer n-grams in the classification. Even when the very large LiveJournal dataset was used, there was a small increase in the number of unclassified text (from 0 to 3). The high number of unclassified documents (930) for the ID3 classifier for Aman's dataset clearly has a significant impact on the effectiveness of the classifier, with F-measure dropping to just 0.02.

Table 4.19: Ekman's emotions classification for the three datasets using the new classifiers ID1, ID2 and ID3.

Dataset	Class.	Thres.	Acc.	Prec.	Rec.	F-meas.	Unclass.
Alm	ID1	0	75.1	0.33	0.33	0.33	1
	ID2	0	73.2	0.29	0.27	0.28	34
	ID3	0	72.8	0.18	0.12	0.14	429
Aman	ID1	0	75.9	0.24	0.25	0.25	31
	ID2	0	76.0	0.21	0.18	0.19	214
	ID3	0	79.5	0.05	0.01	0.02	930
LiveJournal	ID1	0	86.5	0.58	0.32	0.41	0
	ID2	0	78.3	0.53	0.26	0.35	0
	ID3	0	76.5	0.43	0.25	0.31	3

4.2.7 Comparison with other published results

Table 4.20 shows a comparison between our results for ID1 when using threshold $\theta = 0$ and previously published results for the three datasets. The table lists the dataset in column 1, the classifier in column 2, and the classification results for Ekman's emotions in terms of accuracy, precision, recall and F-measure in the remaining columns along with a reference to where the results have been published in the final column. The gaps in the table are because some publications only provided results in terms of accuracy. The best result for each column and dataset are shown in bold font.

Chaffar and Inkpen (2011) reported accuracy results for Alm's dataset using the standard feature-based Naïve Bayes, J48 and SMO machine learning classifiers (where the features were based on words) (Chaffar and Inkpen, 2011). Ghazi et al. (2010) reported accuracy results for both Alm's and Aman's datasets using two variations of the SMO classifier ("flat" and "two-level") (Ghazi et al., 2010). The results recently published by Almahdawi & Teahan used the character-based PPM text compression scheme (Almahdawi and Teahan, 2018).

The results show that the ID1 classifier is competitive with the PPM algorithm, performing significantly better in all measures for Alm's dataset, for three measures for the LiveJournal dataset, but not as well for Aman's dataset. This is probably due to the very different nature of the texts being

Table 4.20: Comparison between the result of the ID1 classifier and the results of this thesis, Chaffar et al. (Chaffar and Inkpen, 2011) and Ghazi et al. (Ghazi et al., 2010).

Dataset	**Classifier**	**Acc.**	**Prec.**	**Rec.**	**F-meas.**	**Reference**
Alm	ID1	**75.1**	**0.33**	**0.33**	**0.33**	This thesis
	PPM	69.2	0.26	0.24	0.25	This thesis
	Naïve Bayes	54.9				Chaffar and Inkman(2011)
	J48	47.5				Chaffar and Inkman(2011)
	SMO	61.9				Chaffar and Inkman(2011)
	Flat SMO	57.4				Ghazi et al. (2010)
	Two-level SMO	59.1				Ghazi et al. (2010)
Aman	ID1	75.9	0.24	0.25	0.25	This thesis
	PPM	**85.0**	**0.50**	**0.42**	**0.46**	This thesis
	Flat SMO	57.4				Ghazi et al. (2010)
	Two-level SMO	59.1				Ghazi et al. (2010)
LiveJournal	ID1	86.5	0.58	**0.32**	**0.41**	This thesis
	PPM	**87.4**	**0.69**	0.26	0.38	This thesis

classified for the three datasets, with Alm's dataset consisting of single sentences from fairy tales, with Aman's dataset consisting of short web blogs and with the LiveJournal dataset also consisting of web blogs, but being considerably longer in length. PPM also uses character n-grams for its classification, whereas ID1 uses word unigrams. Both the ID1 and PPM classifiers significantly outperform the other feature-based machine learning classifiers.

4.3 Conclusion

In this chapter, a new method of text classification based on Information Divergence has been introduced and applied to the problem of recognising Ekman's six basic emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*

and *Surprise*). The new method is a word n-gram based classifier that uses a divergence measure to help in identifying important n-grams that help with the classification. It does this by calculating the compression codelength differences between n-grams that appear in both the training data for the class and its complement.

We used three datasets to evaluate the new method—Alm's fairy tales dataset; Aman's web blog dataset; and a LiveJournal dataset also consisting of web blogs. The results show that the new method is effective when compared with other classification methods. Although the new approach can be applied to any length n-gram, experimental results show that unigrams work best. Results in terms of accuracy, precision, recall and F-measure for the three datasets are competitive when compared to the compression-based classifier, PPM, described in Chapter 3 which relies on character n-grams rather than word n-grams. The results for the new classifier also significantly outperform results achieved by traditional feature-based classifiers such as Naïve Bayes, J48 and SMO.

Chapter 5

Arabic Emotion Recognition

5.1 Background and Motivation

A primary objective of the research described in this chapter is to automatically recognising emotions in Arabic text (specifically, the Iraqi dialect) according to Ekman's (Ekman, 1999) fine-grained emotion classification (*Anger*, *Disgust*, *Fear*, *Happiness*, *Sadness*, *Surprise*). To achieve this goal, a suitable 'gold standard' dataset of Arabic text for research experiments is needed where emotions in the text have been manually annotated. For evaluating any automatic learning system, annotated data is a prerequisite for performing a robust evaluation. However, our research in automatically recognising emotions in Arabic text is obstructed by the lack of publicly available annotated data for written Arabic text. This chapter is based on the paper (Almahdawi and Teahan, 2019).

Automatic text processing of Arabic language text has become a goal for many Natural Language Processing and text mining researchers (Ahmed and Nürnberger, 2009),(Azmi and Alzanin, 2014),(Ahmed and Nürnberger, 2011). However, despite the Arabic language being one of the top five spoken languages, there is a lack of emotion and sentiment datasets. This is one of the reasons for creating a new dataset for emotion recognition.

A study on available datasets shows none of the available Arabic corpora is suitable for research related to emotion recognition. We considered

the Arabic Twitter corpus (Refaee and Rieser, 2014) which consists of 8,868 tweets which has been annotated with a particular positive, negative and neutral sentiment. But this corpus is inappropriate for this research as it does not support fine-grained emotion classification. Instead it supports only negative, positive, and neutral states. The Arabic tweets corpus composed by Abdulla et al. (2013) consists of 1000 positive tweets and 1000 negative tweets and also does not supporting Ekman's emotion recognition (Abdulla et al., 2013). Instead, it supports the polarity of the tweet as positive and negative, so it is unsuitable for this research. Another corpus called AWATIF created by Abdul-Mageed and Diab (2012) is a multi-genre Modern Standard Arabic corpus for the purpose of sentiment analysis and subjectivity which is also not suitable for this research (Abdul-Mageed and Diab, 2012). The corpus called LABR for sentiment analysis in the Arabic language consists of 63,000 books reviews by Aly and Atiya (2013). One more available dataset called HAAD is composed of book reviews in the Arabic language but this dataset is again annotated just for sentiment analysis purposes, not for Ekman's emotions (Al-Smadi et al., 2015).

One of the closest corpora to our research is the corpus of micro tweets developed by Al-Aziz et al. (2015). It consists of 1552 tweets labelled with five emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, and *Sadness*). Also it is written in the Egyptian dialect and Modern Standard Arabic texts (Basili and Pazienza, 1997). Unfortunately, this corpus is not appropriate to our research as it supports just five emotions, not Ekman's six emotions. In addition, the text of this corpus is in the Egyptian regional dialect, whereas the goal of our research is to focus on another regional dialect—Iraqi.

According to the previous limitations of finding appropriate Arabic corpora to meet the requirements of this research, we decided to develop a new dataset. The most important consideration in choosing data in this research is the requirement that the text should often contain emotion-rich expressions. Another important consideration is the data should include many examples for all the emotion classes considered in this research. A question arises concerning where this type of data that expresses personal emotions can be obtained. A survey by Salem reveals that social media is the most

appropriate place for 58% of Arab people to express their emotions toward their government's policies or services. 86% of these people who express their emotions in social media are using Facebook as a platform, 28% uses Twitter, and 28% use WhatsApp and other messaging applications, as shown in Figure 5.1 (Salem, 2017).

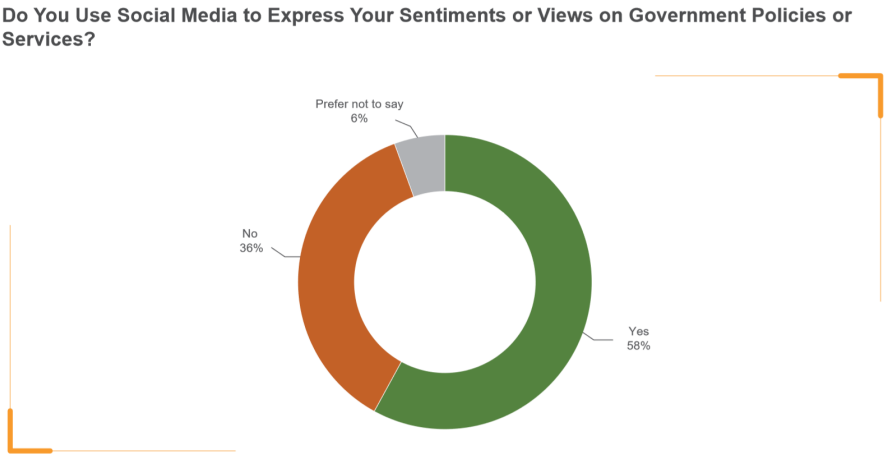


Figure 5.1: The overall percentages of people who use social media to express their emotions (Salem, 2017).

Figure 5.2 shows the percentages of Arab people using social media platforms to express their emotions. The figure shows that Arab people prefer using Facebook as a platform compared to other platforms such as Twitter, Instagram and Snapchat.

Due to this study, Facebook has been chosen as the platform for collecting data in this research. Other considerations for dealing with public posts written by people are the misspelling of words and slang words included in the text. The classification system potentially needs to deal with these in some way.

This chapter is organised as follows. Firstly, this chapter creates an Arabic emotion dataset based on the six Ekman's emotions. Secondly, this chapter evaluates this Arabic dataset using external judges to ascertain the quality of the dataset. Thirdly, this chapter conducts an experimental evaluation using the new dataset by investigating how well various classifiers perform at identifying the emotions in the texts. The results are then discussed for each

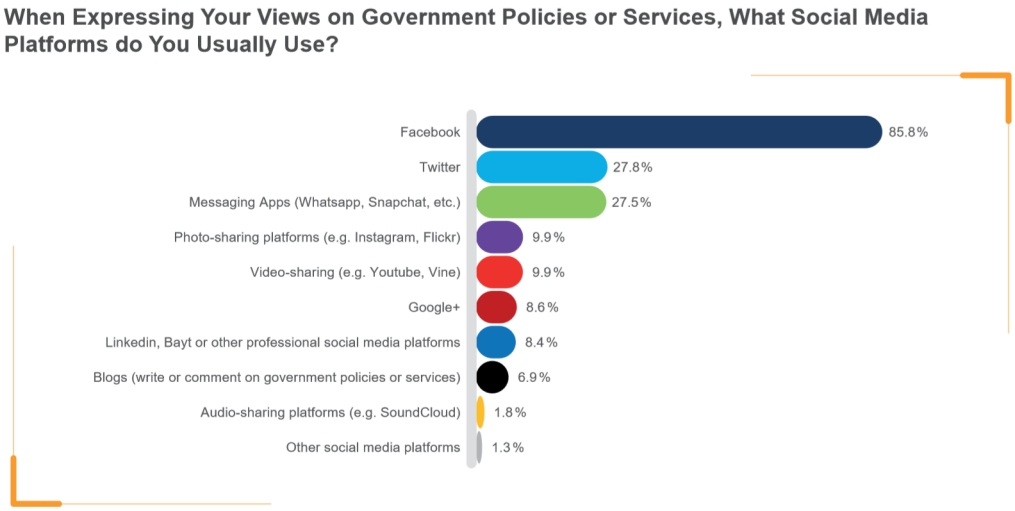


Figure 5.2: The overall percentages of Arab people using social media to express their emotions (Salem, 2017).

classifier and compared in the final section.

5.2 Creating the New Arabic Dataset for Emotion Recognition

In this section, the process of creating a new Arabic dataset for emotion recognition will be described. We have named the new dataset IAEC (which stands for Iraqi Arabic Emotions Corpus). As mentioned in the previous section, the data will be collected from the Facebook platform.

Facebook has features that help users searching for specific information. When a Facebook user is writing his/her post, Facebook supports users declaring his/her emotional state. This emotional state will appear in the post. This is shown in Figure 5.3 which is an example of a Facebook post declaring the emotional state of the user (underlined in red at the top of the figure).

As obvious in Figure 5.3, the user in the post has declared their emotional state using the word "feeling" followed by the emotional state which is "happy". It is this "feeling" feature that we can use to help in searching

Figure 5.3: The post of a user declaring his emotional state in Facebook.

posts for specific emotions. Specifically for the purposes of this research, we can use the search bar of Facebook to search for one of the Ekman's emotions. This is shown in Figure 5.4 which shows the query for searching about posts that declare the angry emotion state.

Figure 5.4: The query for searching Facebook for posts that have the angry emotional state.

After specifying this query, Facebook displays all posts that have the angry emotion. On the left side of the page, Facebook provides the user with a filter to search for more specific posts such as posts from friends or from all users, posts from users around the world or from a specific geographic place, posts from a specific date or not and so on. Figure 5.5 illustrates the filter

that helps the user to search for more specific posts.

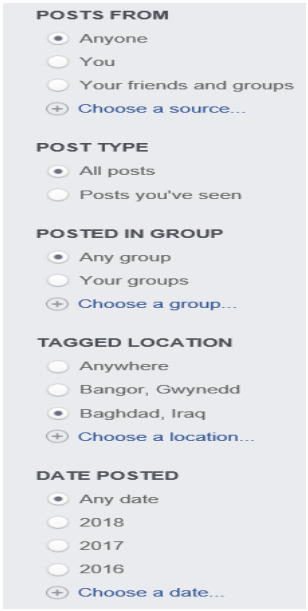


Figure 5.5: Facebook's filter for searching for more specific posts.

We used seed words defined by Aman and Szpakowicz (2007) in the search bar of Facebook for searching for a specific emotion and its synonyms (Aman and Szpakowicz, 2007). We used a manual collection of data rather than automatic collection of data due to noise in the data such as links and images. As stated, we used the query "feeling happy" in the Facebook search bar when searching for happy posts and the same for synonyms of happy, and repeated this for the other emotions.

Table 5.1 provides all the synonyms for each of Ekman's emotions used to collect posts for the dataset. These seed words were defined by Aman and Szpakowicz (2007) and we used these words in collecting the posts for the new Arabic emotion dataset (Aman and Szpakowicz, 2007).

Second, we used the following options in the search filter: we chose the option "anyone" from the filter "POSTS FROM"; the option "All posts" from the filter "POST TYPE"; the option "Any Group" from the filter "POSTED IN GROUP"; the option "Baghdad-Iraq" from the filter "TAGGED LOCATION"; and finally, the option "Any date" from the filter "DATE POSTED".

Table 5.1: Seed words used to collect Facebook posts for the IAEC dataset.

No.	Emotion	Synonyms
1	Anger	Angry, anger, annoyed, enraged, boiling, furious, mad, inflamed.
2	Disgust	Disgusted, sucks, sickening, stupid, unpleasant, contempt, nauseating.
3	Fear	Far, afraid, scared, frightened, insecure, nervous, horrified, panicked.
4	Happiness	Happy, awesome, amused, excited, great, pleased, amazing, cheerful.
5	Sadness	Sad, glooming, sorrowful, down, depressed, lonely, painful, guilty.
6	Surprise	Surprised, confused, astonished, sudden, unexpected, shocked, perplexed.

The reason for choosing the option "Baghdad-Iraq" for the filter "TAGGED LOCATION" is the lack of Arabic corpora specifically for the Iraqi dialect. There are a number of available corpora for Modern Standard Arabic language or for regional Arabic dialects such as the Egyptian and Levante dialects, or for other regional dialects but there are few available for the Iraqi dialect (Basili and Pazienza, 1997). Another reason for choosing "Baghdad-Iraq" as the "TAGGED LOCATION" is to focus on a specific Arabic dialect in order to see how difficult it is to recognise emotions for this dialect since research has shown that processing regional Arabic dialects can be significantly more problematic than processing Modern Standard Arabic, for example (Basili and Pazienza, 1997). Variations between dialects can also pose problems—for example, some words mean one emotion in a certain Arabic dialect, but they can mean a very different emotion in another Arabic dialect. For example, the word نحبك in the Syrian dialect means "love you", but in the Iraqi dialect means the opposite "do not love you". Another example, in the Gulfian dialect is the idiom جبت العيد which means "you disappoint me" but in the Iraqi dialect means "you bring joy and happiness". Analysing emotion variations between Arabic regional dialects is therefore outside the

scope of this research.

5.3 Description of the new Arabic Emotion Dataset

The text in the new Arabic emotion dataset consists of 1365 posts from Facebook. The posts were collected manually as mentioned in the previous section. Table 5.2 shows samples of these posts along with their English translations. We collected the data from December 2016 to August 2018.

Table 5.2: Samples of Facebook posts in the new Arabic dataset.

No.	Emotion	Post
1	Anger	طلعت روحي ابخسارت البرشا حرامات هيج ايفوز الريال والبرشه تطلع بره I nearly died when Barcelona lost the match, so Real Madrid won and Barcelona out.
2	Disgust	هذولة هم اكو امل يسوون شي ايجابي بالمستقبل Is there any hope they will do something positive in the future.
3	Fear	سترك ياربي شنو هاي صواريخ متت خوف بس ابجي همزين بابا بمنه اذني كامت توجعني God save us, what are these rockets, I feared until death, I'm just crying, thank God my father is here, My ears are hurting me.
4	Happiness	الف الف مبروك الله يسعدك و يهنئك بحياتك الزوجية Thousands of congratulations, God bless you in your marital life.
5	Sadness	الف رحمه ونور ع روحك الطاهرة يا بطل A mercy and light upon your pure spirit, you are such a hero.
6	Surprise	احلى مفاجأه من نور عيوني وعمري حمندشي ربي يحفظك الي The sweetest surprise from the light of my eyes and my life Hammandashi, God saves you to me.

Details of the IAEC dataset are described in Table 5.3. The dataset consists of six sub-datasets. Each dataset consists of posts belonging to one

of Ekman's emotions, i.e each sub-dataset represents one class. The table shows that the *Anger* class has the highest number of posts (309) with the fewest number of posts in the *Fear* (148) and *Disgust* (185) classes. This compares with the *Happiness*, *Sadness* and *Surprise* classes which have over 200 posts each. The total number of posts is 1,365, consisting of 22,438 words, and 286,775 characters.

Table 5.3: Number of posts, words, and characters in the IAEC dataset.

No.	**Emotion**	**#Posts**	**#Words**	**#Chars**
1	Anger	309	6,960	71,028
2	Disgust	185	2,936	29,967
3	Fear	148	1,596	16,843
4	Happiness	256	2,514	27,886
5	Sadness	238	3,486	35,759
6	Surprise	229	4,946	52,533

We had some issues with collecting the data. Most Iraqi people were not including their feeling in their posts before 2013. After 2014, Iraqi people started using the feeling option on Facebook to state their emotional state in their posts. Even so, many users of Facebook in Iraq still do not use the feeling option to express their emotional state at the same time that they write their posts. This is due to them not being aware of this feature provided by Facebook, or they do not use any of the more advanced Facebook features at all. These issues led to a lack of suitable posts while we were collecting data for the dataset, and why it took longer than anticipated (about 18 months) to finish collecting the posts for the IAEC dataset.

5.4 Dataset Evaluation

The goal of this comparison was to compare the emotion annotation between four annotators. The comparison was accomplished by measuring the inter-annotation agreement (Passonneau, 2006) among the four annotators. This measurement supports a valuable insight into the dataset usability and

understandability. Four annotators (A, B, C, D) participated in the annotation process of the IAEC dataset. Table 5.4 displays more details about the annotators who participated in the evaluation of IAEC. The tested Facebook posts that were delivered to the annotators were without labels. The annotator was free to choose labels from the six Ekman's emotions for each Facebook post.

Table 5.4: Annotator details who participated in the IAEC annotation process.

	A	**B**	**C**	**D**
Nationality	Iraqi	Iraqi	Iraqi	Iraqi
Qualifications	PhD. degree in Genetic Engineering and Bio-technology	MSc. degree in Science of Mathematics	MSc. degree in Electronics and Communication Engineering	MSc. degree in Mechanical Engineering
Experience in annotation	No previous experience	No previous experience	No previous experience	No previous experience

One of the goals of this evaluation was to find out to what extent an untrained user could understand and use these posts in the dataset. It is known that variation in skills and the interest of the annotators, and the ambiguity in the annotation guidelines leads to disagreement among annotators (Passonneau, 2006). The posts were written in Iraqi slang and many of these posts used idioms to express their emotions, unlike the corpora written in Classical Arabic or Modern Standard Arabic.

It is hard to identify single words that declare or express a specific emotion. So we asked the annotators to evaluate each entire post as belonging to one of Ekman's emotions.

As one of the annotators (D) disagreed with the others, we discarded this and used the annotations for the other three annotators. We used pairwise agreement to measure agreement among the three remaining annotators, i.e. between A $\leftrightarrow$ B, A $\leftrightarrow$ C, B $\leftrightarrow$ C, for each emotion in Ekman's emotions and we used the same pairwise analysis to evaluate the agreement for the whole dataset. Cohen's kappa co-efficient was used to calculate the pairwise

agreement between annotators (Cohen, 1960). Commonly, the kappa co-efficient is used to calculate agreement between two annotators. Table 5.5 shows the pairwise kappa co-efficient between each pair of annotators for each emotion. The inter-annotator agreement value between A $\leftrightarrow$ B was lowest for emotion *Surprise* with 0.721 and the highest value was for the *Fear* emotion with 0.785. On the other hand, the inter-agreement value between A $\leftrightarrow$ C was lowest for the *Fear* emotion with value 0.706, but highest for the *Sadness* emotion with value 0.929. Finally, the inter-agreement value between B $\leftrightarrow$ C was lowest for the *Sadness* emotion with 0.417; however, it was highest for the *Fear* emotion with 1.000.

Table 5.5: Kappa co-efficients for pairwise agreement among annotators per emotion.

Pair	**Anger**	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**
A $\leftrightarrow$ B	0.759	0.739	0.785	0.746	0.741	0.721
A $\leftrightarrow$ C	0.840	0.826	0.706	0.796	0.929	0.710
B $\leftrightarrow$ C	0.827	0.900	1.000	0.829	0.417	0.768

Table 5.6 displays the total inter-annotator agreement values among the three annotators, with the lowest agreement between A $\leftrightarrow$ C (with a value of 0.728), and the best agreement between A $\leftrightarrow$ C (0.825). The table also shows the average inter-annotator agreement for the three annotators with an overall average value of 0.768 which is a substantial agreement (Ku et al., 2007).

Table 5.6: Pairwise agreement amongst annotators.

	A $\leftrightarrow$ B	**A $\leftrightarrow$ C**	**B $\leftrightarrow$ C**	**Average**
Kappa	0.749	0.825	0.728	0.768

5.5 Experimental Results

In this section, various experiments were applied to classify Ekman's emotions from the Arabic text of IAEC. The next two subsections report the results of applying different Weka classifiers (Hall et al., 2009) and the compression-based PPM classifier (see previous chapter 3) to the Facebook posts in IAEC.

5.5.1 Applying Weka classifiers

We applied various classifiers supported by Weka (Hall et al., 2009) to find out the best classifier for Ekman's emotions. We used ten-fold cross-validation in our experiment. Table 5.7 lists results for five classifiers using Weka's `StringToWordVector` filter with `NGramTokenizer` to extract ngrams as features from the text, setting `NGramMaxSize` to 3, and `NGramMinSize` to 1. In this section, we investigated the use of both the Naïve Bayes and multinomial Naïve Bayes classifiers. Multinomial Naïve Bayes is an instance of Naïve Bayes that uses a multinomial distribution for each feature (Russell and Norvig, 2016).

The worst text classifier performances were for ZeroR and Naïve Bayes Multinomial Text, with both achieving the same results with 74.2% accuracy, 0.04 precision, 0.17 recall, and 0.06 F-measure.

Table 5.7: Classification results using five classifiers supported by Weka.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
J48	75.7	0.44	0.22	0.29
ZeroR	74.2	0.04	0.17	0.06
Naïve Bayes	75.9	0.49	0.23	0.31
Multinomial Naïve Bayes Text	74.2	0.04	0.17	0.06
SMO	75.9	0.49	0.22	0.31

Table 5.8 reports the confusion matrix for both ZeroR and Naïve Bayes Multinomial Text classifiers. (The numbers shown in bold font across the

leading diagonal represent the number of correct classifications). The table shows that the classifiers simply labelled every post in the Anger class.

Table 5.8: Confusion matrix of Ekman's emotions classification for the IAEC dataset using ZeroR and Naïve Bayes Multinomial text classifiers.

	Anger	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**
Anger	**309**	0	0	0	0	0
Disgust	185	**0**	0	0	0	0
Fear	148	0	**0**	0	0	0
Happiness	256	0	0	**0**	0	0
Sadness	237	0	0	0	**0**	0
Surprise	229	0	0	0	0	**0**

On the other hand, both the SMO and Naïve Bayes classifiers achieved better results than the previous classifiers. Naïve Bayes was slightly better than SMO although they both achieved the same accuracy of 75.9%, with both Naïve Bayes and SMO achieving 0.49 precision, Naïve Bayes achieving 0.23 recall while SMO achieved 0.22, and both Naïve Bayes and SMO achieving a 0.31 F-measure value.

As shown in Table 5.9, there is still substantial number of misclassifications with most posts being classified in the Anger class. The SMO classifier has 301 correctly classified posts and 8 misclassified posts in the Anger class. In contrast, all posts were misclassified in the Disgust class. There were 24 correctly classified posts and 124 misclassified posts in the Fear class, 12 correctly classified posts and 244 misclassified posts in the Happiness class, 36 correctly classified posts and 201 misclassified posts in the Sadness class, and 3 correctly classified posts and 226 misclassified posts in the Surprise class.

Table 5.10 reports the confusion matrix of applying the Naïve Bayes classifier to classify Ekman's emotions in IAEC. The only difference between the SMO and the Naïve Bayes confusion matrices is the Naïve Bayes has 38 correctly classified posts for the Sadness class while SMO has 36 for the same class. This results in a slightly better recall value of 0.23 for Naïve Bayes compared with 0.22 for SMO.

Table 5.9: Confusion matrix of Ekman's emotions classification for the IAEC dataset using the SMO classifier.

	Anger	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**
Anger	**301**	1	1	4	0	2
Disgust	181	**0**	0	3	1	0
Fear	119	0	**24**	3	2	0
Happiness	238	3	0	**12**	2	1
Sadness	197	0	2	2	**36**	0
Surprise	223	0	0	3	0	**3**

Table 5.10: Confusion matrix of Ekman's emotions classification for the IAEC dataset using the Naïve Bayse classifier.

	Anger	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**
Anger	**301**	1	1	4	0	2
Disgust	181	**0**	0	3	1	0
Fear	119	0	**24**	3	2	0
Happiness	238	3	0	**12**	2	1
Sadness	197	0	0	2	**38**	0
Surprise	226	0	0	1	0	**2**

The second experiment involved using the same Weka classifiers with the same `StringToWordVector` filter. This time the `CharacterNGramTokenizer` filter is used in order to find out whether this affects the classifiers' performance (and so that a comparison can be made to the PPM classifier which is a character-based method rather than a word-based one). The results for this experiment showed that all classifiers achieved the same results as in the previous experiment for all measures.

5.5.2 Applying the PPM classifier

In this experiment, a PPM classifier (see previous chapter 3) is used to classify Ekman's emotions in IAEC. As far as we know, only Amer and Teahan

(2017, 2018) have used the PPM classifier to classify emotions for English text (Almahdawi and Teahan, 2017; Almahdawi and Teahan, 2018) as stated in chapter 3. The results for the PPM classifier are shown in Table 5.11. The table reports that the order 5 PPM classifier (PPMD5) has significantly outperformed all the other classifiers used in the previous two experiments in all measures. Table 5.11 compares the PPM classifier result with the previous classifier results.

Table 5.11: Classification results using PPM classifier compared to classifiers supported by Weka.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
J48	74.4	0.44	0.22	0.29
ZeroR	74.2	0.04	0.17	0.06
NaïveBayes	75.9	0.49	0.23	0.31
Multinomial Naïve Bayes Text	74.2	0.04	0.17	0.06
SMO	75.9	0.49	0.22	0.31
PPMD5	86.9	0.63	0.59	0.61

Analysing the confusion matrix shown in Table 5.12 for the PPMD5 classification provides an insight into why the PPM classifier outperformed the other classifiers. Here there is much less confusion concerning which posts should be in the Anger class.

As stated, an order five PPM classifier (PPMD5) was used in this experiment. To find out which order of PPM model is most suitable for this type of classification for the Arabic language, a further experiment was used to check whether other orders (from order 2 up to order 12) give better results. Table 5.13 reports the results.

Table 5.13 shows that PPMD7 and PPMD9 achieved the best performance at classifying Ekman's emotions for the IAEC and achieved higher results in terms of accuracy, precision, recall and F-measure.

Table 5.12: Confusion matrix of Ekman's emotions classification for the IAEC dataset using the PPMD5 classifier.

	Anger	**Disgust**	**Fear**	**Happiness**	**Sadness**	**Surprise**
Anger	**241**	12	13	8	16	29
Disgust	64	**76**	5	5	9	26
Fear	30	6	**81**	7	9	15
Happiness	15	8	2	**165**	13	53
Sadness	40	10	13	8	**157**	10
Surprise	67	14	8	19	7	**114**

Table 5.13: Classification results of Ekman's emotions for the IAEC dataset using different orders of the PPM classifier.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
PPMD2	85.2	0.56	0.54	0.55
PPMD3	86.0	0.60	0.56	0.58
PPMD4	86.8	**0.63**	0.58	**0.61**
PPMD5	86.9	**0.63**	**0.59**	**0.61**
PPMD6	86.9	**0.63**	**0.59**	**0.61**
PPMD7	**87.1**	**0.63**	**0.59**	**0.61**
PPMD8	87.0	**0.63**	**0.59**	**0.61**
PPMD9	**87.1**	**0.63**	**0.59**	**0.61**
PPMD10	86.9	0.62	**0.59**	**0.61**
PPMD12	86.9	0.62	**0.59**	0.60

5.5.3 Applying the ID1 Classifier

In this section, the ID1 classifier is applied to classify Ekman's emotions in IAEC. One issue with this classifier was in processing the Arabic text, this is because some text was written by users' phones and they sometimes downloaded new font styles from the internet and the classifier was not able to deal with all these supplementary characters of new fonts. To overcome this issue we used the Buckwalter transliteration system (Buckwalter, 1990) to convert Arabic text into an equivalent text that uses a Latin alphabet.

However, even Buckwalter transliteration cannot deal with some of the fonts for two blogs that were collected. So these two blogs were not included in the dataset.

5.5.4 Example outputs produced by the ID1 classifier

This section provides some example outputs produced by the ID1 classifier in order to illustrate important aspects of how it works for Arabic text.

Tables 5.14 and Table 1 to Table 3 in the appendix list the top 30 unigrams for the ID1 classification, for the *Sadness*, *Happiness*, *Fear* and *Disgust* classes for IAEC's dataset. These were produced using training data taken from one of the folds during the ten-fold cross-validation process. The unigrams in the tables (shown in the second column) are arranged in descending order according to the codelength difference values (shown in the last column). The unigrams of Table 5.14 that reflect the particular emotion which is *Sadness* are unigrams ranked 1, 2, 3, 6, 7, 8, 11 and 12.

The unigram رحمه ranked number 1 in Table 5.14 appears 12 times in the training data of *Sadness* emotion, while it appears only one time in the training data of the *Sadness* complement. The unigram رحمه means "mercy" in English. The unigram رحمة ranked number 2 in the Table 5.14 looks very similar to the first unigram but it differs from the first unigram in the last character ة while the first unigram ends with ه. The meaning of the second unigram is also "mercy" in English. The reason for that difference is that people used the Iraqi dialect in their posts, and sometimes the person would write some words in his/her post wrongly due to his/her level of education.

Table 1 in the appendix lists the top 30 unigrams that were produced for one of the folds during the ten fold cross-validation. The unigram مبروك which means "congratulation" appears 11 times in the training data of the *Happiness* emotion, while it appears only once in the training data for the *Happiness* complement. The unigrams in the Table 1 that are ranked 1, 3, 4, 6 to 10, 15, 18, 19 and 26 all reflect the *Happiness* emotion and each one of these unigrams appear significantly more times in the *Happiness* training data than in the training data of the *Happiness* complement.

Table 5.14: The codelength differences unigrams of the ID1 classification in the IAEC's dataset for the Sadness class of one of the folds.

Rk.	Unigram (g)	$c(T_S, g)$	$c(T_{\overline{S}}, g)$	$H(T_S, g)$	$H(T_{\overline{S}}, g)$	$d(T_S, T_{\overline{S}}, g)$
1	رحمه	12	1	7.637	13.182	5.546
2	رحمة	10	1	7.900	13.182	5.283
3	اللهمّ	10	1	7.900	13.182	5.283
4	اليه	9	1	8.052	13.182	5.131
5	البطل	9	1	8.052	13.182	5.131
6	روحك	8	1	8.222	13.182	4.961
7	يرحمك	13	2	7.521	12.182	4.661
8	يرحمهم	5	1	8.900	13.182	4.283
9	لنا	4	1	9.222	13.182	3.961
10	عند	4	1	9.222	13.182	3.961
11	روح	4	1	9.222	13.182	3.961
12	رحم	4	1	9.222	13.182	3.961
13	العين	4	1	9.222	13.182	3.961
14	وفي	3	1	9.637	13.182	3.546
15	والدي	6	2	8.637	12.182	3.546
16	الدكتور	3	1	9.637	13.182	3.546
17	ابد	3	1	9.637	13.182	3.546
18	إليه	3	1	9.637	13.182	3.546
19	انا	7	3	8.414	11.597	3.183
20	وياك	2	1	10.222	13.182	2.961
21	وهو	2	1	10.222	13.182	2.961
22	نجم	2	1	10.222	13.182	2.961
23	مهدي	2	1	10.222	13.182	2.961
24	ليلة	2	1	10.222	13.182	2.961
25	كنت	2	1	10.222	13.182	2.961
26	قوة	2	1	10.222	13.182	2.961
27	قلوبنا	2	1	10.222	13.182	2.961
28	غالي	2	1	10.222	13.182	2.961
29	عنه	2	1	10.222	13.182	2.961
30	ظهر	2	1	10.222	13.182	2.961

5.5.5 Confusion matrix results for each classifier

The results from compiling the confusion matrices for the IAEC dataset for the ID1 classifier are discussed in this section. In order to gain insight into and illustrate how well the ID1 classifier performs, and provide a more detailed analysis which reports important aspects of the classification not revealed by the overall results listed below. As stated, the Arabic text of IAEC was converted prior to classification using the Buckwalter translitration. In order to illustrate how this works, ID1 classifier was applied in IAEC to classify Ekman's emotions. The following blog was written in the Iraqi Dialect "اجمل الاوقات مع احلى الاصدقاء" which means in English "The most beautiful times with my sweetest friends". The equivalent text for this blog after using the Buckwalter translitration is "Ajml AlAwqAt mE AHlY AlASdqA".

Tables 5.15 to Table 5.19 list the confusion matrices that were produced for the IAEC dataset with different thresholds (θ) for the ID1 classifier.

Table 5.15: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = -2$.

	Threshold $\theta = -2$						
	A	D	F	H	$Sd.$	$Su.$	U
A	**153**	4	2	0	34	116	0
D	111	**2**	0	0	18	54	0
F	57	2	**3**	2	30	54	0
H	37	2	1	**11**	48	156	0
$Sd.$	77	0	0	1	**112**	47	0
$Su.$	97	2	0	3	19	**104**	0

Each row of the matrix provides the number of instances in the actual (ground-truth) classes that were assigned the predicted class shown in the respective columns. These are labelled as follows: A for the *Anger* class; D for the *Disgust* class; F for the $Fear$ class; H for the $Happiness$ class; $Sd.$ for the *Sadness* class; and $Su.$ for the *Surprise* class. The column labelled U lists the number of unclassified texts for each class (i.e. the classifier returned the $Unclassified$ class).

The numbers in the diagonal written in bold font represent the correctly classified texts for each class. The numbers off-diagonal represent the mis-

Table 5.16: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = -1$.

	Threshold $\theta = -1$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**155**	5	2	1	35	111	0
D	108	**4**	1	0	16	56	0
F	51	5	**4**	2	28	58	0
H	30	3	1	**12**	46	164	0
Sd.	54	1	0	0	**132**	50	0
Su.	83	10	2	3	20	**108**	0

Table 5.17: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = 0$.

	Threshold $\theta = 0$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**115**	13	5	13	23	140	0
D	60	**18**	2	6	18	80	0
F	26	7	**16**	5	25	69	0
H	12	6	2	**47**	29	159	0
Sd.	24	3	3	7	**134**	66	0
Su.	56	9	1	14	18	**128**	0

Table 5.18: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = 1$.

	Threshold $\theta = 1$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**68**	48	26	16	92	47	1
D	37	**48**	21	13	23	42	1
F	13	12	**58**	10	30	25	0
H	8	7	22	**110**	47	62	0
Sd.	15	14	22	20	**148**	17	1
Su.	35	23	17	47	41	**63**	0

classified texts.

We can see a number of noticeable results for the Information divergence classifier (ID1) on the IAEC dataset. For thresholds $\theta = -2$, $\theta = -1$ and $\theta = 0$, there are zero unclassified blogs as listed in the column labelled U. The largest number of unclassifieds was when ID1 used threshold $\theta = 2$ with

Table 5.19: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = 2$.

	Threshold $\theta = 2$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**54**	44	79	35	78	15	4
D	29	**24**	47	25	46	11	2
F	14	15	**82**	16	17	4	0
H	10	11	42	**125**	42	23	3
Sd.	24	10	34	15	**151**	2	1
Su.	27	26	50	51	46	**26**	0

10 unclassifieds. The largest number of true positives along the diagonal was for the threshold $\theta = 1$. There are significant mis-classifications off-diagonal, with a large number of false positives for the *Surprise* when the thresholds $\theta = -2$, $\theta = -1$, $\theta = 0$ and $\theta = 1$ (Table 5.16, Table 5.15, Table5.17, and 5.18). The number of false positives increases for the *Fear* class when ID1 uses the threshold $\theta = 2$ (Table 5.19).

There were a large number of false negatives in the *Happiness* class when using thresholds $\theta = -2$, $\theta = -1$ and $\theta = 0$ and this noticeably decreased until reaching the threshold $\theta = 0$. The *Anger* class has noticeably the largest number of false negatives when using thresholds $\theta = 1$ which increased when threshold $\theta = 2$ was used.

5.5.6 Overall Results

The overall results of the experiments with the ID1 classifier are shown in Table 5.20. This lists the threshold θ in the first column, and then in the remaining columns the Accuracy, Precision, Recall and F-measure along with the number of texts that were assigned the *Unclassified* class. The highest column value for each dataset is shown in bold font.

Overall, several noticeable trends in the results can be recognised from Table 5.20. The number of unclassifieds for the IAEC dataset significantly decreases as the threshold θ gets smaller, and reachs zero when $\theta = 0.5$ and the reduction in the value of θ has no further effect.

Accuracy peaks for IAEC's dataset when $\theta = 1$. The best values for

Table 5.20: Ekman's emotions classification for the IAEC datasets using the new ID1 classifier.

Threshold θ	**Accuracy**	**Precision**	**Recall**	**F-measure**	**Unclass.**
2.0	78.1	0.33	0.35	0.34	10
1.5	78.7	0.35	**0.37**	0.36	9
1.0	**78.8**	0.36	**0.37**	**0.37**	3
0.5	78.3	0.38	0.35	0.36	0
0.0	77.9	0.42	0.32	0.36	0
-0.5	77.8	**0.46**	0.31	**0.37**	0
-1.0	76.8	0.37	0.27	0.31	0
-1.5	75.8	0.37	0.25	0.30	0
-2.0	76.1	0.37	0.25	0.30	0

precision, recall and F-measure are highest when θ is higher (-0.5). The highest value of accuracy, recall and F-measure occur when threshold $\theta = 1$. However, this results in three unclassified blogs.

The highest value of accuracy occur for threshold $\theta = 1$, the highest value of precision for threshold $\theta = -0.5$, for recall when threshold $\theta = 1.5$ and $\theta = 1$, and finally F-measure when threshold $\theta = 1$ and $\theta = -0.5$ (Table 5.21). Although, the best result was found for $\theta = -0.5$, choosing the threshold $\theta = 0$ that was found to be effective for the other experiments, the result works out to be 0.36 for F-measure.

Table 5.21: Confusion matrices for Ekman's emotions classification for IAEC's Dataset using the ID1 classifier for thresholds $\theta = -0.5$.

	Threshold $\theta = -0.5$						
	A	*D*	*F*	*H*	*Sd.*	*Su.*	U
A	**137**	6	3	2	16	144	0
D	81	**20**	1	1	11	71	0
F	35	8	**11**	4	17	73	0
H	23	9	1	**26**	24	171	0
Sd.	48	0	0	2	**123**	64	0
Su.	53	18	2	4	11	**137**	0

In summary, setting the threshold θ to the value (-0.5) provides a good trade-off between accuracy, precision, recall, and F-measure, while at the same time produces zero unclassifieds.

5.5.7 Comparison with other previous results

Table 5.22 shows a comparison between the results for ID1 when using threshold $\theta = -0.5$ and the previous listed in section 5.5.2 above for the IAEC dataset. The table lists the classifier in column 1, and the classification results for Ekman's emotions in terms of accuracy, precision, recall and F-measure in the remaining columns.

Table 5.22: Classification results using PPM classifier compared to classifiers supported by Weka.

Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
J48	74.4	0.44	0.22	0.29
ZeroR	74.2	0.04	0.17	0.06
NaïveBayes	75.9	0.49	0.23	0.31
Multinomial Naïve Bayes Text	74.2	0.04	0.17	0.06
SMO	75.9	0.49	0.22	0.31
PPMD5	86.9	0.63	0.59	0.61
ID1	77.8	0.46	0.31	0.37

The results show that the ID1 classifier is competitive with the other Weka classifiers, performing significantly better in all measures for IAEC's dataset. The PPM classifier significantly outperforms both the feature-based machine learning classifiers and information divergence ID1 classifier in recognising Ekman's emotions in Arabic text (IAEC dataset).

5.6 Summary and Conclusion

We created an emotion corpus consisting of 1365 Facebook posts annotated according to Ekman's emotions called IAEC. We made use if the IAEC dataset in three experiments. The first experiment was to test five classifiers supported by Weka data analytic tool to classify Ekman's emotions from the IAEC corpus. The best performance was achieved by the Naïve Bayes classifier and SMO classifier with 75.9% accuracy, but recall was a

slightly better for the Naïve Bayes classifier with 0.22.

In the second experiment, we used the PPMD5 classifier to classify blogs using Ekman's emotions in the IAEC. Surprisingly, this classifier significantly outperformed all the other classifiers in the first experiment with 86.9% accuracy, 0.63 precision, 0.59 recall, and 0.61 F-measure. Later, further experiments using PPM with different orders found that PPMD7 and PPMD9 achieved higher accuracy than other orders with a value of 87.1%.

In the third experiment, we used the new ID1 classifier that was defined in the previous chapter to classify blogs using Ekman's emotions in the IAEC. The BuckWalter translitration system was used to convert the Arabic text of IAEC prior to classification. The result was better than the first experiment results but was worse than the result of the second experiment, with 77.8% accuracy, 0.46 precision 0.31 recall, and 0.37 F-measure.

After analysing all the experiments on the IAEC dataset, it is clear that the PPM classifier outperforms the other classifiers, and the ID1 classifier outperforms all the remaining classifiers except PPM. We found that Naïve Bayes and SMO classifiers are better than J48, Multinomial Naïve Bayes for text and ZeroR classifiers.

In the future work, we think it would be better to include more data to train the classifiers supported by Weka and the ID1 classifier help to improve their results.

Chapter 6

Conclusion

6.1 Introduction

This chapter discusses the work accomplished by this thesis, and it highlights the important results. It also reviews the research questions and the aim and objectives. The future work with recommendations are addressed at the end of this chapter.

6.2 Summary and conclusions

This thesis investigated the feasibility of using the compression-based classification method in the field of emotion recognition. A new method for automatically recognising emotions in text has also been proposed. Although, there has been wide spread research in the field of automatic emotion recognition in text, it is the first time that classification-based compression and information divergence have been used in the field of emotion recognition.

PPM has never been used before in the emotion recognition field. PPM achieved better results when it was applied to the problem of emotion recognition in English text compared to traditional classifiers such as Naïve Bayes, ZeroR, J48, Multinomial Naïve Bayes and SVM. Distinguishing between *Happiness vs Sadness*, PPM achieved better results than traditional feature based classifiers. For distinguishing between *emotional vs. non-emotional* texts in

Aman's dataset PPM achieved lower accuracy than the feature based classifiers but better precision, recall and F-measure. For Ekman's emotion classification, PPM achieved better results than previously published results for the traditional classifiers. As well, we found that punctuation in text could affect the result of classification, especially when punctuation in the text relates to the emotion such as occur with the Alm and LiveJournal's datasets. On the other hand, punctuation has a negative affect on the results of classification when the punctuation in the text does not relate so well to the emotion such as occurs with Aman's dataset.

For the Arabic Text, PPM has also achieved better results than the traditional classifiers for Ekman's emotion classification. PPM also outperforms other traditional classifiers in terms of accuracy, precision, recall and F-measure.

We also proposed a new classifier based on information divergence called ID. We found through the experiments in chapter 4 and chapter 5 that classification of Ekman's emotion using information divergence based on word unigrams (ID1) is better than using bigrams and trigrams due to the increase in the number of unclassified texts for the latter. We have also found that the ID1 result outperforms other traditional classifiers in Ekman's emotion classification for both English and Arabic. However, when we compared the result of the ID classifier with the PPM classifier, we found that for English text, PPM achieved better results for Ekman's emotion classification than the ID1 classifier in terms of accuracy, precision, recall and F-measure. For the LiveJournal's dataset, PPM achieved slightly better results in accuracy and precision; however, ID1 achieved better results in recall and F-measure and the performance of ID1 is better than PPM since ID1 achieved an F-measure of value 0.41 while PPM achieved an F-measure of value of 0.38. For Alm's dataset, ID1 achieved much better results than PPM in terms of accuracy, precision, recall and F-measure.

For Arabic text, PPM achieved better results in terms of accuracy, precision, recall and F-measure for Ekman's emotion classification for the IAEC dataset.

6.3 Review of Research Questions

All research questions of this thesis in section 1.3 have been addressed. The particular research questions from section 1.3 were as follows:

1. *Can new methods be developed that are more effective for emotion recognition than existing approaches?*

   As shown in the experiments of chapters 3, 4 and 5, the PPM and the ID1 classifiers have successfully been applied to the problem of emotion recognition in text and they outperform other traditional feature-based classifiers.

2. *Would the Prediction by Partial Matching (PPM) compression-based method perform better than other common methods for emotion recognition?*

   As shown in the experiments of chapters 3 and 5, PPM has successfully been applied to the problem of emotion recognition in text. The PPM compression-based method outperforms other traditional word-based classification methods (Naïve Bayes, J48, ZeroR and SMO) in Ekman's emotions classification.

   PPM was also successfully applied to the problem of recognition of the *emotional* vs $non-emotional$ text. The PPM classifier performs better than other traditional word-based classifiers (ZeroR, Naïve Bayes, J48 and SMO) in precision, recall and F-measure at this task. PPM has been successfully applied to the problem of recognition for the *Happiness* vs *Sadness* texts.

3. *What is the most effective PPM model for emotion recognition? For example, does the PPMD order models affect the results of the emotion classification?*

   For English emotions datasets as shown in the experiments of chapter 3, we have found PPMD4 was the better order to classify Ekman's emotions in Alm and LiveJournal's datasets, while, PPMD3 was the best order to classify emotions Aman's dataset.

For Arabic Emotion dataset (IAEC), we have found that PPMD7 was the best order to classify Ekman's emotions in the IAEC dataset. There is no specific order model to be recommended for PPM for the problem of emotion recognition in English text. This depends on the text although we have found that an order 5 PPM model (PPMD5) is competitive. For the IAEC, the PPMD7 model was found to be best for Arabic text.

4. *Can these methods also be applicable to a language non-related to English (Arabic) and how effective are these methods?*

   As shown in the experiments of chapter 5, PPM has been successfully applied to the Arabic language. As well, the PPM outperforms other classifiers at classifying Ekman's emotions in Arabic text such as Naïve Bayes, J48, ZeroR, Multinomial Naïve Bayes text, SMO and ID1.

6.4 Review of Aim and Objectives

The aim and objectives of this thesis outlined in section 1.4 have been achieved successfully as detailed below. New methods have been added and applied to the problem of Emotion recognition in different text. A novel word-based classifier has been developed and applied to the problem of emotion recognition in different texts successfully.

- *To apply the Prediction by Partial Matching (PPM) compression-based classification method to the problem of automatically recognising emotions in text.*

  We have successfully applied PPM to automatically recognising emotions in text. This is achieved in Chapter 3.

- *To evaluate and validate the PPM method using different standard datasets and compare with other results achieved by other traditional classifiers*

  We evaluated and validated the PPM method in chapters 3 and 5 by using the ten fold cross-validation experimental methodology. We also

compared the results achieved by PPM in Chapters 3 and 5 with other word-based traditional classifiers and the PPM outperforms traditional word-based classifiers.

- *To create a new Iraqi Arabic Emotion Recognition dataset (IAER).*

  We have designed and created a new Iraqi Arabic Emotion corpus (IAEC) annotated according to Ekman's emotions as detailed in Chapter 5.

- *To evaluate and validate the adapted PPM method using the IAEC, and compare the results of PPM with other traditional classifiers.*

  We have successfully evaluated and validated for PPM by using ten fold cross-validation to classify Ekman's emotions in the IAEC dataset. We have also compared the PPM results with other traditional classifiers in both word-based and character-based. We have found that PPM outperforms other classifiers as shown in Chapter 5.

- *To develop and design a further new method for automatically recognising emotions in text.*

  We have successfully developed and designed a further a new method for automatically recognising emotions in English and Arabic text based on information divergence as shown in Chapter 4.

6.5 Limitations

We have faced some limitations while we were preparing and running the experiments. These limitations are listed bellow:

- LiveJournal's dataset is a large unbalanced dataset. As a result, the *Happiness* class often dominated the other classes and most mis-classifications occurred was with this class. In addition, processing of this dataset would mean that experiments would require more than one day to complete. it would be interesting to investigate experimenting with a smaller, balanced dataset.

- The mis-use of emojis from some people affects the result of the classifiers as emojis are often used to draw shapes and other things in their blogs not related to the emotion. This was one of the reasons why we used plain text in our classification experiments.

- There was a lack of people in Iraq who used the feeling option while they posting their Facebook posts. As a results, creating the IAEC dataset took about 18 months to complete. This dataset would benefit by being increased in size but a lot more time would be required to collect the extra data.

- There is also the issue of unclassified items with the ID1 classifier. This is specially true for Aman's dataset due to punctuation and the ID1 classifier using only the common unigrams between the emotion and its complement.

6.6 Future Work

Based on this thesis, several questions have been raised and deserve further investigation. The issues can be listed as follows:

- We investigated character-based PPMD but we never investigate the word-based PPMD. This would be worth exploring in the future.

- As mentioned in the limitations section above, the ID1 classifier has the issue of unclassified items. Further investigation is needed to solve this problem in the future by somehow including unique unigrams in the classification or merging unique unigrams with the common unigrams which may yield improved classification results.

- The IAEC dataset needs to increase its size due to the lack of training data for each class. This will require a further 2 or 3 years effort.

- According to the available research on emotion recognition in text, most have used the traditional feature-based classifiers such as ZeroR, Naïve Bayes and SVM. Therefore, these classifiers were chosen in order

to compare PPM and ID1 results. There has been relatively few publications which have used neural networks and deep learning, for this problem, and future work needs to explore how well these methods will work to this problem.

References

Abdul-Mageed, M. and Diab, M. T. (2012). Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *Language Resources and Evaluation LREC*, pages 3907–3914, Istanbul, Turkey. European Language Resources Association (ELRA).

Abdulla, N. A., Ahmed, N. A., Shehab, M. A., and Al-Ayyoub, M. (2013). Arabic sentiment analysis: Lexicon-based and corpus-based. In *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2013*, pages 1–6, Amman - Jordan. IEEE.

Agrawal, A. and An, A. (2012). Unsupervised emotion detection from text using semantic and syntactic relations. In *The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 346–353, Macau SAR, China. IEEE Computer Society Washington, DC, USA.

Ahmed, F. and Nürnberger, A. (2009). Evaluation of n-gram conflation approaches for arabic text retrieval. *Journal of the American Society for Information Science and Technology*, 60(7):1448–1465.

Ahmed, F. and Nürnberger, A. (2011). A web statistics based conflation approach to improve arabic text retrieval. In *Federated Conference on Computer Science and Information Systems (FedCSIS), 2011*, pages 3–9, Szczecin, Poland. IEEE.

Al-Kazaz, N. R. and Teahan, W. J. (2016). An automatic cryptanalysis of

transposition ciphers using compression. In *International conference on cryptology and network security*, pages 36–52, Milan, Italy. Springer.

Al-Mahdawi, A. and Teahan, W. J. (2019). Emotion recognition for text using information divergence. *Journal of IEEE Transactions on Aective Computing*. Pending.

Al-Smadi, M., Qawasmeh, O., Talafha, B., and Quwaider, M. (2015). Human annotated arabic dataset of book reviews for aspect based sentiment analysis. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 726–730, Rome, Italy. IEEE.

Alamri, M. M. and Teahan, W. J. (2019). Automatic correction of arabic dyslexic text. *Computers*, 8(1). MDPI.

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *the conference on human language technology and empirical methods in natural language processing*, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Almahdawi, A. and Teahan, W. J. (2017). Emotion recognition in text using ppm. In *SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 149–155, Cambridge, UK. Springer.

Almahdawi, A. and Teahan, W. J. (2018). Automatically recognizing emotions in text using prediction by partial matching (ppm) text compression method. In *International Conference on New Trends in Information and Communications Technology Applications*, pages 269–283, Baghdad, Iraq. Springer.

Almahdawi, A. J. and Teahan, W. J. (2019). A new arabic dataset for emotion recognition. In *Intelligent Computing-Proceedings of the Computing Conference*, pages 200–216. Springer. London.

Altamimi, M. and Teahan, W. J. (2017). Gender and authorship categorisation of arabic text from twitter using ppm. *International Journal of Computer Science & Informa tion Technology (IJCSIT)*, Vol 9(No 2).

Aman, S. and Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205, Pilsen, Czech Republic. Springer.

Aman, S. and Szpakowicz, S. (2008). Using roget's thesaurus for fine-grained emotion recognition. In *the Third International Joint Conference on Natural Language Processing: Volume-I*, Hyderabad, India,. The Association for Computer Linguistics.

Arabic, M. E. (2015). What is spoken Arabic / the Arabic dialects? `http://www.myeasyarabic.com/site/what_is_spoken_arabic.htm`. Accessed: 4 Nov. 2018.

Arnold, M. B. (1960). Emotion and personality. Columbia University Press.

Azmi, A. M. and Alzanin, S. M. (2014). Aara'–a system for mining the polarity of saudi public opinion through e-newspaper comments. *Journal of Information Science*, 40(3):398–410. Sage Publications Sage UK: London, England.

Basili, R. and Pazienza, M. T. (1997). Lexical acquisition and information extraction. In Pazienza, M. T., editor, *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, pages 44–72, Berlin, Heidelberg. Springer Berlin Heidelberg.

Binali, H., Wu, C., and Potdar, V. (2010). Computational approaches for emotion detection in text. In *the IEEE international conference on digital ecosystems and technologies (DEST 2010)*, pages 172–177, Dubai, United Arab Emirates. IEEE.

Brown, P. F., Pietra, V. J. D., Mercer, R. L., Pietra, S. A. D., and Lai, J. C. (1992). An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40. MIT Press.

Bruna, O., Avetisyan, H., and Holub, J. (2016). Emotion models for textual emotion classification. In *Journal of Physics: Conference Series*, volume 772, page 012063. IOP Publishing.

Buckwalter, T. (1990). Arabic transliteration. `www.qamus.org/transliteration.htm`. Accessed 28, Jan. 2019.

BuiltWith (2009). Encoding usage distribution on the entire internet. `https://trends.builtwith.com/encoding/`. Accessed: 20, Aug. 2018.

Cabanac, M. (2002). What is emotion? *Behavioural processes*, 60(2):69–83. Elsevier.

Carter, M. G. (1998). The arabic language. *Bulletin of the School of Oriental and African Studies*, 61(3):550–551. Cambridge University Press.

Chaffar, S. and Inkpen, D. (2011). Using a heterogeneous dataset for emotion analysis in text. In *Canadian Conference on Artificial Intelligence*, pages 62–67, Newfoundland and Labrador, Canada. Springer.

Charniak, E. (1993). Statistical language learning. MIT Press, Cambridge, Massachusetts.

Cherif, W., Madani, A., and Kissi, M. (2015). New rules-based algorithm to improve arabic stemming accuracy. *International Journal of Knowledge Engineering and Data Mining*, 3(3-4):315–336. Inderscience Publishers.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1):51–89. Wiley Online Library.

Cleary, J. and Witten, I. (1984). Data compression using adaptive coding and partial string matching. *IEEE transactions on Communications*, 32(4):396–402. IEEE.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46. Sage Publications Sage CA: Thousand Oaks, CA.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *the 12th international conference on World Wide Web*, pages 519–528, Budapest, Hungary. ACM.

Davis, M. (2012). Unicode over 60 percent of the web. `https://googleblog.blogspot.com/2012/02/unicode-over-60-percent-of-web.html`. Accessed: 28, Aug. 2018.

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200. Taylor & Francis.

Ekman, P. (1999). Facial expressions. *Handbook of cognition and emotion*, vol. 16(no. 301):p.p e320. New York.

Ekman, P., Friesen, W. V., and Ellsworth, P. (1972). Emotion in the human face: Guidelines for research and a review of findings. *New York. Permagon*.

Elbeheri, G., Everatt, J., Reid, G., and Mannai, H. a. (2006). Dyslexia assessment in arabic. *Journal of Research in Special Educational Needs*, 6(3):143–152. Wiley Online Library.

elKaliouby, R. (2017). We need computers with empathy. `https://www.technologyreview.com/s/609071/we-need-computers-with-empathy`. Accessed: 15, Sep. 2018.

Ephal, I. (1982). *The Ancient Arabs: nomads on the borders of the Fertile Crescent, 9th-5th Centuries BC*. Brill. Jerusalem: MagnesPress.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181. JMLR. org.

Fink, C. R., Chou, D. S., Kopecky, J. J., and Llorens, A. J. (2011). Coarse- and fine-grained sentiment analysis of social media text. *Johns hopkins apl technical digest*, 30(1):22–30.

Frank, E., Chui, C., and Witten, I. H. (2000). Text categorization using compression models. University of Waikato, Department of Computer Science.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296.

Frijda, N. H. (1986). *The emotions*. Cambridge University Press.

Ghazi, D., Inkpen, D., and Szpakowicz, S. (2010). Hierarchical versus flat classification of emotions in text. In *the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 140–146, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gray, J. A. (1982). Précis of the neuropsychology of anxiety: An enquiry into the functions of the septo-hippocampal system. *Behavioral and Brain Sciences*, 5(3):469–484. Cambridge University Press.

Gupta, N., Gilbert, M., and Fabbrizio, G. D. (2013). Emotion detection in email customer care. *Computational Intelligence*, 29(3):489–505. Wiley Online Library.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The Weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18. ACM.

Hancock, J. T., Landrigan, C., and Silver, C. (2007). Expressing emotion in text-based communication. In *the SIGCHI conference on Human factors in computing systems*, pages 929–932, San Jose, CA, US. ACM.

Haralick, R. M. (1976). The table look-up rule. *Communications in Statistics-Theory and Methods*, 5(12):1163–1191.

Howard, P. G. (1993). The design and analysis of efficient lossless data compression systems. Technical report. Brown University.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, Seattle, WA, USA. ACM.

International, L. (2000). Codeset Overview. `http://www.langbox.com/codeset.html`. LangBox International. Accessed: 8, Sep. 2018.).

Izard, C. E. (1971). The face of emotion. New York, Appleton-Century-Crofts.

James, W. (1884). What is an emotion? *Mind*, 9(34):188–205. JSTOR.

Jelinek, F. (1990). Self-organized language modeling for speech recognition. *Readings in speech recognition*, pages 450–506.

Jindal, N. and Liu, B. (2008). Opinion spam and analysis. In *the 2008 international conference on web search and data mining*, pages 219–230, Stanford University, Stanford, CA 94305, USA. ACM.

Jo, Y. and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *the fourth ACM international conference on Web search and data mining*, pages 815–824, Hong Kong, China. ACM.

Karlgren, J. and Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. In *the 15th conference on Computational linguistics-Volume 2*, pages 1071–1075, Kyoto, Japan. Association for Computational Linguistics.

Kemper, T. D. (1987). How many emotions are there? wedding the social and the autonomic components. *American journal of Sociology*, 93(2):263–289. University of Chicago Press.

Khmelev, D. V. and Teahan, W. J. (2003). A repetition based measure for verification of text collections and for text categorization. In *the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 104–110, Toronto, ON, Canada. ACM.

Kim, S.-B., Han, K.-S., Rim, H.-C., and Myaeng, S. H. (2006). Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466. IEEE.

Knight, K. (1999). Mining online text. *Communications of the ACM*, 42(11):58–61. ACM.

Korting, T. S. (2006). C4. 5 algorithm and multivariate decision trees. *Image Processing Division, National Institute for Space Research–INPE Sao Jose dos Campos–SP, Brazil*.

Ku, L.-W., Lo, Y.-S., and Chen, H.-H. (2007). Test collection selection and gold standard generation for a multiply-annotated opinion corpus. In *the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 89–92, Prague, Czech Republic. Association for Computational Linguistics.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15, Chemnitz, Germany. Springer.

Li, H., Pang, N., Guo, S., and Wang, H. (2007). Research on textual emotion recognition incorporating personality factor. In *the IEEE International Conference on Robotics and Biomimetics, 2007. ROBIO 2007.*, pages 2222–2227, Sanya, China. IEEE.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM. Hong Kong, China.

Lin, W.-H., Wilson, T., Wiebe, J., and Hauptmann, A. (2006). Which side are you on?: identifying perspectives at the document and sentence levels. In *the tenth conference on computational natural language learning*, pages 109–116. Association for Computational Linguistics (ACL). New York, USA.

Lin, Y. (2002). Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6(3):259–275. Springer.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167. Morgan & Claypool Publishers.

Ma, C., Prendinger, H., and Ishizuka, M. (2005). Emotion estimation and reasoning based on affective textual interaction. In *International Conference on Affective Computing and Intelligent Interaction*, pages 622–628. Springer.

Macdonald, M. C. (2009). *Literacy and identity in pre-Islamic Arabia*. Ashgate.

Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. AAAI press.

McDougall, W. H. (1926). *This is the life!* AA Knopf.

Microsoft (2018). Code page 1256 windows arabic. `https://msdn.microsoft.com/en-us/library/cc195058.aspx`. Accessed 3, Sep. 2018).

Mihalcea, R. and Liu, H. (2006). A corpus-based approach to finding happiness. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 139–144. AAAI press, Menlo Park, California.

Minsky, M. (2007). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster.

Mishne, G. et al. (2005). Experiments with mood classification in blog posts. In *the ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, volume 19, pages 321–327. ACM.

Moffat, A. (1990). Implementing the ppm data compression scheme. *IEEE Transactions on communications*, 38(11):1917–1921. IEEE.

Mowrer, O. (1960). Learning theory and behavior. Hoboken, NJ, US: John Wiley & Sons Inc.

Mukherjee, A. and Liu, B. (2012). Aspect extraction through semi-supervised modeling. In *the 50th annual meeting of the association for computational linguistics: Long papers-volume 1*, pages 339–348, Jeju Island, Korea. Association for Computational Linguistics.

Munezero, M., Montero, C. S., Kakkonen, T., Sutinen, E., Mozgovoy, M., and Klyuev, V. (2014). Automatic detection of antisocial behaviour in texts. *Informatica*, 38(1). Slovenian Society Informatika.

Nadali, A., Kakhky, E. N., and Nosratabadi, H. E. (2011). Evaluating the success level of data mining projects based on crisp-dm methodology by a fuzzy expert system. In *the 3rd International Conference on Electronics Computer Technology (ICECT), 2011.*, volume 6, pages 161–165, Kanyakumari, India. IEEE.

Nagel, M. (1987). Kanal, ln, a. rosenfeld (eds.): Progress in pattern recognition 2. north holland, amsterdam 1985, 402 s., 35. $-$ $-$; $dfl.160.$ $-$ $-.isbn0444877231.Biometrical Journal, 29(6) : 702 - -702.$

Najeeb, M., Abdelkader, A., and Al-Zghoul, M. B. (2014). Arabic natural language processing laboratory serving islamic sciences. *International Journal of Advanced Computer Science and Applications*, 5(3):114–117.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2010). Emoheart: conveying emotions in second life based on affect sensing from text. *Advances in Human-Computer Interaction*, 2010:1. Hindawi Publishing Corp.

Oatley, K. and Johnson-Laird, P. N. (1987). Towards a cognitive theory of emotions. *Cognition and emotion*, 1(1):29–50. Taylor & Francis.

Olson, D. L. and Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.

Ortony, A. and Turner, T. J. (1990). What's basic about basic emotions? *Psychological review*, 97(3):315. American Psychological Association.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Panksepp, J. (1982). Toward a general psychobiological theory of emotions. *Behavioral and Brain sciences*, 5(3):407–422. Cambridge University Press.

Parrott, W. G. (2001). *Emotions in social psychology: Essential readings*. Psychology Press.

Passonneau, R. (2006). Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation. In *the 5th International Conference on Language Resources and Evaluation (LREC)*., GENOA - ITALY.

Picard, R. W. (1997). Affective computing. MIT press, Cambridge, MA, 1997.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion in r. plutchik & h. kellerman (eds.) emotion: Theory, research, and experience (vol. 1, pp. 189-217). New York: Academic" Press.

Plutchik, R. and Kellerman, H. (1980). Theories of emotion, vol. 1 of emotion: Theory, research, and experience. New York: Academic Press.

Popescu, A.-M., Nguyen, B., and Etzioni, O. (2005). Opine: Extracting product features and opinions from reviews. In *the HLT/EMNLP on interactive demonstrations Proceedings*, pages 32–33, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

Pratama, B. Y. and Sarno, R. (2015). Personality classification based on twitter text using naive bayes, knn and svm. In *the International Conference on Data*

and Software Engineering (ICoDSE), 2015, pages 170–174, Yogyakarta, DIY, Indonesia. IEEE.

Putten, M. v. (2017). The development of the triphthongs in quranic and classical Arabic. Leiden Center for the Study of Ancient Arabia (LeiCenSAA).

Qin, Y.-p. and Wang, X.-k. (2009). Study on multi-label text classification based on svm. In *the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09.*, volume 1, pages 300–304, Tianjin, China. IEEE.

Ramiandrisoa, F., Mothe, J., Benamara, F., and Moriceau, V. (2018). Irit at e-risk 2018. pages 367–377.

Rasheed, Z. T. (2008). Arabic is the tie that binds. `https://www.aljazeera.com/focus/arabunity/2008/01/2008525185325418882.html`. Accessed 16, Aug. 2018.

Refaee, E. and Rieser, V. (2014). An Arabic Twitter corpus for Subjectivity and Sentiment Analysis. In *the Ninth International Conference on Language Resources and Evaluation (LREC 14), European Language Resources Association (ELRA)*, pages 2268–2273. Reykjavik, Iceland.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.

Salem, F. (2017). Social media and the internet of things towards data-driven policymaking in the arab world: Potential, limits and concerns. Mohammed Bin Rashid Al Maktoum Global Initiatives, Dubai.

Schubert, L. (2015). Computational linguistics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2015 edition.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47. ACM.

Sebastiani, F. (2005). Text categorization text mining and its applications. *Text Mining and its Applications, A*, pages 109–129. WTP press, Southampton, UK. Forthcoming.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423. University of Illionois press, Urbana, IL.

Shelke, N. M. (2014). Approaches of emotion detection from text. *International Journal of Computer Science and Information Technology*, 2(2):123–128.

Simons, G. F. and Fennig, C. D. (2018). Ethnologue: Languages of the world, twenty-first edition. Dallas, Texas. `http://www.ethnologue.com`. Accessed: 12, Aug. 2018.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89.

Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text. In *the 2008 ACM symposium on Applied computing*, pages 1556–1560, New York. ACM.

Strapparava, C., Valitutti, A., et al. (2004). Wordnet affect: an affective extension of wordnet. In *Lrec*, volume 4, pages 1083–1086, Lisbon.

Tao, J. and Tan, T. (2005). Affective computing: A review. In Tao, J., Tan, T., and Picard, R. W., editors, *Affective Computing and Intelligent Interaction*, pages 981–995, Berlin, Heidelberg. Springer Berlin Heidelberg.

Teahan, W. J. (1998). *Modelling English text*. PhD thesis, University of Waikato.

Teahan, W. J. (2000). Text classification and segmentation using minimum cross-entropy. In *Content-Based Multimedia Information Access-Volume 2*, pages 943–961. Le centre de hautes etudes internationales d'informatique documentaire,.

Teahan, W. J. and Harper, D. J. (2001). Combining ppm models using a text mining approach. In *Data Compression Conference, 2001. Proceedings. DCC 2001.*, pages 153–162. IEEE.

Teahan, W. J. and Harper, D. J. (2003). Using compression-based language models for text categorization. In *Language modeling for information retrieval*, pages 141–165. Springer.

Tomkins, S. S. (1984). Affect theory. *Approaches to emotion*, 163(163–195). Hillsdale, NJ: Erlbaum.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Uszkoreit, H. (2000). What is computationallinguistics? techreport, Department of Computational Linguistics and Phonetics of Saarland University., Department of Computational Linguistics and Phonetics of Saarland University.

Voeffray, S. (2011). Emotion-sensitive human-computer interaction (hci): State of the art-seminar paper. *Emotion Recognition*, pages 1–4.

Wang, S., Chen, Z., and Liu, B. (2016). Mining aspect-specific opinion using a holistic lifelong topic model. In *The 25th International Conference on World Wide Web Conferences Steering Committee.*, pages 167–176. , Republic and Canton of Geneva, Switzerland.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). 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)*, pages 587–592. IEEE.

Watson, D. and Tellegen, A. (1985). Toward a consensual structure of mood. *Psychological bulletin*, 98(2):219. US: American Psychological Association.

Watson, J. B. (1930). Behaviorism. New York: Norton.

Weekley, E. (2012). *An etymological dictionary of modern English*, volume 2. Courier Corporation.

Weiner, B. and Graham, S. (1984). An attributional approach to emotional development. *Emotions, cognition, and behavior*, pages 167–191. New York: Cambridge University Press.

Wu, C.-H., Chuang, Z.-J., and Lin, Y.-C. (2006). Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, 5(2):165–183.

Yadollahi, A., Shahraki, A. G., and Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):25.

Yang, H., Willis, A., Roeck, A. D., and Nuseibeh, B. (2012). A hybrid model for automatic emotion recognition in suicide notes. *Biomedical Informatics Insights*, 5s1:BII.S8948.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM. Berkeley, US, 1999.

Yong-feng, S. and Yan-ping, Z. (2004). Comparison of text categorization algorithms. *Wuhan university Journal of natural sciences*, 9(5):798–804.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhang, Y., Li, Z., Ren, F., and Kuroiwa, S. (2005). Semi-automatic emotion recognition from textual input based on the constructed emotion thesaurus.

In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 571–576. IEEE. 576, Wuhan, China, October 2005.

.1 The common ngrams for each emotion in IAEC

Table 1: The codelength differences unigrams of the ID1 classification in the IAEC's dataset for the Happiness class of one of the folds.

Rk.	Unigram(g)	$c(T_H, g)$	$c(T_{\overline{H}}, g)$	$H(T_H, g)$	$H(T_{\overline{H}}, g)$	$d(T_S, T_{\overline{H}}, g)$
1	مبروك	11	1	7.751	13.185	5.435
2	كروب	6	1	8.625	13.185	4.560
3	اجمل	5	1	8.888	13.185	4.297
4	خير	13	3	7.510	11.600	4.091
5	اخويه	4	1	9.210	13.185	3.975
6	الخير	11	3	7.751	11.600	3.850
7	شاء	14	4	7.403	11.185	3.783
8	ولادة	3	1	9.625	13.185	3.560
9	احلا	3	1	9.625	13.185	3.560
10	الغالي	11	4	7.751	11.185	3.435
11	صباح	8	3	8.210	11.600	3.390
12	مساء	5	2	8.888	12.185	3.297
13	جميع	5	2	8.888	12.185	3.297
14	اخوتي	5	2	8.888	12.185	3.297
15	الف	29	13	6.352	9.485	3.133
16	يابه	2	1	10.210	13.185	2.975
17	وكل	6	3	8.625	11.600	2.975
18	واحلى	4	2	9.210	12.185	2.975
19	واحله	2	1	10.210	13.185	2.975
20	مكانكم	2	1	10.210	13.185	2.975
21	مصطفى	2	1	10.210	13.185	2.975
22	كرم	2	1	10.210	13.185	2.975
23	كادر	2	1	10.210	13.185	2.975
24	شكراً	4	2	9.210	12.185	2.975
25	شعبان	2	1	10.210	13.185	2.975
26	سعيدة	2	1	10.210	13.185	2.975
27	زيارة	2	1	10.210	13.185	2.975
28	سعيدة	2	1	10.210	13.185	2.975
29	زيارة	2	1	10.210	13.185	2.975
30	رزقني	2	1	10.210	13.185	2.975

Table 2: The codelength differences unigrams of the ID1 classification in the IAEC's dataset for the Fear class of one of the folds.

Rk.	Unigram(g)	$c(T_F, g)$	$c(T_{\overline{F}}, g)$	$H(T_F, g)$	$H(T_{\overline{F}}, g)$	$d(T_F, T_{\overline{F}}, g)$
1	نتائج	1	1	7.375	13.277	5.902
2	لكيتة	1	1	7.375	13.277	5.902
3	كولش	1	1	7.375	13.277	5.902
4	كدامكم	1	1	7.375	13.277	5.902
5	قريب	1	1	7.375	13.277	5.902
6	عنده	1	1	7.375	13.277	5.902
7	عمليه	1	1	7.375	13.277	5.902
8	طبيعي	1	1	7.375	13.277	5.902
9	صوت	1	1	7.375	13.277	5.902
10	صاير	1	1	7.375	13.277	5.902
11	رح	1	1	7.375	13.277	5.902
12	دعائكم	1	1	7.375	13.277	5.902
13	حال	1	1	7.375	13.277	5.902
14	الوضع	1	1	7.375	13.277	5.902
15	المستشفى	1	1	7.375	13.277	5.902
16	الكراده	1	1	7.375	13.277	5.902
17	السبت	1	1	7.375	13.277	5.902
18	الرحيم	1	1	7.375	13.277	5.902
19	الرحمن	1	1	7.375	13.277	5.902
20	الرجيم	1	1	7.375	13.277	5.902
21	انفجار	2	3	6.375	11.692	5.317
22	يم	1	2	7.375	12.277	4.902
23	قوي	1	2	7.375	12.277	4.902
24	عود	1	2	7.375	12.277	4.902
25	شنو	8	16	4.375	9.277	4.902
26	بالدنيا	1	2	7.375	12.277	4.902
27	اشو	1	2	7.375	12.277	4.902
28	بسم	1	3	7.375	11.692	4.317
29	بالله	1	3	7.375	11.692	4.317
30	اهل	1	3	7.375	11.692	4.317

Table 3: The codelength differences unigrams of the ID1 classification in the IAEC's dataset for the Disgust class of one of the folds.

Rk.	Unigram(g)	$c(T_D, g)$	$c(T_{\overline{D}}, g)$	$H(T_D, g)$	$H(T_{\overline{D}}, g)$	$d(T_S, T_{\overline{D}}, g)$
1	سبايكر	3	1	5.977	13.294	7.316
2	كافي	3	2	5.977	12.294	6.316
3	يشوف	1	1	7.562	13.294	5.732
4	ويطلع	1	1	7.562	13.294	5.732
5	ل	1	1	7.562	13.294	5.732
6	كامل	1	1	7.562	13.294	5.732
7	عمه	1	1	7.562	13.294	5.732
8	صفحة	1	1	7.562	13.294	5.732
9	شهيد	1	1	7.562	13.294	5.732
10	سيد	1	1	7.562	13.294	5.732
11	دعوة	2	2	6.562	12.294	5.732
12	تعالو	1	1	7.562	13.294	5.732
13	ترد	1	1	7.562	13.294	5.732
14	بحياتي	1	1	7.562	13.294	5.732
15	ب	1	1	7.562	13.294	5.732
16	اهالي	1	1	7.562	13.294	5.732
17	انتم	2	2	6.562	12.294	5.732
18	الزين	1	1	7.562	13.294	5.732
19	الحكومة	1	1	7.562	13.294	5.732
20	اسمه	1	1	7.562	13.294	5.732
21	يله	1	2	7.562	12.294	4.732
22	فقط	1	2	7.562	12.294	4.732
23	عنه	1	2	7.562	12.294	4.732
24	شهداء	2	4	6.562	11.294	4.732
25	حرام	1	2	7.562	12.294	4.732
26	جذب	1	2	7.562	12.294	4.732
27	اهل	1	2	7.562	12.294	4.732
28	فديت	1	3	7.562	11.709	4.147
29	خاص	1	3	7.562	11.709	4.147
30	بدون	1	3	7.562	11.709	4.147