Enhancing Word Representations for Emotional Intensity and Offensive Language Detection in Arabic Microblog Text

Abdullah I. Alharbi

A Thesis Submitted to
the University of Birmingham
for the Degree of
Doctor of Philosophy

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
May 2022

Abstract

Social media motivates people to express their emotions and share them publicly. However, at the same time, there are those who use it to spread racism and offensive language. Detecting emotional intensity and offensive language can be challenging in the context of social media microblogs, such as Twitter. This task becomes even more complicated when morphology-rich languages, such as Arabic, are involved. Social media communications typically consist of a range of dialects and sub-dialects that are not ruled by consistent standards. Therefore, there is a need to adopt effective methods and resources to better comprehend and treat a variety of linguistic forms when seeking to understand the emotional intensity and offensive language in Arabic short texts.

In this dissertation, we study two main problems: detection of emotional intensity and of offensive language in Arabic microblogs. First, we propose a novel combination of static character- and word-level embeddings (ACWE) to improve the detection of emotional intensity. For this purpose, we create word-and character-level embeddings using a large number of tweets enriched by the diversity of affective vocabulary words and Arabic dialects. ACWE significantly outperforms state-of-the-art pre-trained Arabic word embeddings in emotional intensity tasks. Second, we enhance contextualised language models by incorporating ACWE to identify emotional intensity. We show that our proposed method obtains state-of-the-art results in seven affect tasks, including our main task, emotional intensity detection. Lastly, we exploit emotional intensity and other affect-related tasks in the offensive language task using transfer learning approaches. We find that incorporating the best-performing contextual language models with anger intensity and emotion-related tasks enhances the performance of offensive language detection.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. Mark Lee. I would like to thank you for encouraging me throughout my research and allowing me to grow as a research scientist. Your advice on both my research and career has been invaluable.

I would also like to thank my thesis group members, Prof. David Parker and Dr. Phillip Smith, for dedicating their time as committee members during times of difficulty and hardship. As a result of your positivity and encouragement, I significantly benefited from your objective comments and suggestions.

I am very fortunate and grateful to King Abdulaziz University and the Saudi Arabia Cultural Bureau in London for facilitating and financing my studies at the University of Birmingham and for giving me the chance to gain experience from many scientific conferences and workshops.

Not least of all, I owe a lot to my loving parents, Ibrahim Alharbi and Norah Almdara, whose words of encouragement and push for tenacity have been a source of inspiration throughout this journey. A special thanks go to my close family members, my wife, Khawlah; daughter, Norah; and sons, Asim and Mohammed. Without their love and support throughout this journey, I could not have accomplished this success. Finally, I would like to acknowledge all the friends who supported me and motivated me during the long and challenging process of my PhD studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Every day, large volumes of opinion data are generated via multiple forums, personal blogs and social media platforms. One of the most popular social networking microblogs is Twitter, which allows online users to voice their views and share their emotions on a variety of topics and trending issues. On the one hand, this huge growth of user-generated content contributes to freedom of expression, but, on the other hand, it has led to an increase in racism and abuse. These opportunities and challenges brought about by technology have encouraged researchers to study, analyse and find solutions to deal with them.

Language is used not only by individuals to express their sentiments and emotions but also to demonstrate the intensity of such emotions. Although research into sentiment and emotion classification is widespread, studies on the level or intensity of emotion are scarce (S. Mohammad et al., 2018). Detecting the intensity of emotion can be useful in a variety of different contexts, as is evident from comparing two or more sentences that convey the same emotion at different intensities. For example, two customers may represent their experience thus:

1. The service was outstanding! Amazing! We cannot wait to visit you again!

2. Although the service was not outstanding, we were generally happy.

In this case, the first customer was extremely happy, while the second customer was less so. Determining the happiness level of customers will enable companies to understand more about customer impressions than simply rating the two comments as positive or happy, and this in turn will allow them to review and develop their services to raise the level of customer satisfaction.

Likewise, politicians and governments can identify a society's anger, whether intense or mild, towards political figures, elections and political affairs. By measuring such anger as conveyed in social media, the authorities can then make more effective decisions and predict communities' reactions. In public health, it is useful to differentiate between levels of negative emotions and their impact on mental health. For example, high and persistent feelings of sadness and/or fear can lead to depression and possibly suicide. Recognising such levels of feelings may help to track them early and avoid dangerous consequences.

While microblogging platforms can be used positively and productively, they may also be employed for destructive purposes, such as disseminating angry or offensive messages to others. Users who wish to spread insults can use these channels to reach millions of people at the click of a button. Such occurrences of online abuse have caused emotional and psychological health concerns for users, leading to reactions ranging from account deactivation to instances of self-harm and suicide (Kelly et al., 2018; Hinduja & Patchin, 2019; Kumar et al., 2020). To prevent this spread of negativity, systems are needed that can automatically identify messages containing offensive language from short texts or tweets.

Detecting emotional intensity and offensive language from text can be challenging, particularly in the context of social media microblogs such as Twitter. These difficulties relate to the limited number of words and significant noise in content, including typographical errors, slang and symbols. This task becomes even more complicated when morphology-rich languages, such as Arabic, are involved (Al-Ayyoub, Khamaiseh, Jararweh, & Al-Kabi, 2019). Social media communications typically consist of a range of dialects and sub-dialects that are not ruled by consistent standards. In this context, therefore, there is a need to adopt effective methods and resources to better comprehend and treat a variety of linguistic forms when seeking to understand emotional intensity and offensive language in Arabic short texts.

1.2 Problem Definition and Research Questions

In this thesis, we study two main problems: detection of emotional intensity and of offensive language in Arabic microblogs. In Chapter 3, we propose a novel combination of static character- and word-level embeddings to improve the detection of emotional intensity. In Chapter 4, we enhance contextualised language models by incorporating our proposed static embeddings for identifying emotional intensity. In Chapter 5, learning from emotional intensity and other emotion-related tasks is transferred to offensive language detection. This section describes the research problems and formulates related research questions (**RQ**).

1.2.1 Emotional Intensity Detection

The main objective of this task is to develop techniques that can automatically detect the intensity of four main emotions (sadness, anger, joy and fear) from a given short text. Emotional intensity (EI) can be determined as an ordinal classification (-oc) or regression (-reg) task. For the EI-oc task, the given short text should be analysed and allocated to a class from 0 to 3, where 0 refers to an emotion unrelated to the target, 1 for the lowest EI and 3 for the highest EI that can be inferred. On the other hand, the EI-reg task is annotated by real-value scores, ranging from 0 (the lowest intensity) to 1 (the highest intensity).

Word embedding is one of the most important methods used for many natural language processing tasks (Devlin et al., 2014; J. Zhang et al., 2014; Lin et al., 2015;

Bordes et al., 2014). While most research work is concerned with building word-level embedding models for Natural Language Processing (NLP) tasks in general or sentiment analysis, to our knowledge there are no models available for use at the word- and character-level designed to detect emotion intensity. This brings us to the first research question that explored and answered in Chapter 3:

> **RQ1:** To what extent can generating character- and word-level embeddings improve the detection of emotional intensity? In addition, can a combination of character- and word-embedding models enhance the accuracy of emotional intensity detection?

Recently, Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2019) has been shown capable of generating effective representation for NLP tasks in various contexts. These dynamic language models have been effectively applied to Arabic sentiment and emotion classification (Abu Farha & Magdy, 2021; Al-Twairesh, 2021). However, to the best of our knowledge, no work has employed contextualised word embeddings for detecting emotional intensity in Arabic. This leads us to the second research question, which is investigated and answered in Chapter 4:

> **RQ2:** Which pre-trained language models yield the most benefit in emotional intensity detection? Can integrating both static word embeddings and contextual language models enhance the detection of emotional intensity?

1.2.2 Offensive Language Detection

This work will use transfer learning approaches to improve the effectiveness of the offensive language task. This is a binary classification task, in which a given short text should be predicted as offensive or inoffensive. The majority of proposed methods target offensive language identification as a single task. However, to the best of our knowledge, there is no existing research that aims to incorporate emotional intensity into the offensive language detection task. This leads us to the third research question, which is explored and answered in Chapter 5:

> **RQ3:** Can offensive language detection be improved by transferring emotional intensity and affect-related features? Which affect-related features are most beneficial in offensive language detection?

1.3 Contributions

In this thesis, we make three main contributions to the literature. The first two contributions focus on detection of emotional intensity; the third concentrates on the detection of offensive language. They can be summarised as follows.

1.3.1 Combining Character- and Word-level Embeddings for Emotional Intensity Detection

- We create word- and character-level embeddings using a large number of tweets enriched by the diversity of affective vocabulary words and Arabic dialects.

- Our generated models are released to be used as pre-trained word embeddings for applications and research into the analysis of Arabic sentiment and emotion.

- We propose a novel method of combining character- and word-level embeddings to improve the detection of emotional intensity.

- We perform a systematic analysis to investigate the effectiveness of applying pre-processing techniques to a large training corpus prior to generating word embeddings a study that has not previously been examined for noisy user-generated text.

- We use six datasets to evaluate the performance of using our models as input features into machine learning algorithms.

1.3.2 Enhancing Contextualised Language Models with Static Character and Word Embeddings for Emotional Intensity

- We use our generated ACWE as input to various deep learning approaches to examine the impact of using such advanced learning on emotional intensity detection.

- We provide a comprehensive comparison of the effectiveness of six contextualised language models and evaluate them on emotional intensity datasets, such a study has not previously been conducted.

- We propose a novel method for enhancing contextualised language models by incorporating ACWE in emotional intensity tasks.

- Our proposed method improves the performance of language models and achieves state-of-the-art results for emotional intensity detection.

- We use eight datasets for related tasks to evaluate the robustness of the proposed method, which yields state-of-the-art or competitive results.

1.3.3 Affect Transfer Learning for Arabic Offensive Language Identification in Social Media

- We investigate the offensive language datasets and their relationship to emotional intensity and other affect-related tasks.

- We propose several combinations of affect-related features to transfer the most effective for offensive language detection.

- We compare the performance of different word-embedding and language models for offensive language detection.

- We use two datasets to evaluate the robustness of the proposed model, which yields state-of-the-art.

1.4 Thesis Organization

This thesis studies two main problems: emotional intensity and offensive language detection in Arabic microblogs. In Chapter 2, we first briefly describe the background of Arabic and the main challenges of automatically processing Arabic. We then discuss the concept of emotion models and emotion-related tasks from a NLP perspective. Subsequently, we review studies focused on generating Arabic pre-trained embedding models. In addition, we describe a recent effective approach for representing words: contextual language models. For classifying short texts or tweets, we discuss three main learning methods: machine learning, deep learning and transfer learning. Finally, we present a summary of the most effective proposed approaches for emotional intensity and offensive language detection.

In Chapter 3, we first introduce the generation of character-level and word-level embeddings pre-trained on a massive number of tweets. We then employ a novel method that combines both levels of models (ACWE) to represent each word morphologically and semantically. We evaluate the effectiveness of our proposed method using six datasets for emotional intensity and affect-related tasks.

Chapter 4 presents various DL approaches to examine the impact of using ACWE with advanced learning on emotional intensity detection. In addition, we examine the performance of different contextualised language models for emotional intensity detection. Finally, we propose a method for enhancing contextualised language models by integrating ACWE in emotional intensity tasks. Our proposed method improves the performance of language models and achieves state-of-the-art results for emotional intensity tasks.

Chapter 5 aims to improve the performance of the offensive language task by using transfer learning approaches. First, a combination of pre-trained static word-embedding and contextual language models is used as a form of transfer learning. Additionally, emotional intensity and other emotion-related tasks are leveraged as a feature transfer learning method. Our proposed transfer learning method achieves state-of-the-art in the offensive language detection tasks.

Chapter 6 summarises the contributions of the previous chapters and discusses future research directions.

1.5 Publications

- Alharbi, A. I., & Lee, M. (2020, June). Combining character and word embeddings for affect in Arabic informal social media microblogs. In International Conference on Applications of Natural Language to Information Systems (pp. 213-224). Springer, Cham. (**Best Paper Award**)

- Alharbi, A. I., & Lee, M. (2020, May). Combining character and word embeddings for the detection of offensive language in Arabic. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (pp. 91-96).

- Alharbi, A. I., & Lee, M. (2020, December). BhamNLP at SemEval-2020 Task 12: An ensemble of different word embeddings and emotion transfer learning for Arabic offensive language identification in social media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 1532-1538).

- Alharbi, A. I., Smith, P., & Lee, M. (2021). Enhancing contextualised language models with static character and word embeddings for emotional intensity and sentiment strength detection in Arabic tweets. Procedia Computer Science, 189, 258-265.

- Alharbi, A. I., & Lee, M. (2021, April). Multi-task learning using a combination of contextualised and static word embeddings for Arabic sarcasm detection and sentiment analysis. In Proceedings of the Sixth Arabic Natural Language Processing Workshop (pp. 318-322). (**Winning System Award**)

- Alharbi, A. I., Smith, P., & Lee, M. (2022). Integrating character-level and word-level representation for affect in Arabic tweets. Data & Knowledge Engineering.

Chapter 2

Background and Related Work

2.1 Arabic Microblog Informal Text

Spoken by over 400 million people and the official language in 22 countries, Arabic is the fourth-most-common language on the Internet. It has, furthermore, recently experienced rapid growth in terms of online usage, with the number of Arabic speakers on the Internet growing by an astonishing 9348% in the last two decades (*Top Ten internet languages in the world - internet statistics*, 2020). Arabic comes in three forms, namely Classical Arabic (the original form), Modern Standard Arabic (MSA), and Dialectal Arabic (DA) (Habash, 2010). Over the centuries, the original form developed into MSA, which is now the official language in Arab countries and has a written syntax and morphology.

Meanwhile, DA, which is the colloquial language of daily life, differs between countries and can even vary within a single country. While several hundreds of local varieties are recognized by Arabic dialectologists, there is a general agreement that these can be sorted into the five main regional dialects, namely Iraqi, Levantine, Egyptian, North African and Gulf Arabic (Elnagar et al., 2021; Alshutayri & Atwell, 2019; Bouamor et al., 2018). Unlike MSA, DA does not uniformly follow specific grammar rules, and it contains numerous words with different pronunciations. Furthermore, DA uses a number of words taken from different languages and words that are specific to a given dialect.

Researchers in the field of NLP generally face several modelling challenges regarding Arabic. Specifically, these complexities include dialectal variation, morphological richness orthographic variation, and a lack of resources, as detailed below.

Dialectal Variation

The languages of daily use are the native dialects learned by children as they grow up. These dialects are mostly spoken but are increasingly also written due to the significant growth of online user-generated platforms. These dialects have unique vocabularies and grammar rules that are not only dissimilar from one another but also differ substantially from those of MSA. These differences are sometimes caused by replacing one letter or more in an MSA word. For example, (متضايق- mtDAyq[1] - upset) is an MSA word that can be observed when the letter (ض- D) is replaced with (د- d) or (ز- z) in Egyptian dialects. (متزايش- mtzAy$) is another variant that can be seen, in Moroccan dialects, as an example of replacing multiple letters. In addition, entire words can be used in dialects that do not even exist in MSA. For example, (فلة- fl p) is a Gulf dialect word used to describe something impressive or what makes a person feel happy.

Morphological Richness

Because Arabic involves considerable inflection in terms of number, gender, person, case and aspect, in addition to several attachable clitics, Arabic words can take one of many forms. Consequently, individual Arabic words exist that, when translated into English, comprise five-word sentences, e.g., (وسأفرح - ws>frH- and I will be happy). Arabic thus also has more unique word types than English, leading to considerable difficulties for machine learning models. While the abovementioned example was drawn from MSA, in DA, this problem becomes even more severe, with the dialects' morphological

[1] Buckwalter's transliteration is used to represent Arabic orthography with morphological information (Buckwalter, 2004)

differences manifesting in affixes and clitics that are absent in MSA: in three dialects, the example shown above would be (hAfrH, bAfrH and mnfrH).

Orthographic Variation

In written Arabic, optional diacritical marks are used to denote short vowels, in addition to phonological information crucial to identifying individual words. These marks are usually used in children's books or religious texts to avoid orthographic ambiguity. However, these diacritics are not common in contexts such as articles, blogs and social media. An example of this orthographic ambiguity can be seen in the aforementioned word فلة, which can also refer to a house (فِلّة- fil p). It is worth mentioning that dialectal words are spelt in the way in which they are pronounced, which leads to orthographic variations due to phonological variations.

Resource Poverty

In NLP, the lack of an annotated dataset represents a constraint during training in supervised learning settings. Building such resources requires human interaction in the form of manually labelling each document or sentence, which is a time-consuming and very expensive process. While MSA has seen more resources extracted from news, Wikipedia and other classical resources, this is not the case for DA. More effort is needed to create and publicly share both annotated and unannotated corpora for DA (Guellil et al., 2021). In addition, there are no effective morphological tools or analysers, such as stemming, that can be used effectively to pre-process and treat DA words. Farasa (Abdelali et al., 2016) and Madamira (Pasha et al., 2014) provided effective results only for MSA and Egyptian dialects.

Each of the above factors is not specific to the Arabic language. Morphologically rich languages include Turkish and Finnish, orthographical ambiguity can be found in Hebrew and dialectical variation is a prominent feature of several languages. Nevertheless, through combining these features, DA presents a different level of complexity, introducing a special case of study in the field of NLP (Darwish et al., 2021).

2.2 Emotion Analysis: Background

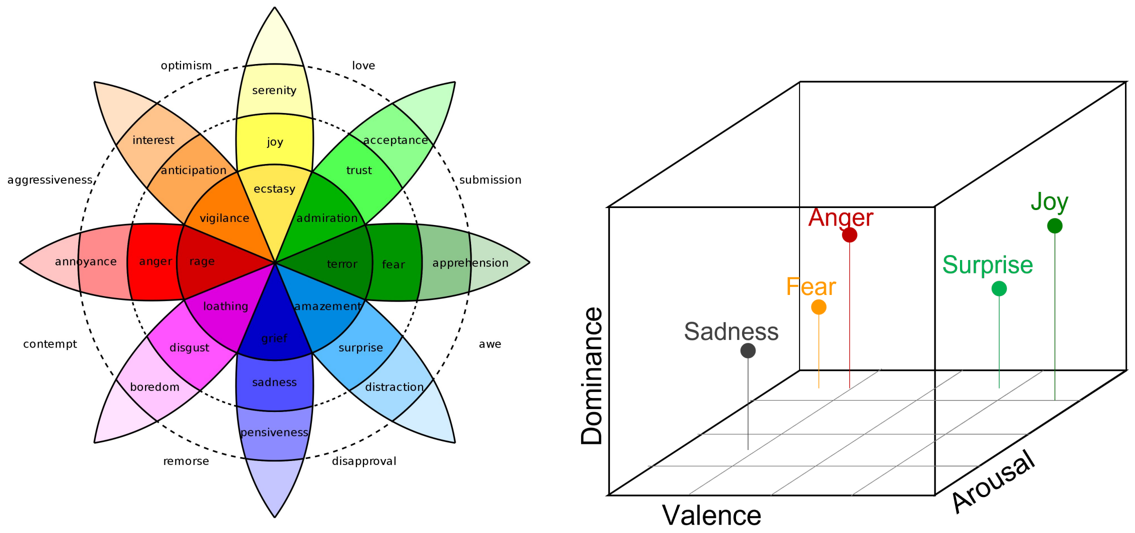
This section provides important background information related to emotion analysis and other related tasks. We briefly describe the concept of emotion models, followed by a discussion of the levels of analysis that must be considered when dealing with the problem of emotion analysis.

2.2.1 Emotion Models

According to Borod (2000), emotion models are techniques of classifying human emotions based on a ranking system or criteria. Human emotions may be classified based on their forms, levels and other characteristics. These variables can be combined to construct emotion models. Calvo and Kim (2013) distinguish between dimensional and categorical emotion frameworks.

The category model assumes that there are fundamental emotions that may be categorised. Ekman (1992) proposes a basic-emotion model reflecting this concept, with six basic emotions: joy, anger, fear, sadness, surprise, and disgust. Plutchik (1980) proposes a taxonomy of eight emotions and divides them into four binary pairs: joy versus sadness, anger versus fear, trust versus disgust, and surprise versus anticipation, as presented in Figure 2.1a. It is worth mentioning that there is no specific number of emotional classes used. There are several hypotheses, each of which implies a distinct and constantly shifting system of categorisation.

In terms of dimensional models, each emotion is assigned a place within a larger emotional categorisation using the dimensional method. Russell's Circumplex Model of Affect (Russell, 1980) arranges the emotions in a two-dimensional circular space with an arousal dimension and a valence dimension. Arousal delineates between activation and deactivation states, while valence specifies the extent to which an emotion is pleasant



(a) Plutchik emotion model (Plutchik, 2001)

(b) Mehrabian emotion model

Figure 2.1: Example of Emotion Models

or unpleasant. Mehrabian's model (Mehrabian, 1980) employs a three-dimensional approach consisting of valence, arousal and dominance (VAD). It represents a further example of a dimensional method. The dominance aspect, in this paradigm (2.1b), reflects whether the person feels in command of their circumstances.

Because of the ease of understanding and employing the categorical method, it is the most commonly used models for emotion analysis tasks in the field of NLP (S. Mohammad, 2016). However, emotional categories and emotional dimensions are used to symbolise diverse states. This implies that no particular emotion framework outperforms another. In fact, the choice of an emotion model depends mainly on the nature and objective of the study case. In this thesis, we use a public dataset released by S. Mohammad et al. (2018) which considers the intensity or level of four emotions from the Plutchik emotion model: anger, joy, fear and sadness.

2.2.2 Affect-related Tasks

In this section, we provide a brief background for several affect tasks that can be seen in NLP literature. We use the word ‘affect’, following (S. Mohammad et al., 2018), to mean a range of sentimental and emotional tasks. For the purpose of this dissertation, we distinguish between these tasks as follows:

Sentiment Analysis (SA)

SA is one of the most common affect tasks, one that is widely studied in the NLP research area. SA aims to classify a text into positive or negative, and many research work adds a third classification indicating neutrality. This simple classification has encouraged the use of SA in analysing people's opinions of or attitudes toward a particular product, event or topic. Although interest in Arabic SA has recently increased, it still lags far behind as compared to other high-resource languages, such as English, French and Chinese (Al-Ayyoub et al., 2019).

Over the last decade, researchers have given considerable attention to Arabic SA due to the large quantity of data available from Arabic social media that reflects opinions and sentiments. In both MSA and specific dialects, there are large numbers of SA datasets. One of the early works that focused on MSA was proposed by (Abdul-Mageed & Diab, 2011). They constructed a labelled corpus extracted from news sources and manually annotated for the task of SA. They further extended the dataset by involving different domains (Wikipedia and Web Forums)(Abdul-Mageed & Diab, 2012). LABR (Aly & Atiya, 2013) is an example of the largest annotated corpus (63K) that has been extracted from books.

In terms of annotated DA datasets, the majority of early studies focus on a specific dialect. ASTD (Nabil et al., 2015a) is one of the earliest works aimed at building a corpus of around 10K tweets, mainly in the Egyptian dialect. The collected tweets were then labelled into four classes: Negative, Neutral, Positive or Objective. AraSenTi (Al-Twairesh et al., 2017) is another corpus for SA that contains 17,573 tweets from MSA and Saudi dialects. These tweets were annotated using four classes (Negative, Neutral, Positive or Mixed). A shared task was organised in SemEval 2017 for SA in several languages, including Arabic. The size of the Arabic dataset is 9,455 tweets annotated using three main labels: Negative, Neutral and Positive.

More recently, ArSen (Abu Farha & Magdy, 2020a) has been generated using a combination of SemEval 2017 and ASTD datasets. The ASTD tweets were subsequently reannotated into three sentiment classes (negative, neutral and positive). The tweets were composed in multiple Arabic dialects and annotated using Amazon's Mechanical Turk. It is worth mentioning that sarcasm annotation was added because the authors believe there is a strong correlation between sarcasm and sentiment. They further extended the dataset with extra tweets, yielding a larger dataset (Abu Farha, Zaghouani, & Magdy, 2021) of around 15K tweets. ASAD, publicly released by (Alharbi et al., 2021), is the largest sentiment analysis of Arabic tweets (95K tweets) to date. Each tweet was annotated with one of the three classes (Positive, Negative or Neutral). The dataset contains various Arabic dialects.

Emotion Classification (EC)

The aim of this task is to classify a given text in a way that is fine grained and moves beyond the polarity of SA. It can be classified into a larger set of emotions, such as anger, joy, fear, sadness and disgust. Such classification typically adopts a standard set of emotions, varying in number and based on the psychological theories of emotion mentioned in Section 2.2.1. The majority of the earliest works on EC studies were based on the categorical emotion model, and a few works focus on the dimensional model for English (Preoţiuc-Pietro et al., 2016; Buechel & Hahn, 2017).

El Gohary et al. (2013) proposed one of the earliest studies on Arabic EC. They constructed a dataset consisting of 2,514 sentences extracted from children's stories (MSA). These sentences were annotated using six emotions (anger, disgust, fear, joy, sadness and surprise), adopting the basic emotions of the Ekman model. Similarly, (Rabie & Sturm, 2014) used the same emotion classification system. However, they created a corpus by collecting 1,776 tweets in the Egyptian dialect. Abdul-Mageed, AlHuzli, and Duaa'Abu Elhija (2016) also used the same emotion taxonomy, but they collected 3,000 tweets using a set of words regardless of focusing on specific dialects. Al-Khatib and El-Beltagy (2018) created a larger dataset with around 11K tweets filtered using the geo-location of Egypt. The authors added empathy and love to Ekman's emotions based on their observation of the collected tweets falling under these two emotions.

Sentiment Intensity (SI)

The aim of SI is to detect sentiment strength or level, which can be classified into five or seven ratings. This can be useful in helping a service provider, as an example, to understand customer sentiments more fully. A number of researchers have collected and built datasets based on the rating of five classes, ranging from 1 (very negative) to 5 (very positive) (Aly & Atiya, 2013; Elnagar & Einea, 2016; Elnagar, Khalifa, & Einea, 2018). The textual data of these works were taken from hotel and book reviews.

S. Mohammad et al. (2018) organised a shared task in SemEval 2018, and one of the main tasks was valance, which is another term for SI. They collected 2,600 tweets from multiple dialects. They annotated these tweets based on two tasks: valance ordinal and valance regression. For valance ordinal, the tweet was classified to one of the seven classes, ranging from -3 (very negative) to +3 (very positive), while 0 refers to the neutral class. On the other hand, the classes in the valance regression task are real-valued scores from 0 (very negative) to 1 (very positive).

Emotional Intensity (EI)

While there is a considerable body of work on the aforementioned tasks, research focused on EI is limited. The only annotated corpus in Arabic EI is proposed by (S. Mohammad et al., 2018). They collected tweets from multiple dialects and divided them into four emotion groups (anger, fear, joy and sadness). Each of these sub-datasets was annotated using two tasks: ordinal EI or EI regression. For each emotion, tweets from the ordinal EI task were classified using a range from 0 (no emotion) to

3 (high intensity for the given emotion). On the other hand, the EI regression task was scored using a real value from 0 (no emotion) to 1 (high intensity for the given emotion).

Sarcasm Detection

'Sarcasm' is defined as a figurative type of language in which the expression is meant to communicate the opposite of its literal meaning. Within sentiment and emotion analysis, detecting sarcasm is vital because sarcasm typically implies a negative feeling, even though positive language is used. Due to this noticeable correlation, researchers associated sarcasm with different affect tasks (Riloff et al., 2013; Bouazizi & Otsuki Ohtsuki, 2016; Felbo, Mislove, Søgaard, Rahwan, & Lehmann, 2017; Majumder et al., 2019).

Recently, Abu Farha et al. (2021) combined two sarcasm datasets (Abu Farha & Magdy, 2020a; Abbes et al., 2020) to create the largest corpus used in sarcasm detection tasks, with 15,548 tweets in multiple dialects (ArSarcasm). Each one of these tweets was labelled not only with sarcasm, if applicable, but also with a sentiment class. We decided to include sarcasm as one of the related tasks due to the strong correlation between sarcasm and the other affect tasks (B. Liu, 2020) and also due to the availability of such multi-annotated corpuses.

Thus far, we briefly described our main task (EI) and four related tasks. For each one of these tasks, we reviewed works that aimed to create annotated datasets, which is the most important resources for training machine and deep learning algorithms. In this thesis, we use these four related tasks to accomplish two main goals. Firstly, we use them to evaluate the robustness and generalisation of the proposed methods to detect emotional intensity (Chapter 3 and 4). Secondly, we exploit these affect-related tasks in the task of offensive language detection, as will be explained in Chapter 5.

2.2.3 Emotional Intensity Detection Levels

In general, emotional intensity can be analysed at multiple levels, including the level of words and the level of tweets or sentences. These distinct levels are described below.

Word Level

Words play an important role in understanding and expressing our emotions and their various intensities. Certain words, such as *fabulous*, *terrifying*, *depressed* and *cheerful*, carry an emotion that is central to what they mean. Other words lack such an emotion at their core but are still associated with an emotion. As an example, *celebration* and *promotion* have happy connotations, while *violence* and *insult* evoke anger. In addition, some words convey a high emotional intensity that is central to their meaning, while other words have only a low level of intensity. As an example, *exhilarated* and *pleased* indicate different levels of happiness.

There has been a reasonable amount of research on developing lexicons for Arabic sentiment and emotion classification (Badaro et al., 2014; Ibrahim et al., 2015; El-Beltagy, 2016; Al-Twairesh et al., 2016; Badaro, Jundi, et al., 2018). These lexicons have been built via manual annotation or automation approaches. By contrast, (NRC-Aff-Int) is the only lexicon for Arabic emotional intensity, as proposed by (S. M. Mohammad, 2018). This lexicon provides the real-valued affect intensity scores for four basic emotions: anger, fear, joy and sadness. The researchers selected (English) terms that are commonly used on social networks and manually annotated them. For the Arabic lexicon version, they translated all these English terms into Arabic using a machine translation system.

Manually constructed resources typically consist of a limited number of entries because the annotation process is expensive and time-consuming. Moreover, despite cultural differences, automated translation from English to Arabic leads to obtaining MSA words, that is, without the dialectal words. These limitations motivate the exploration of unsupervised methods with which to create word-level resources. Generating a word embeddings model, as an alternative word-level resource, has attracted increased attention in NLP. We will explain this in detail in Section 2.3.

Short-text Level

Systems operating at the short-text or sentence level aim to detect the emotional intensity score or class for an entire text. The assumption is that each short text (e.g., a tweet) expresses a level of emotion on a target or entity (e.g., an event). To achieve this goal, there are two main approaches: supervised learning, unsupervised learning.

Unsupervised learning has the advantage of not requiring labelled training data to develop a model for emotional intensity detection. Lexicon-based methods are an example of unsupervised learning. Lexicons are simply used to assign to the words in a given text their emotional intensity scores if those words are present in that lexicon. Lexicon-based methods have been largely applied to sentiment and emotion classification, particularly in traditional texts, such as forums, product reviews and blogs (Ding et al., 2008; Taboada et al., 2011). Only using lexicons as the main method has recently been less explored in texts written by social networks due to the complex nature of the text in English (Giachanou & Crestani, 2016), and this becomes even more challenging in Arabic (Al-Ayyoub et al., 2019).

In contrast, supervised learning methods are based principally on the presence of labelled training data. Training data are labelled using predefined classes or real-value, so a classification or regression model can be trained based on such labelled instances. Eventually, this classifier can be used to predict the class of new, unlabelled text for evaluation purpose. The proposed systems to detect emotional intensity using supervised learning have been approved as more effective than using only unsupervised learning. A variety of supervised learning algorithms can be used, as will be explained in Section 2.4.

2.3 Word Representations

2.3.1 Static Word-Level Embeddings

Most studies on Arabic word embedding focus on the application of word-level models (Zahran et al., 2015; Soliman et al., 2017; Abu Farha & Magdy, 2019; Altowayan & Tao, 2016), and to a lesser extent, on character-level models (Altowayan & Elnagar, 2017). An early study aimed at building word-level embeddings for Arabic (Zahran et al., 2015) employed three techniques (CBOW, skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)) to create word-level representations in a vector space for MSA. To pretrain their word embedding models, they used a significant corpus of Arabic texts (5.8 billion words) collected from several sources, including documents that had been translated, news articles, the Arabic Gigaword corpus (Parker et al., n.d.) and the Arabic Wikipedia. This model holds a 300-dimensional vector comprising six million words and phrases.

AraVec (Soliman et al., 2017) has one of the most well-known collections of open-source word embeddings, comprising six separate word embedding models for use with Arabic. The training data was derived from three sources: Twitter, Wikipedia and Common Crawl. Similar to (Zahran et al., 2015), they utilised CBOW and skip-gram to learn word representations for Arabic natural language processing (NLP) applications.

Recently, Abu Farha and Magdy (Abu Farha & Magdy, 2019) generated the largest word-level embedding model using 250 million Arabic tweets. Although numerous words were used to train the models, they could not identify the same words in different forms that were employed in real human speech, as a result of the limited nature of these word-level models. Generally, the effectiveness of the embedding depends on the task (Qu et al., 2015) and is affected considerably by the variety of words related to the task at hand (Çano & Morisio, 2017).

2.3.2 Static Character-Level Embeddings

To the best of our knowledge, there is no pre-trained Arabic character-level embedding model targeting Arabic dialects. Furthermore, we are not aware of any research investigating the impact of preprocessing techniques on the generated word embedding models. The only exception is the study by (Babanejad et al., 2020), who systematically studied different preprocessing factors in well-formed English datasets (e.g., Wikipedia and news databases). Our research generated character-level and word-level embedding models for informal Arabic social media content (noisy user-generated text). Additionally, we systematically compared the impact of the preprocessing on the effectiveness of the generated models at the character level and word level across six downstream tasks.

2.3.3 Combining Character-level and Word-level Embeddings

Initial attempts at combining character-level and word-level information in English were undertaken by Dos Santos and Gatti (2014), producing a deep neural network architecture capable of sentiment analysis (SA) using sentence-, word- and character-level representations. They used a pre-trained word-level embedding model that employs word2vec but did not include a pre-trained character-level model. Instead, character vectors were initialised using random sampling.

Recently, (Lei et al., 2018) proposed a word embedding model (charCNN) for integrating word representations from word-level and character-level models to capture morphological information, such as word suffixes and prefixes. Unlike the model created by Dos Santos and Gatti (2014), charCNN represents a fully conversational network with no max pooling layer for superior semantic information capture in character chunks. To the best of our knowledge, no study has attempted a combination of the two levels (the character level and word level) to generate word representations specifically for tasks on affect in informal Arabic text.

In contrast to the aforementioned studies, we separately pre-trained character n-grams and a word embedding model on a large dataset to learn semantics and morphology separately. We then combined the character n-grams and word embedding model as input features in a supervised learning framework for downstream tasks.

2.3.4 Contextual Word Embeddings

To generate a universal vector representation of a word, static word embeddings take into account the full array of sentences in which a word is used. However, the meaning of a given word can vary based on the context in which it is used. This led Peters et al. (2018) to introduce deep contextualised word representations, known as embeddings from language models (ELMo), and are designed to generate a more effective representation for NLP tasks in various contexts.

Devlin et al. (2019) recently proposed Bidirectional Encoder Representations from Transformers (BERT), which employs a transformer network to extract contextual word embeddings. BERT is different from ELMo in that it employs a range of pre-training tasks for the specific purpose of language modelling. Delvin et al. created two different model sizes: a base model comprising 12 encoder layers, and a large model with 20 encoder layers. BERT has pushed the state of the art forward for several NLP tasks (Devlin et al., 2019), They have also released a multilingual BERT model (mBERT) with target languages other than English.

The successful creation of monolingual BERT models for languages other than English led to the generation of contextualised word embeddings specifically for Arabic, such as AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2021), which have both been applied to various sentiment classification and emotion classification tasks. AraBERT and MARBERT outperform other models in terms of effectiveness performance across several benchmark datasets (Alomari et al., 2017; Nabil et al., 2015b; Al-Twairesh et al., 2017). However, we are not aware of any extant study that employs contextualised word embeddings for emotional intensity and/or sentiment strength in Arabic.

2.4 Learning Methods

2.4.1 Machine Learning Methods

The growing trend of employing machine learning methods in the field of NLP has led to a significant amount of research on emotion analysis (Ravi & Ravi, 2015). Machine learning approaches are based on the use of statistical techniques to provide computers with the ability to automatically build models from a given data set. A variety of machine learning algorithms can be employed in emotion analysis. These algorithms are subsequently discussed in brief.

Linear Regression

Typically, the objective of regression models is to use independent variables (X) to forecast a dependent variable (Y). They are usually employed to ascertain the link between predicted values and associated variables. As such, linear regression can be used to identify a linear relationship between an input (X) and an output (Y).

Support Vector Machines (SVM) and Support Vector Regression (SVR)

The Support Vector Machine (SVM) is one of the most common linear classifiers. Linear classifiers are designed to identify the most fitting separators that divide vectors into different classes. Drucker et al. (1997) introduced Support Vector Regression (SVR) by extending SVM to the resolution of regression problems; SVR is superior to the linear regression model in terms of making complex predictions. Figure 2.2 presents an example of a one-dimensional SVR. The line represent the label (y) data, while the data points represent the predicted values (y). The bounds that are ϵ in distance to the reference data are represented by the two dashed lines, with ϵ being a user-selected parameter. Only the outlier values (represented by the dashed lines) are used to construct the model using SVR. To train the SVR model, we must solve Equation 2.1 and Equation 3.2:

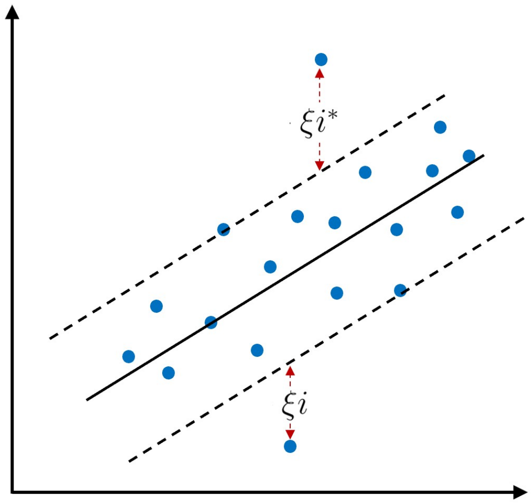


Figure 2.2: Support Vector Regression (SVR)

$$Minimise \ \frac{1}{2}\left\|w\right\|^2 + c\sum_{i=1}^{n}(\xi i^* + \xi i) \tag{2.1}$$

$$Subject\ to \begin{cases} y_i \left\langle w, x_i \right\rangle - b \leq c + \xi i^* \\ \left\langle w, x_i \right\rangle + b - y_i \leq \epsilon + \xi i \end{cases} \tag{2.2}$$

where y_i is the training tag or label, x_i is the i-th training data, w is the learned weight vector, and ξi represents the inter-bound distance and predicted values outwith the bounds. An extra user-determined parameter is ϵ, which is a constraint controlling the penalties exerted on observations outwith the bounds that mitigate the probability of overfitting.

Boosting Methods

Boosting is a means of taking weak learners and converting them into strong learners. With boosting, all new trees are fit onto an adapted version of the primary dataset. The gradient boosting method (GBM) is one of the most frequently applied boosting methods.

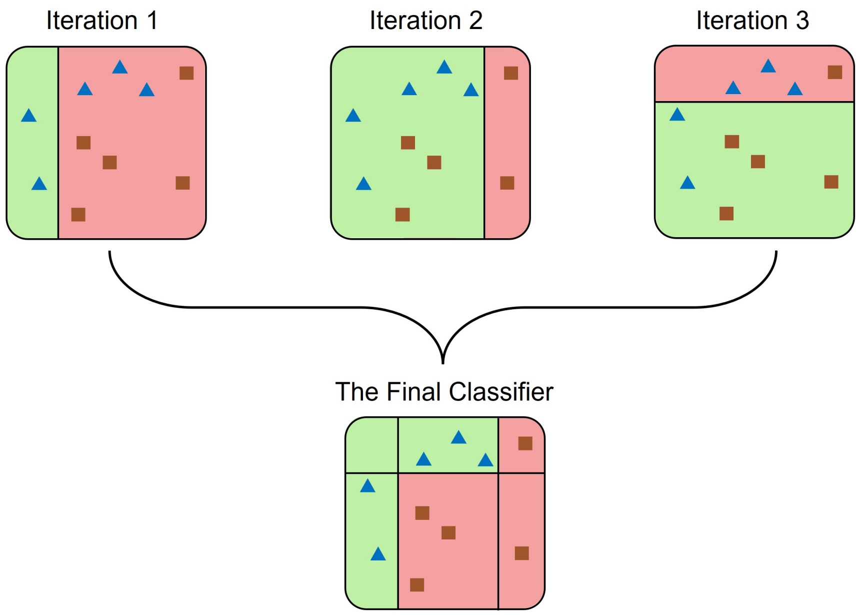


Figure 2.3: Example of three iterations used to obtain the final classifier

This algorithm commences with the training of a decision tree in which every instance or observation has equal weight. Once the initial tree has been assessed or evaluated, the weights of instances that offer a simple classification are decreased, while the weights of instances that are more problematic to classify are increased. This process is aimed at enhancing the accuracy of the predictions of the initial tree. Thus, the new classifier or model is *Tree 1* + *Tree 2*. The classification error is then calculated using the new combined double tree model, with a further tree being grown for the prediction of revised residuals. The process is then repeated for a specific cycle of iterations, as illustrated in Figure 2.3. Every extra tree is helpful for the classification of instances that were problematic for previous trees. Therefore, the weighted combination of all the predictions from the trained three models is the prediction of the final model.

Gradient boosting is a gradual, additive and sequential way of training many models. The gradient boosting algorithm undertakes the function using gradients from the loss function ($y = ax + b + e, e$ *requires particular attention because it represents an error term*). The loss function is a measurement that indicates how well the coefficients

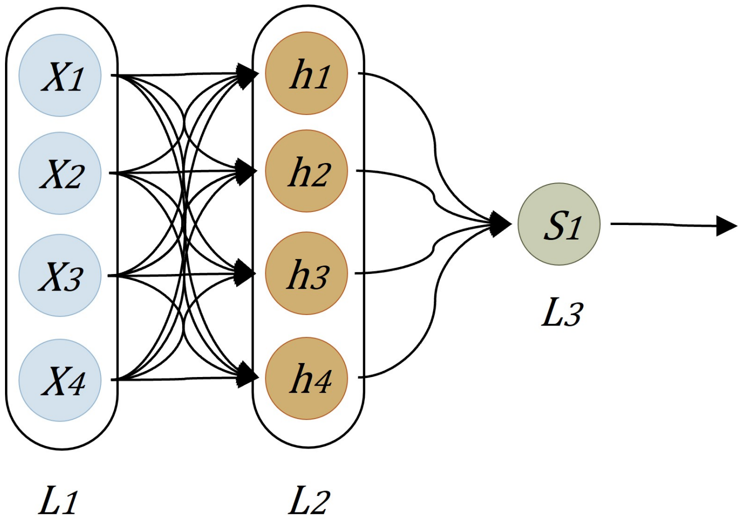


Figure 2.4: Feedforward neural networks

of the model fit with all underlying data. XGBoost (Chen & Guestrin, 2016) is an optimised distributed gradient boosting library that supports efficiency, flexibility and portability. It offers parallel tree boosting to solve numerous data science problems swiftly and accurately.

2.4.2 Deep Learning Methods

Neural Networks

Neural networks (NNs) are used in deep learning to learn tasks that involves networks with multiple layers. This exploits the superior learning capacity of neural networks, which was previously thought to be practical only with a minimal number of layers and small datasets.

Figure 2.4 presents a basic description of a feedforward neural network comprised of triple layers. $L1$ represents the input layer, corresponding to the input vector $(X1, X2, X3)$. The intercept term $L3$ represents the output layer, corresponding to the output vector $(S1)$. $L2$ represents a hidden layer, the output of which is unobservable as a network output. If $L1$ has a circle, it represents an element within the input vector. Circles in $L2$ or $L3$ represent neurons, which are the basic building blocks for computations in neural networks and may also be called activation functions. Neurons joined by a line are connected, enabling the flow of information. All connections have weights that regulate the signals passing between two neurons, and neural networks learn by adjusting these weights. Each neuron reads the output from the neurons in the pre-layer, performs information processing and then transmits the output of this processing to neurons in the next layer. As seen in Figure 2.4, the neural network modifies the weights based on training examples $(x(i), y(i))$. Once the training process is completed, the network generates a complex set of hypotheses $w, b(x)$ that are consistent with the data provided (Zhang et al., 2018).

Convolutional Neural Networks (CNN)

One of the first CNN models for NLP was proposed by (Kim, 2014). It has a simple deep learning architecture in which input sentences are converted into embedding vectors which are then fed into the model as a matrix. The pre-trained word embedding model is used to initialise the embedding layer weights. To generate features, convolutions are performed word by word using the input and in various kernel sizes, for example, two or three words at a time. The retrieved features are then compressed or reduced using a max pooling layer. Finally, these features are passed on to a fully connected softmax layer for downstream tasks. Another CNN architecture, which was proposed by (Y. Zhang et al., 2016), derives various features from different word embedding models instead of only one pre-trained model.

Recurrent Neural Networks (RNNs)

The logic underpinning the recurrent neural network (RNN) is to take the input sequence into account. To accurately anticipate the word that will appear next in a given sentence, it is crucial to recall the word that was featured in the previous time step. These are referred to as RNNs because this step in the process is performed for each distinct input. Because the previous word is taken into consideration during the process of generating a prediction in these neural networks, an RNN is comparable to a memory storage unit that stores information on a short-term basis.

Long Short-term Memory (LSTM)

LSTM is a type of RNN architecture specifically created to model temporal arrangements and their associated long-range dependencies. The activation function is not used within the recurrent components of an LSTM, stored values are not modified, and the gradient does not typically disappear during the process of training. Typically, LSTM units are executed in blocks that comprise multiple units. These blocks incorporate three or four gates (e.g., input gate, forget gate, output gate), which draw on the logistic function to control information flow. LSTM incorporates supplementary forget gates over the basic RNN. Its distinctive device allows it to avoid both the exploding gradient problem and vanishing problems (Y. Wang et al., 2016).

2.4.3 Transfer Feature Learning

The preceding learning methods assume that the training and testing processes are performed on the same task. Transfer learning, on the other hand, allows us to train on domains or tasks and then transfer the information to a different domain or target task. Transfer learning has been explored in several situations and is known by various names depending on the transferring approach. Notably, both pre-trained static word embedding and contextual language models are considered transfer learning approaches. In Section 2.3, we explained these word representations in detail. Other forms of transfer learning are discussed in this section, including multi-task learning and feature-representation transfer approaches (R. Liu et al., 2019; Ruder, 2019).

Multi-task learning (MTL) is an inductive transfer approach that enhances generalisation by using domain knowledge found in the training signals of similar tasks as a source of inductive bias. The intuition behind MTL is that a useful feature for one task will be useful— and thus predictive—for other similar tasks (Caruana, 1997). One of the studies to employ MTL settings to simultaneously learn several affect tasks is a study by (Akhtar et al., 2018), which demonstrates that sharing learning between related tasks is beneficial for the learning of each task and results in improved performance. Rajamanickam et al. (2020) proposed an MTL framework that incorporates emotion and abusive tasks.

In contrast to MTL, which involves tasks being learned simultaneously, there are two stages in the feature-representation transfer approach. First, the model is trained on the source task to learn feature representations, and then the knowledge learned is transferred to the target task. In this thesis, we use this approach to transfer features learned from anger intensity tasks and other affect-related tasks, and then leverage these features for offensive language detection tasks. To the best of our knowledge, no existing research has incorporated this range of affect tasks into an offensive language detection task.

2.5 Current State-of-the-Art Systems

2.5.1 Emotional Intensity Detection

Although considerable research has been performed with the aim of analysing emotions and sentiments, studies that examine emotional intensity (or emotional levels) are uncommon (S. Mohammad & Bravo-Marquez, 2017). Notable research on detecting emotional intensity can be found in SemEval 2018 Task 1 (Affect in Tweets) (S. Mohammad et al., 2018). Most of the proposed systems that performed well at the competition combined machine learning with deep neural networks. The learning algorithms in these systems were fed text representations and features extracted from current lexicons related to emotions and sentiments.

AffecThor (Abdou et al., 2018) emerged as the best-preforming Arabic emotional intensity detection method for both classification and regression tasks. The researchers put forward a system that combined hand-crafted lexicons with pre-trained word-level embeddings built using 4 million tweets. These integrated representations were employed as input in a supervised learning framework. The training was performed using an ensemble deep network architecture comprising BiLSTM with attention and CNN with max pooling.

Similarly, Jabreel and Moreno (2018) put forward an ensemble method that combines two models. The first model, n-channels ConvNet, is based on a deep learning methodology, while the second model is an XGBoost regressor based on features based on lexicons and embedding models. This ensemble technique facilitated improved performance on emotional intensity detection tasks. The system was ranked second place based on the Pearson correlation results.

The system put forward by (Abdullah & Shaikh, 2018) was ranked third place on regression tasks, while third place the classification tasks went to the Emotion Mining in Arabic (EMA) system (Badaro, El Jundi, et al., 2018). Both systems employed pre-trained word-level embeddings (AraVec) within supervised learning frameworks. The EMA system applied additional processing methods such as stemming and manual conversion of emojis to their corresponding text descriptions.

SEDAT (Abdullah et al., 2018) is an example study that involves the use of machine translation to overcome the deficiency of resources for emotional intensity detection in Arabic. They proposed two models: ArTweets and TraTweets, which were combined by weighting their predictions. Similar to the aforementioned approaches, ArTweets uses Ara2vec in combination with combined deep neural networks (CNN-LSTM). For TraTweets, they translated Arabic tweets into English to benefit from the available English resources. The extracted features for this model resulted in an input 4908-dimensional vector, which was fed into a deep neural network. Although SOTA English features were used in the SEDAT study, their models did not outperform the AffecThor system, which used only Arabic resources.

More recently, AlZoubi et al. (2020) developed an ensemble approach to target emotional intensity regression tasks. They integrated three models: BiGRU-CNN, CNN and XGB Regressor, and extended the training dataset using a semi-supervised approach called pseudo-labelling learning (Lee et al., 2013). The concept behind this method is based on simply training the model using available labelled tweets and then automatically labelling unlabelled tweets using the trained model. Our proposed approach outperforms the best-performing system from the SemEval 2018 Task 1 (AffecThor) by 0.7%.

From the aforementioned proposed methods, we observe that ensemble models of machine learning and deep learning algorithms and combinations of word-level embeddings and lexicons are the primary factors common to most of these studies. However, we are unaware of any research that has used either pre-trained character-level embeddings or contextual language models specifically for emotional intensity detection in Arabic texts.

2.5.2 Offensive Language Detection

There are several extensive studies on offensive language detection in English across various categories, including abusive language, sexism, religious hate speech, and racial hate speech detection (Davidson et al., 2017; Malmasi & Zampieri, 2017; Kumar et al., 2018; Waseem et al., 2017; Zampieri et al., 2019). In contrast, only a few studies have been conducted in this area for the Arabic language (Mubarak & Darwish, 2019a).

One of the earliest studies on the detection of offensive language in Arabic was conducted by (Mubarak et al., 2017). They argue that some users have a higher likelihood of using offensive language than others. They then used this insight to construct a list of Arabic words that are offensive. Subsequently, they developed an extensive corpus of Arabic tweets that were manually annotated into three categories: clean, obscene and offensive.

Another significant contribution was made by Alakrot et al. (2018), who developed a corpus of offensive Arabic comments that had been shared on YouTube, creating a dataset that includes 16,000 comments in specific Arabic dialects (Egyptian, Libyan and Iraqi). The comments are categorised into one of three classes: offensive, inoffensive and neutral. They then trained an SVM classifier to detect the offensive comments. Based on their experiments, they concluded that using n-gram features improves the accuracy of the classifier, while a combination of n-gram features and stemming negatively impacts the performance of the system.

Mubarak and Darwish (2019a) expanded the list of offensive words compiled by Alakrot et al. for their research (Mubarak et al., 2017), using the expanded list to build a massive training corpus for automatic offensive tweet detection. They investigated three methods for classifying each tweet as either offensive or unoffensive: a lexicon-based approach, SVM, and a deep learning method. For the deep learning method, they employed a character-level classifier that achieved better results than the other two methods.

More recently, two shared task competitions on *Arabic offensive language detection* were conducted to contribute to the development of this area (Mubarak et al., 2020; Zampieri et al., 2020). The researchers released a dataset containing 10,000 tweets in multi-dialects. The tweets were then manually labelled by native Arabic speakers as either offensive or unoffensive. Tweets labelled as offensive were further annotated as hate speech or non-hate speech as part of another task (hate speech detection). However, our aim in this thesis is to focus primarily on general offensive language detection. These two competitions have led to several new approaches to offensive language detection in Arabic being proposed. The majority of the proposed methods approach offensive language identification as a single task and most are based on single task supervised settings.

The best-performing system using the aforementioned datasets was proposed by Hassan et al. (2020). They implemented several preprocessing methods, including the removal of repeated letters, non-Arabic characters, punctuation and diacritics, and they experimented with simple and advanced learning algorithms: SVM, CNN, CNN-

BiLSTM and mBERT. The proposed system combined these algorithm models by performing majority voting as an ensemble method.

Alami et al. (2020) proposed an approach that utilises an Arabic-specific BERT (AraBERT). They added [MASK] (the special token in BERT) instead of emojis and then added a description translated from English to Arabic for each emoji in the tweet. They found that this process outperformed methods that used the vanilla model. Similarly, Keleg et al. (2020) used AraBERT with a list of offensive words as additional support. They found that a manually built list for identifying profane words as offensive can support machine classifiers and improve their performance.

S. Wang et al. (2020) proposed a multilingual approach using XLM-R, with Arabic included. They fine-tuned XLM-R in all languages and predicted the final classes based on 10 cross-validation ensembles. They achieved a competitive result (F1 score of 89.89%). Safaya et al. (2020) proposed a combination of BERT and CNN using ArabicBERT, and they also explored using mBERT and CNN with randomly initialised embeddings. The proposed system obtained similar results to those obtained by S. Wang et al. (2020).

Husain (2020) investigated the impact of intensive preprocessing methods, relying on the assumption that preprocessing aids dimensionality reduction and facilitates the removal of irrelevant data in Arabic social media posts. Similar to Hassan et al. (2020), they converted emojis to their equivalent Arabic textual descriptions. To reduce variation in dialects, a manual set of terms was built to convert several dialectal words to MSA, and common animal names were replaced with the general category *animal*. For the training process, SVM was used in combination with character-based features of 2 to 5 grams. The proposed approach achieved a competitive result, just 0.7% behind (Hassan et al., 2020).

Concurrently, a few studies attempted to take advantage of similar tasks using transfer learning approaches. Djandji et al. (2020); Abu Farha and Magdy (2020b) proposed an MTL method by including a hate speech detection task. Sentiment and emotion classification tasks were also leveraged for offensive language detection in Arabic (Elmadany et al., 2020). First, they used mBERT to fine-tune a sentiment and emotion dataset (Abdul-Mageed et al., 2020) and then the trained BERT models were exploited to further fine-tune the dataset for offensive language detection tasks. Husain and Uzuner (2021) provides a comprehensive survey of Arabic offensive language detection by reviewing applied methods and available resources.

In this thesis, we employ a different form of transfer learning approach (i.e., transfer feature learning) to benefit from a range of different affect tasks, including emotional intensity detection tasks. To the best of our knowledge, no existing research has incorporated emotional intensity into offensive language detection tasks.

2.6 Summary

This chapter provides an overview of Arabic emotional intensity and offensive language detection. In its opening, we present a brief background of the Arabic language and the most prominent challenges to the automatic processing of the Arabic language. These challenges are generally linked to linguistic characteristics and a dearth of resources for Arabic dialects in particular. We then briefly discuss the concept of emotion models and then emotion-related tasks from the perspective of NLP. We also mention two levels of analysis (word and short text) that are typically considered when dealing with the challenge of emotion analysis. Subsequently, we delved in depth into a discussion on the effective proposed techniques and methods for targeting these two levels of analysis.

For word-level analysis, we discussed character and word embeddings as one of the effective word representation methods. We then reviewed studies focused on generating Arabic pre-trained embedding models. In addition, we described a recent effective approach for representing words: contextual language models. For short-text level analysis, we discussed three main learning methods: machine learning, deep learning and transfer learning. Finally, we presented a summary of the available annotated datasets and the most effective proposed approaches for emotional intensity and offensive language detection.

Chapter 3

Combining Static Character and Word Embeddings for Emotional Intensity Detection

In this chapter[1], we use a combination of character-level and word-level models to discover more effective methods to represent Arabic emotional intensity words in short-text or tweets. We evaluate our embeddings by incorporating them into a supervised learning framework for a range of affect tasks. Our models outperform the state-of-the-art, Arabic pre-trained word embeddings in these tasks. In addition, our models enhance the previous state-of-the-art in the Arabic emotion intensity classification task, outperforming the top systems used in advanced ensemble learning models and several additional features.

3.1 Introduction

Language is not only employed by human beings for expressing emotions or sentiment; it is also used to display the intensity of such feelings. The term “affect”refers to a variety of categorisations related to emotions, which range from classifying sentiments

[1] This chapter is adapted from A. I. Alharbi et al., 2022

(positive-negative) and finer grained categorisation of how strong a sentiment or emotion is (e.g., extreme sadness, mild sadness). It is a significant challenge to detect affect in text, particularly when looking at social media, e.g. Twitter, because the language employed is constrained by numerical limits and is also highly informal, using both symbols and slang.

However, it is an even greater challenge when looking at languages that have a rich morphology, such as Arabic (Al-Ayyoub et al., 2019). Arabic social media users usually employ a variety of dialects and sub-dialects when communicating. While Modern Standard Arabic (MSA) has certain standards and rules, Arabic dialects used on social media usually lack such rules and standards. Thus, when examining Arabic affect in tweets, we need to develop tools and resources that can offer a better understanding and interpretation of the varied linguistic forms employed.

One of the central techniques in NLP is word embedding (Devlin et al., 2014; J. Zhang et al., 2014; Arslan et al., 2018; Mahmoud & Zrigui, 2019; X. Li et al., 2019). Word embedding employs dense vectors for representing words that project into continuous vector space, which reduces dimension numbers (Mikolov et al., 2013). Nevertheless, such models can be ineffective when used with Arabic tweets. When we attempt to train the word-level models with informal language, it has been demonstrated that such models have difficulty in recognising a variety of forms of the same words that share meanings. Such unknown words, referred to as out-of-vocabulary (OOV) problem, are one of the primary limitations of word-level embedding models.

In contrast, using character-level embedding can be effective in overcoming OOV words by employing their capacity for learning character n-grams (word parts). Nevertheless, character-level embedding is so sensitive that the model will encode every variant of the morphology of the word that has greater closeness within the embedded space than words which are similar semantically. Table 3.1 illustrates a pair of examples of emotional intensity words in Arabic dialects متنرفز (mtnrfz)[2] and مروق

[2] Buckwalter's transliteration is used to represent Arabic orthography with morphological informa-

Table 3.1: Most similar words of different affect words using character and word level embeddings.

	Character-level model	Word-level model
Example of a negative query term: متنرفز mtnrfz (uptight)	متنرفزه mtnrfz (uptight-feminine)	معصب mESb (angry)
	متنرفزين mtnrfzyn (uptight-plural)	متوتر mtwtr (tense)
	نتنرفز ntnrfz (uptight-present verb)	متضايق mtDAyq (annoyed)
	بيتنرفز bytnrfz (uptight-future verb)	متنرفزه mtnrfz (uptight-feminine)
	تتنرفز ttnrfz (uptight-feminine verb)	منفس mnfs (furious)
Example of a positive query term: مروق mrwq (relaxed)	ومروق wmrwq (and relaxed)	مصحصح mSHSH (mindful)
	مروقه mrwqh (relaxed-feminine)	ومروق wmrwq (and relaxed)
	ومروقه wmrwqh (and relaxed-feminine)	فايق fAyq (awake)
	رايق rAyq (relaxed)	مفلل mfll (restful)
	رايقه rAyqh (relaxed-feminine)	مستانس mstAns (happy)

(mrwq), where word similarity is generally derived from character-level morphology and word-level semantics.

In this chapter, we take advantage of character- and word-level models to discover an effective means of representing Arabic affect in tweets; the resulting model is called Affect Character and Word Embeddings (ACWE). Initially, each model was trained with a large collection of tweets that were specifically selected to ensure demonstrable variations in affect terms from different Arabic dialects. A novel method was then used to concatenate the two models so that each word was represented semantically and morphologically.

tion (Buckwalter, 2004)

The ACWE model was evaluated by applying it as an input feature under a supervised learning framework using emotional intensity benchmark datasets from SemEval-2018 Task 1 (Affect in Tweets) (S. Mohammad et al., 2018) and other tasks that were related to the affect area of study (sentiment analysis, emotion classification and sarcasm detection). Our method advances a state-of-the-art approach to the task of Arabic in emotional intensity, and it outperformed top systems that used combinations of deep neural networks and several other features. Additionally, our method obtained superior outcomes compared with other Arabic pre-trained word embedding models. ACWE has been released for use in pre-trained word embeddings for applications and research relying on Arabic sentiment and emotion analysis and related tasks[3].

The rest of this chapter is organised as follows. Section 3.2 and 3.3 provide a detailed discussion of how the data we used was collected and pre-processed. Section 3.4 explains our methodology for generating word-level and character-level embeddings and also how they are combined. Section 3.5 describes the experimental setup which includes the datasets, off-the-shelf pre-trained word embeddings and supervised learning models. Section 3.6 presents the results of using the experimental models on the downstream tasks. The main findings of our research are discussed in Section 3.7. Finally, Section 3.8 concludes the chapter and provides some suggested future directions.

3.2 Data Collection

The size and variety of the training dataset are major aspects to be considered in improving word embedding quality. To this end, 10 million tweets were gathered using the Twitter API. We aimed to collect tweets that contained 1) various affect-associated words and 2) different Arabic dialects. Enriching our data with such varieties can improve the effectiveness of the generated word embedding models to target Arabic affect in social media.

[3] `https://github.com/aialharbi/ACWE`

3.2.1 Various Affect-associated Words:

To ensure that these tweets cover a range of affect-associated words, we initially employed the English NRC lexicon (S. M. Mohammad, 2018) for a selection of 63 words[4], which represent different levels and intensity of emotions. The lexicon includes common English terms that are associated with emotions to different degrees. Each term has a real-valued affect intensity score and its corresponding emotion (e.g. anger, fear and joy). We then translated these words into Arabic using Reverso context[5], an online translation service. This tool was also used to find synonyms for the selected words, thereby extending the range of terms from 63 to 228. At this point, the collection of terms covered MSA affect words, which is a predicted outcome from English-to-Arabic translation.

3.2.2 Different Arabic Dialects:

To ensure that the collected tweets would reflect a range of dialects, we employed an MSA term list to search for dialect synonyms in two online dictionaries (Mo3jam[6] and Atlas Allhajaat[7]), which extended the term list by adding 217 new dialectical affect words. Additionally, emojis can be used as a universal language, according to Kralj Novak et al. (2015). We chose the 30 most popular emojis from the sentiment scores established in (Kralj Novak et al., 2015), and these were fed into the list of terms. Lastly, we made the assumption that any tweet from a particular Arabic-speaking country is most likely be written using a dialect of the country from which it originated. We retrieved tweets including every identified term (around 500 terms) by using the Twitter Search API and inputting the geolocations of various Arab countries.

[4] These are words that directly convey meanings of sentiments or emotions, such as *anger* and *rage*. They are not words that indirectly convey sentiments, such as *dead* and *tears*.

[5] `http://context.reverso.net`

[6] `http://en.mo3jam.com`

[7] `http://atlasallhajaat.com`

3.3 Data Pre-processing

Data collected from Twitter generally includes content that is not useful for affect classification tasks such as mentions, links and unknown symbols. This type of 'noise' has to be treated before training models in order to reduce both the noise and the size of the vector space (Q. Li et al., 2017; Singh & Kumari, 2016). In this study, we applied different pre-processing techniques to investigate their impact on affect tasks. These methods were integrated at the word embedding generation phase and the downstream task stage (classification/regression datasets). Figure 3.1 illustrates the stages and steps of applying these pre-processing methods. We studied the following pre-processing techniques, which we believe are the most important for Arabic content in social media:

- **Cleaning (*clean*):** Common text pre-processing methods, such as removal of unknown symbols, other language letters, diacritics, punctuation marks and URLs, are applied in the first instance.

- **Normalisation of letters (*norm*):** Letters that appeared in different forms in the original tweets are rendered into a single form. For example, the 'hamza' on characters {أ,إ} is replaced with {ا}, while the 't marbouta' {ة} is replaced with {ه}.

- **Elongated words (*elong*):** Social media users often repeat some letters for emphasis, such as 'happyyyyy' and 'saaad'. This non-standard writing is treated by removing the repeated characters.

- **Hashtag segmentation (*hashSeg*):** Hashtags are used to draw attention to words or phrases that are trending, such as #sad and #fun. While it is common to remove both the hash symbol and words, we removed the hash symbol but kept the words. Users sometimes express their emotions using hashtags, so it is considered useful to retain them. In addition, Arabic Twitter users typically combine multiple words as one hashtag; thus, we segmented such forms to be treated as individual words.

- **Emoji removal (*emojiRemove*):** We applied this method to remove emojis from the text. By default, emojis are retained.

- **Stemming (*stem*):** We used this technique to reduce a word to its root form. We used an open source Python toolkit for Arabic (CAMeL tools) (Obeid et al., 2020) to stem the target text.

We systematically investigated the impact of these pre-processing techniques individually. We grouped either some of them or all of them before generating word embeddings and when targeting the downstream tasks. To maximize the stability of the outcome, we carefully considered the sequence of the aforementioned pre-processing techniques. For example, hashtag segmentation has to be applied prior to stemming in order to stem individual words coming from hashtags. We used the following order to combine the above mentioned methods: *clean*, *norm*, *hashSeg*, *elong*, *emojiRemove* and *stem*.

3.4 Embedding Models

A large collection of tweets containing many Arabic affect-related words was retrieved and pre-processed to generate a language model at both character and word level. Word embeddings are learned representations of text, with words of similar meanings

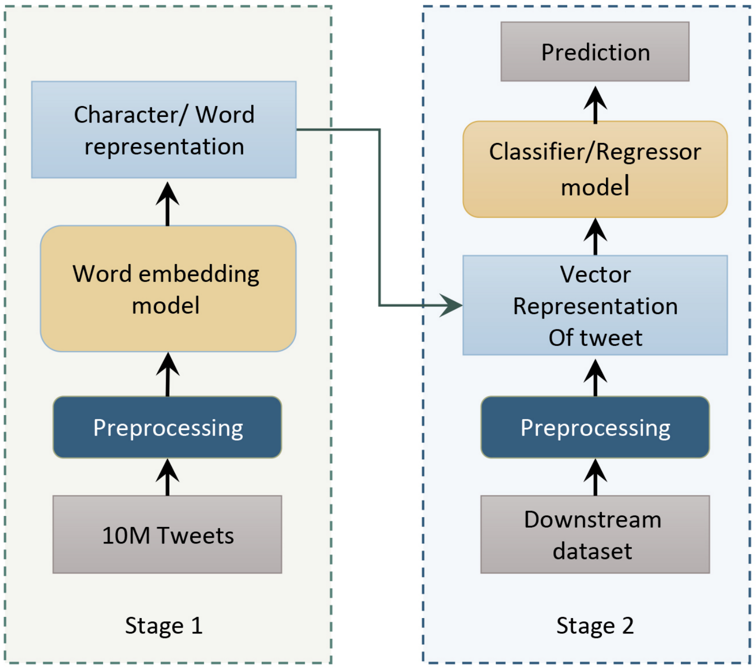


Figure 3.1: The stages and steps of applying the pre-processing techniques.

represented in similar ways. An essential element of this methodology is the concept of employing dense distributed representations for every word. Here, each word is encoded to a real-valued vector with a few hundred dimensions. Given a large corpus, there are different models and levels available for learning word embeddings. We used the word2vec model (Mikolov et al., 2013) and fastText model (Bojanowski et al., 2017) for word- and character-level embeddings, respectively. We leveraged these pre-trained embeddings as an input feature after combining them with a novel concatenation approach. These main steps are detailed in the following subsections.

3.4.1 Word-level Embeddings (WE)

We used the word2vec algorithm (Mikolov et al., 2013) to learn individual words and their embeddings from the harvested data. Word2vec adopts two learning techniques, namely, the continuous bag-of-words (CBOW) and skip-gram (SG) models. The abstract architectures of the CBOW and SG models are shown in Figure 3.2. Using a simple neural network, the SG model is trained by predicting the words surrounding a given target word and minimises the following loss function:

$$E = -log(p(\vec{w_t}) \mid \vec{W_t})) \tag{3.1}$$

where w_t represents the given word, and the words coming before and after the target word (window) are denoted by W_t. All of the inputs and outputs are of the same size and encoded with one-hot coding. The CBOW model works in a similar way, but rather than predict the context on the basis of the target word, it predicts the target word on the basis of the surrounding words.

Both models (CBOW and SG) were trained on the collection of tweets that was retrieved to create affect word embeddings. The Gensim library[8] was used to implement the word2vec models. Every tweet was assumed to represent a sentence, with the input for the word-level model being a list of pre-processed tweets that were tokenised into words. We examined different pre-processing methods to study their impact on the effectiveness of the generated models. One of the primary parameters for training the models was (window), which is the maximum distance between the target word and the surrounding context. We compared different values (3, 5 and 7) to select the parameter value that best improves the performance of the models. We also compared different vector sizes (300, 200 and 100) to study the impact of these factors on the final generated models.

3.4.2 Character-level Embeddings (CE)

The wide variation in the form of Arabic dialect words contributes to the OOV problem. Therefore, effective resources and tools are needed to better understand and treat these various linguistic forms when targeting affect tasks in Arabic tweets. We employed a character n-grams model (fastText) (Bojanowski et al., 2017) to learn the morphological

[8] `http://radimrehurek.com/gensim/models/word2vec.html`

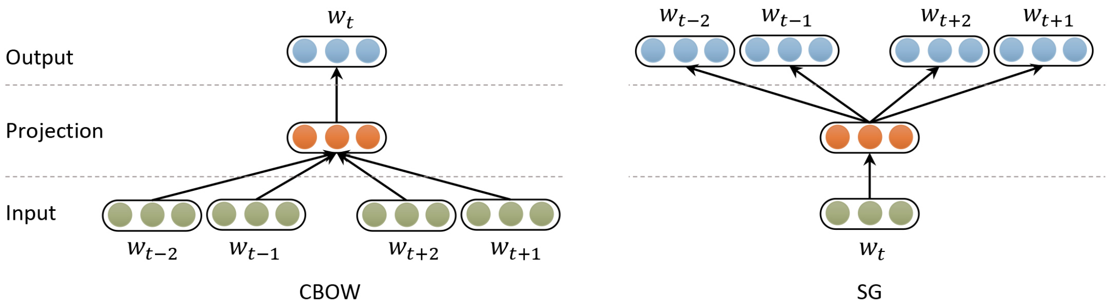


Figure 3.2: The general architecture of CBOW and SG models.

features present in each word. FastText differs from word2vec in that it can learn vectors for character n-grams. Thus, fastText can identify words that are similar in meaning but have different word formations. The input for this CE model was a composed of n-grams for each word in a given tweet. For example: the token متنرفز (mtnrfz) was composed of 2- or 3-grams as follows:

’<m’, ’mt’ , ’tn’ , ’nr’, ’rf’, ’fz’, ’z>’. 2-grams
’<mt’, ’mtn’ , ’tnr’ , ’nrf’, ’rfz’, ’fz>’. 3-grams.

The ’<’ and ’>’ are special symbols appended to indicate the token start and end. After training the model, we obtained the embeddings for all the n-grams given retrieved tweets. The word representation vector for a given token can be taken by the sum of its n-grams. Using this character-level information enabled the model to represent a rare word since it is strongly likely that some of its n-grams can be found in other words.

As in the word-level model generation, we examined different pre-processing methods to study their impact on the effectiveness of the generated models. The Gensim library[9] was employed to implement the fastText model. The input of the character-level model was a list of a bag of character n-grams for each tweet. We adopted the identical primary parameters used for WE. Additionally, in order to control character n-gram length, we examined different values of n (2 and 3).

[9] http://radimrehurek.com/gensim/models/fasttext.html

3.4.3 Affect Character and Word Embeddings (ACWE)

At this stage, we have two pre-trained models: character-level *CE* and word-level *WE*. As explained in Section 3.1, while *CE* seems to encode all variants of a word's morphology closely in the embedded space, *WE* seems to give more importance to semantic similarity. To take advantage of both models, we propose ACWE, a novel approach that aims to concatenate these two pre-trained embeddings; hence, it can be used as an input feature for a range of sentiment, emotion and related downstream tasks.

Given a tweet t_i that has a sequence of words $\{w_1, w_2, ..., w_n\}$, our goal is to morphologically and semantically represent each word in each tweet $w_i \in t_i$ as an n-dimensional continuous vector. To achieve this goal, we assume that each word $w_i \in t_i$ is represented semantically by *WE(*w_i*)* and morphologically by *CE(*w_i*)*, where *WE(*w_i*)* is the word embedding of w_i, while *CE(*w_i*)* is the character embedding of w_i. The $ACWE(w_i)$ method is used to concatenate both embeddings, and it can be obtained in the following cases:

$$ACWE(w_i) = \begin{cases} CE(w_i) \bigoplus WE(w_i), & \text{if } w_i \in (CE|V|\,,\, WE|V|) \\ CE(w_i) \bigoplus WE(\mathit{find\_alternative}(w_i)), & \text{if } w_i \notin (\,WE|V|) \\ zeros\ \ of(CE + WE)\ dimensions & \text{otherwise} \end{cases} \tag{3.2}$$

The first case is a direct concatenation of *CE(*w_i*)* and *WE(*w_i*)*, and it arises if w_i can be found in both embeddings. However, if w_i cannot be found in *WE*, we assume this is due to variants in the given word's morphology. Consequently, instead of using a vector of zeros for unseen w_i, it will be replaced by another word's morphology that can be realised by *WE*. Alternative words can be obtained using *find_alternative(*w_i*)*, which aims to find an alternative word to be represented by (w_i). Finally, if w_i cannot

Table 3.2: Examples of unseen words and the steps of how to find the alternative words.

Step	Examples of OOV from WE		
	زعلااتك zElAAtk (your upsets)	ومتحلطمه wmtHlTmh (and feel broken-feminine)	هفضحه hfDHh (will expose him)
The five most similar words using CE	زعلاان zElAAn (upset) زعلاتك zElAtk (your upsets) زعلاانه zElAAnh (upset-feminine) زعلاتي zElAty (my upsets) زعلاتك zElAtk (your upsets)	متحلطمه mtHlTmh (feel broken-feminine) ومتحلطم wmtHlTm (and feel broken) متحلطم mtHlTm (feel broken) تّحلطمه tHlTmh (his broken feeling) لحلطمه lHlTmh (will break his feeling)	هفضح hfDH (will expose) افضحه AfDHh (expose him) هفضحك hfDHk (will expose you) بتفضحه btfDHh (he will be exposed) هتفضح htfDH (will expose)
The final selected word	زعلاتك zElAtk (your upsets)	متحلطمه mtHlTmh (feels broken-feminine)	هفضح hfDH (will expose)

be determined using *CE* and *WE*, it will be represented by a vector of zeros.

find_alternative(w_i) is a method that aims to find the most similar word on the basis of 1) the cosine similarity of the w_i vector and the vectors for each word in *CE* and 2) the most similar word that shares the maximum number of letters. To identify the most similar word on the basis of the cosine similarity, we applied the (most_similar) function from Gensim to find the five most similar words. This function is used to compute the cosine similarity between the weight vectors of the given unseen word and the vectors for each word in *CE*. From these potential candidates, which are likely to be different variants of the unseen word, we select the word that shared the maximum number of characters and can be recognised by *WE*. Table 4.2 presents three examples of three unseen words that can not be found by *WE*; they are replaced using the *find_alternative(w_i)* method.

3.5 Experimental Setup

In this section, we provide information about the datasets used to evaluate our models, the official metrics for each task, and an overview of the state-of-the-art pre-trained Arabic word embeddings compared against our models and supervised learning models used word embedding models as input feature.

3.5.1 Datasets

We evaluated our models using different affect tasks in the SemEval 2018 task 1 (Affect in Tweets) datasets (S. Mohammad et al., 2018). We selected these datasets because of the variety of affect tasks and Arabic dialects present in the data. In addition, two related downstream tasks (sentiment and sarcasm classification), described below, were used to evaluate the robustness of the models. In total, we used six datasets in our experiments as follows:

- **Emotion Intensity Regression Task (*EI-reg*):** In this task, there were four sub-sets for each emotion (anger, fear, sadness and joy). When given an emotion and a tweet, the goal was to determine the emotional intensity (EI) that most accurately is expressed by the target tweet. The data contained 1800 Arabic tweets divided by three sets: a training, development (dev) and test set for each emotion. The EI-reg task was annotated by real-valus scores, ranging from zero (the lowest intensity) to one (the highest intensity).

- **Emotion Intensity ordinal classification Task (*EI-oc*):** This task is similar to EI-reg, however, it aimed at predicting EI classes ranging from 0 to 3, where 0 refers to an unrelated emotion, 1 for the lowest EI and 3 for the highest EI that can be inferred. Table 3.3 presents the details of the dataset of EI-reg and EI-oc.

- **Valence Intensity regression Task (*V-reg*):** When given a tweet, the task is to predict the valence (V) that most effectively represents the tweeter's valance

Table 3.3: Number of tweets in *EI-reg* and *EI-oc* datasets and the statistics of the datasets splits.

Task	Emotion	Labels	Train	Dev	Test	Total
EI-reg/ EI-oc	anger	0 to 1 (real-value)/ 0,1,2,3 (classes)	877	150	373	1,400
	fear		882	146	372	1,400
	joy		728	224	448	1,400
	sadness		889	141	370	1,400

Table 3.4: Number of tweets in *V-reg* and *V-oc* datasets and the statistics of the datasets splits.

Task	Labels	Train	Dev	Test	Total
V-reg/V-oc	real-value/ 7 classes	932	138	730	1,800

or sentiment using a real-value score. The V-reg task scores ranged from 0 to 1, from most negative to most positive.

- **Valence Intensity ordinal classification Task (*V-oc*):** The aim in this task is to classify a given tweet to one of the seven class labels, ranging from -3 (very negative) to +3 (very positive), where 0 indicates neutrality. Table 3.4 presents the details of the dataset of EI-reg and EI-oc.

- **ArSentiment (*ArSen*):** (Abu Farha & Magdy, 2020a) generated using a combination of SemEval's 2017 (Rosenthal et al., 2017) and ASTD (Nabil et al., 2015b) datasets. The dataset consists of 10,547 tweets, of which 8,075 were extracted from the SemEval's dataset, and the remaining 2,472 were extracted from ASTD. The extracted tweets were subsequently reannotated into three sentiment classes: positive, negative, or neutral. The tweets were composed in various Arabic dialects and were annotated using Amazon's Mechanical Turk.

Table 3.5: Number of tweets in *ArSen* and *ArSarc* tasks and distribution of classes.

Task	Label	Train	Test	Total	Class %
ArSen	Positive	1,362	316	1,678	16%
	Negative	2,813	716	3,529	33%
	Neutral	4,262	1,078	5,340	51%
ArSarc	False	7,100	1,765	8,865	84%
	True	1,337	345	1,682	16%

- **Arabic Sarcasm detection (*ArSarc*):** The *ArSen* dataset was also used to apply a new annotation that can be used to detect sarcasm. Tweets were labelled with either a sarcasm and not-sarcasm tag, where 16% were labelled as being sarcastic (1,682 tweets). Every tweet was examined and annotated by three separate annotators, who achieved an 86.7% agreement level. Table 3.5 presents an overview of the dataset size and label distribution.

3.5.2 Evaluation Metrics

For each aforementioned dataset, we followed the evaluation metric provided by the authors. The metrics used for evaluating our models over these six datasets are as follows:

- **Pearson:** Pearson's correlation coefficient aims to calculate the correlation between the score predicted by our system and the score given by the test data. Pearson is the official metric for the affect tasks (*V-oc*, *V-reg*, *EI-oc* and *EI-reg*). For EI tasks, we calculated the average (macro-average) for all four emotions to obtain the final result for each task.

- **Macro F1-score:** F1 can be interpreted as a weighted average of precision and recall, where 1 refers the best result and 0 for the worst one. Macro F1-score calculates the average of the F1 score of each class. We adopted the same official metric provided by the organisers and researchers for tasks: *ArSen* and *ArSarc*.

3.5.3 Pre-trained Word Embeddings

To evaluate the effectiveness of WE, CE and ACWE, we used three Arabic pre-trained word embeddings which are (to the best of our knowledge) the most commonly available resources released to the research community as free to use as the following:

- **Ara2Vec (Soliman et al., 2017):** Ara2Vec consists of six different word embedding models derived from different sources. These are word-level models that aim to learn word representations for general NLP tasks. We employed the model that was trained on Twitter data because our downstream tasks contained tweets.

- **Mazajak (Abu Farha & Magdy, 2019):** Mazajak is a word-level model, and it is considered the largest Arabic word-level embedding. A total of 250 million Arabic tweets were used to build this language model.

- **Altwyan (Altowayan & Tao, 2016):** They trained their model using a corpus from a variety of public text contents, most of which were news articles (150 million words) and consumer reviews (40 million words).

Table 3.6 presents a summary of important information about each of these models with their sizes and pre-trained corpora.

3.5.4 Supervised Learning Models

To evaluate the word embedding models, we incorporated them into various supervised learning framework settings for the aforementioned tasks. First, we pre-processed the datasets using the techniques described in Section 3.2. Then, we adopted different approaches to predict real-value scores for the regression tasks and class categories for classification tasks. To this end, we used and compared the following supervised learning models:

Table 3.6: Different pre-trained Arabic word embeddings used for experimental evaluation.

Model	No. of Tokens	Corpus	Size
Ara2Vec	4,347,845	General - Twitter	77M Tweets
Mazajak	1,476,715	Sentiment - Twitter	250M Tweets
Altwaian	159,175	Sentiment - Twitter	190M words
Our generated Arabic word Embeddings			
WE	626,212	Affect - Twitter	10M Tweets
CE	441,025	Affect - Twitter	10M Tweets

- **Logistic Regression:** We used logistic regression as the baseline for comparing the impact of the pre-processing techniques. We then reported the results to consider the use of the best performing methods for more advanced training models.

- **XGBoost:** The extreme gradient boosting (XGBoost) learning model (Chen & Guestrin, 2016) is a state of-the-art method in machine learning (Orzechowski et al., 2018) for a number of regression tasks. This is an algorithm of decision trees in which new trees correct the errors of trees which are already part of the model. Trees are added to the model until no further changes can be made. Regularisation is incorporated into the XGBoost algorithm to control overfitting. This model is frequently employed for different problems because it performs excellently on a wide variety range of significant challenges. In this study, we input tweet vector representations obtained from an average of real-value word vectors for every word with matching vector representations derived from the pre-trained embeddings.

3.6 Results

The results of our experiments are evaluated using Pearson's correlation coefficient and F-measure metrics based on the official metrics for each task. Our results and findings are discussed in the following subsections.

3.6.1 Effect of Pre-processing Techniques

We applied different pre-processing techniques to investigate their impact on downstream affect tasks. We compared the effect of applying individual pre-processing methods or groups thereof to both stages, as illustrated in Figure 3.1. Since raw tweets usually contain noisy data, we essentially applied clean as the default pre-processing method. The other pre-processing methods (Section 3.3) were applied besides *clean* to investigate different scenarios of the pre-processing methods.

The results of our experiments are presented in Table 3.7. For WE and CE, the use of *norm*, *elong* and *hashSeg* individually with *clean* can lead to larger improvements across all datasets compared with those of *emojiRemove* and *stem*. Emojis are an important element and convey meanings in affect tasks; therefore omitting them has a negative impact on the results. Although stemming words improved results in previous works in English and Standard Arabic, they did not show a positive effect in our experiments. We believe this is because current stem tools cannot handle Arabic dialects. This can explain why incorporation of all pre-processing methods negatively affected the performance of our models across all the datasets. In particular, application of a simple pre-processing method (*clean*) alone produced better results compared with those of the combination of all methods. Integration of *clean*, *norm*, *elong* and *hashSeg* improved the results by an average of 2.5% across all datasets. Finally, the results showed no considerable performance difference between the application of the pre-processing methods in WE and CE.

Table 3.7: Performance results for evaluating the impact of pre-processing techniques using WE and CE models cross six datasets.

Models	Pre-processing	EI-oc	EI-reg	V-oc	V-reg	Ar Sen	Ar Sarc
WE	*clean*	0.444	0.547	0.676	0.663	0.629	0.609
	clean+norm	0.461	0.541	0.700	0.679	0.619	0.611
	clean+elong	0.462	0.557	0.717	0.654	0.633	0.608
	clean+hashSeg	0.452	0.564	0.706	0.661	0.638	0.615
	clean+emojiRemove	0.442	0.534	0.635	0.603	0.613	0.586
	clean+stem	0.458	0.529	0.677	0.632	0.608	0.588
	all methods	0.443	0.524	0.682	0.626	0.610	0.596
	clean+norm+elong+hashSeg	0.512	0.554	0.712	0.686	0.637	0.615
CE	clean	0.487	0.561	0.705	0.675	0.646	0.640
	clean+norm	0.503	0.539	0.722	0.691	0.657	0.647
	clean+elong	0.495	0.557	0.713	0.678	0.657	0.647
	clean+hashSeg	0.483	0.552	0.724	0.692	0.654	0.648
	clean+emojiRemove	0.477	0.529	0.685	0.638	0.654	0.633
	clean+stem	0.479	0.489	0.708	0.663	0.630	0.621
	all methods	0.483	0.503	0.704	0.656	0.642	0.624
	clean+norm+elong+hashSeg	0.538	0.557	0.745	0.695	0.660	0.656

3.6.2 Comparison with State-of-the-Art Pre-Trained Arabic Word Embeddings

We compare five pre-trained word embeddings, namely, three open-source models and both of our generated models. In addition, we compared these models with the ACWE method. The information presented in Tables 3.8, 3.9 and 3.10 show the effectiveness of each model in the supervised framework of performing affect-sensitive tasks. From the reported results, *CE* significantly outperformed those of the other models. We believe that the main reason for this was associated with OOV problems. Although these models were trained using a massive corpus, the word-level embeddings could not capture more than 1200 words from each dataset. Nonetheless, the ACWE method improved the results by 1.3% to 5% across all datasets. This indicates the effectiveness of the proposed method and the importance of leveraging character-level and word-level embeddings in Arabic affect tasks in the context of microblogs.

Table 3.8: Pearson correlation coefficient results for our models and state-of-the-art pre-trained Arabic Word Embeddings for EI-reg task

Model	EI-reg				
	anger	fear	joy	sad	avg.
Ara2Vec	0.556	0.536	0.688	0.641	0.605
Mazajak	0.555	0.576	0.683	0.623	0.609
Altwyan	0.297	0.333	0.449	0.497	0.415
Our generated Arabic word Embeddings					
WE	0.539	0.529	0.653	0.607	0.587
CE	0.601	0.595	0.704	0.658	0.643
ACWE	**0.638**	**0.622**	**0.758**	**0.686**	**0.676**

3.6.3 Comparison with Top Systems Analysing Sentimental and Emotional Intensity Tasks

Most of the top-performing systems proposed for this shared task employed ensemble approaches to combine different machine and deep learning models. The majority

Table 3.9: Pearson correlation coefficient results for our models and state-of-the-art pre-trained Arabic Word Embeddings for EI-oc

Model	EI-oc				
	anger	fear	joy	sad	avg.
Ara2Vec	0.472	0.526	0.604	0.594	0.549
Mazajak	0.450	0.512	0.646	0.530	0.534
Altwyan	0.272	0.312	0.425	0.489	0.375
Our generated Arabic word Embeddings					
WE	0.479	0.511	0.628	0.556	0.544
CE	0.511	0.531	0.647	0.606	0.576
ACWE	**0.543**	**0.572**	**0.675**	**0.609**	**0.600**

Table 3.10: Performance results for our models and state-of-the-art pre-trained Arabic Word Embeddings for four affect-related tasks

Model	V-reg	V-oc	ArSen	ArSarc
Ara2Vec	0.773	0.723	0.665	0.638
Mazajak	0.720	0.680	0.653	0.634
Altwyan	0.515	0.535	0.569	0.524
Our generated Arabic word Embeddings				
WE	0.756	0.702	0.647	0.625
CE	0.783	0.731	0.660	0.646
ACWE	**0.818**	**0.767**	**0.671**	**0.659**

of these systems employed models based on hand-engineered features, such as the sentiment and emotional lexicons found in the Arabic language. In our work, we only used our embedding models as the input feature for XGBoost, a machine learning classifier or regressor. As shown in Table 3.11, we achieved competitive results. We outperformed the top system in the EI-oc task by 1.3% and ranked second in the remaining tasks. Our goal was not to fully address affect tasks but to demonstrate that the use of a well-generated word embedding model could yield competitive results. We will investigate other features and resources and employ more advanced learning methods to improve the results in next chapter.

Table 3.11: Pearson correlation coefficient results for our ACWE and top systems using sentimental and emotional intensity tasks

Task	1st best	2nd best	Our ACWE
Ei-reg	**0.685**	0.667	0.676
EI-oc	0.587	0.574	**0.600**
V-reg	**0.828**	0.816	0.818
V-oc	**0.809**	0.752	0.767

3.7 Discussion

Our aim, in this chapter, is to take advantage of both character- and word-level models to discover effective methods of obtaining better representations for emotional intensity in tweets in Arabic dialects. To achieve this, we built a large corpus containing a variety of affect words and Arabic dialects. We systematically compared different pre-processing techniques to examine their effect on the effectiveness of the generated word embedding models. Finally, we employed different machine and deep learning algorithms to evaluate our models using eight downstream tasks.

Our experiments with our generated models and off-the-shelf embeddings show the importance of leveraging affect-specific word embedding models as well as the ability of character-level models to overcome the OOV problem. From our observation, about 5%–10% of the words in each dataset could not be identified by the word-level embeddings. Most of these words were Arabic dialects or misspellings, which are common among user-generated text in social media.

Our experiments with different pre-processing techniques show the importance of applying simple methods to clean noisy data (such as user mentions and none Arabic letters). Moreover, emojis and hashtag words are useful and can convey valuable information for model training. Therefore, these words should be segmented instead of being removed. Although stemming words provided better results in MSA, in our work, the stem method did not improve the performance of the models.

Future research directions on Arabic affect in tweets are listed as follows.

- Given the success of contextualised word embedding models (BERT (Devlin et al., 2019) as an example) in different NLP tasks, these sophisticated models can be trained on our collected data to improve results.

- Training BERT from scratch is time consuming, and off-the-shelf models (such as AraBERT) may not perform well because such models usually are trained on MSA. Therefore, one possible direction is to enhance these large models with our generated models to target Arabic affect in tweets (Roy & Pan, 2020).

- Multi-task learning (MTL) is an approach to inductive transfer that enhances generalisation by using the domain knowledge found in similar tasks as an inductive bias. At present, most studies regard Arabic tasks as individual tasks. Exploiting the relationship between the different affect tasks may enhance findings.

3.8 Conclusion

In this chapter, we take advantage of both character-level and word-level models to discover more effective means of representing Arabic affect in tweets, which we call Affect Character and Word Embeddings (ACWE). We first trained both levels of models on a massive number of tweets, which were collected carefully to ensure that there was significant variation of affect and Arabic dialects in the words. We then employed a novel method that concatenates both levels of models to represent each word morphologically and semantically. We evaluated the effectiveness of our ACWE model by applying it only as a feature under a supervised learning, using six datasets for affect tasks and related tasks. Our method advances a state-of-the-art approach to the task of classifying Arabic emotional intensity, outperforming the top-performing systems. In addition, our method achieves better results compared to other Arabic pre-trained word embeddings. ACWE has been released to be used in pre-trained word embeddings for applications and research relying on Arabic sentiment and emotion analysis.

In the next chapter, we will apply more sophisticated algorithms (deep learning methods) to investigate their impact on the result using embeddings (ACWE). In addition, we will employ contextualised word embeddings to fully target emotional intensity task.

Chapter 4

Enhancing Contextual Word Embeddings for Arabic Emotional Intensity Tasks

In the previous chapter, we proposed ACWE enriched by the diversity of affective vocabulary words and Arabic dialects. ACWE takes advantage of the combination of character and word level to represent words morphologically and semantically. This method improved emotional intensity detection compared to other pre-trained static embeddings. Although we did not use advanced learning algorithms, we showed how the use of a well-generated, word-embedding model with simple machine learning algorithms can provide competitive results. In this chapter, we aim to improve the detection of emotional intensity by investigating advanced deep learning algorithms and contextualised language models. First, we use our generated ACWE as input to various DL approaches in order to examine the impact of using such advanced learning on our downstream tasks. Second, we investigate and evaluate the performance of six contextualised language models in emotional intensity detection. Finally, we propose a method for enhancing contextualised language models by incorporating ACWE in emotional intensity tasks, in particular, as well as other affect-related tasks. Our proposed method improves the performance of language models and achieves state-of-the-art results for eight downstream tasks.

4.1 Introduction

A number of studies have analysed sentiment classification; however, studies on alternative aspects of affect (sentiment strength and emotional intensity) are limited (S. Mohammad et al., 2018). Detecting affect from text can be challenging, particularly in the context of social media microblogs. This task becomes even more complicated when morphology-rich languages, such as Arabic, are involved (Al-Ayyoub et al., 2019). Social media communications typically consist of a range of dialects and sub-dialects that are not ruled by consistent standards. As such, there is a need for effective methods and resources that can be adopted to better comprehend and treat a variety of linguistic forms when seeking to understand affect in Arabic tweets.

Traditional, or static, word embedding has been used effectively for a range of NLP tasks (Devlin et al., 2014; J. Zhang et al., 2014; Lin et al., 2015; Bordes et al., 2014). It uses dense vectors to represent words projecting into a continuous vector space, thereby decreasing the number of dimensions. Most research on affect tasks has derived from static word embedding models, such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017). However, although these early frameworks achieved some significant advances, they lacked contextualised information. Recently, Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2019) has been shown to be able to generate effective representations for NLP tasks in various contexts. These dynamic language models provide words with different representations based on the contexts of these words. Transformer-based language models are particularly useful for Arabic sentiment and emotion classification tasks (Abu Farha & Magdy, 2021; Al-Twairesh, 2021). However, to the best of our knowledge, no work has employed contextualised word embeddings for Arabic emotional intensity and sentiment strength.

Recent research shows that the combination of static and dynamic word embeddings can benefit downstream tasks (Roy & Pan, 2020; Peters et al., 2019; Z. Zhang et al., 2019; Alghanmi et al., 2020). In our work, we propose an approach to enhancing a contextualised language model (Abdul-Mageed et al., 2021) with the integration of our generated static character and word embeddings, ACWE, (proposed in Chapter 3). We hypothesise that static embedding models trained specifically on a corpus that is rich in Arabic-affect-related words can boost the performance of language models. Furthermore, character-level embedding (CE) has been proved to overcome out-of-vocabulary (OOV) words. Our proposed method achieves state-of-the-art results for Arabic emotional intensity and sentiment strength tasks.

The rest of this chapter is organised in the following manner. The methodology for our proposed approach is explained in Section 4.2. Sections 4.2 and 4.4 provide the experimental setup and results. Finally, the outcomes and conclusion for our work are discussed in Sections 4.5 and 4.6.

4.2 Methodology

In this section, we describe the main components of our proposed method and how they work together. Firstly, we provide a brief description of the static word embeddings. Secondly, we examine four deep learning approaches fed by static word embeddings. Thirdly, we investigate and compare the performance of six contextual language models based on Arabic emotional intensity detection. Finally, we explain our proposed method, which integrates the best-performing contextual language model and our generated static word embeddings.

4.2.1 Static Word Embeddings

Word embedding is a process by which text is represented as a dense vector with sets of semantically similar words positioned near each other. Every word is encoded in a real-valued vector that consists of numerous dimensions, 300 in our case. Two static word embedding models are used in this study -WE and CE- which were discussed in detail in Chapter 3. We briefly describe these models in the following subsections:

- **WE:** We employ a pre-trained WE model generated specifically to target Arabic affect in social media. This model was trained on a massive number of Arabic tweets rich in affect-related words from a variety of dialects. It was built by using the word2vec algorithm (Mikolov et al., 2013) to learn individual words and their representations. Although we used many words to train this model, it cannot recognise every single word in Arabic informal texts. Therefore, the OOV words are expected to arise. This is one disadvantage of using a WE model.

- **CE:** Since a significant number of different dialects are spoken in Arabic, more significant OOV issues can develop with this language than with other languages. Moreover, we observe that the WE model can deliver high value in terms of semantic similarities. However, CE can more efficiently encode variations in word morphology and align them better within vector representations. Therefore, we employ a pre-trained CE model, which has been proven to perform well in Arabic affect tasks. CE was produced by training fastText (Bojanowski et al., 2017) on a dataset containing 10 million tweets. This corpus included multiple affect words (i.e. words that express emotions and sentiments of varying intensity levels) from numerous Arabic dialects.

To this end, we follow the same combination method mentioned in the previous chapter (ACWE). We combine static character and word-level models to take advantage of both of them, semantically and morphologically representing words in our Arabic affect tasks. Hence, we employ a deep learning algorithm using ACWE to set the weights of the embedding layer. In the following section, we describe four deep learning methods that are fed by ACWE as input.

4.2.2 Deep Learning Methods

This section discusses different deep learning models that are utilised with ACWE to evaluate their performance on emotional intensity detection and other affect-related tasks. We use the following models in our study:

- **CNN:** The convolutional neural network proposed by (Kim, 2014). In this deep learning architecture, the pre-trained static word embeddings (ACWE) are used to initialise the embedding layer weights. These weights are then updated during the training process to make them appropriate for the downstream tasks. Then, different filters and kernels are applied to generate features which are then max pooled. Finally, these features are passed to the fully connected softmax layer for the downstream tasks. Figure 4.1 presents the general architecture of this model.

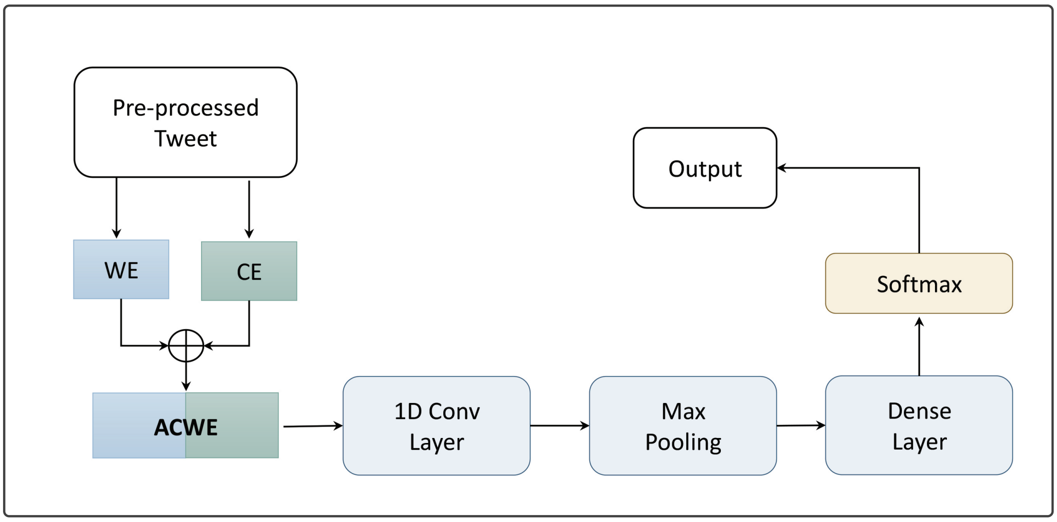


Figure 4.1: The general architecture of CNN model.

- **MG-CNN:** Another CNN architecture proposed by (Y. Zhang et al., 2016), which derives various features from different word embedding models. However, instead of passing word embedding models separately, we combine them using

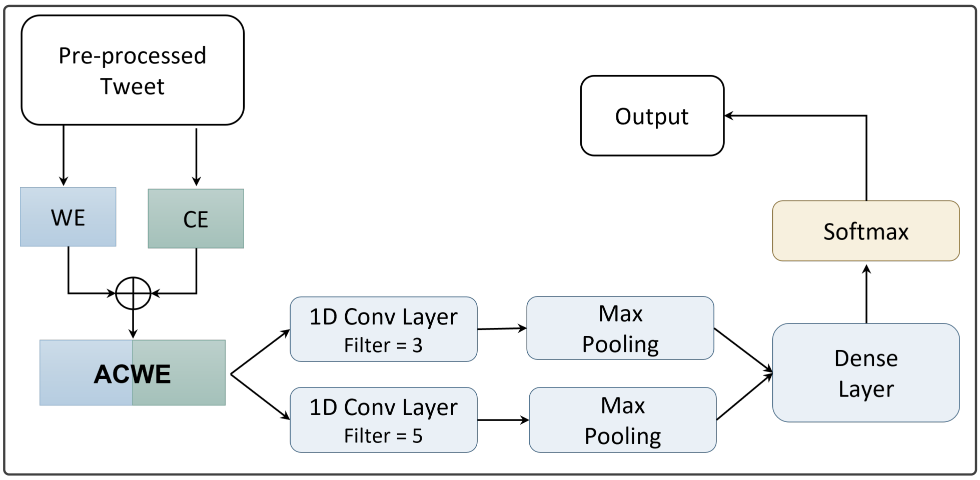


Figure 4.2: The general architecture of MG-CNN model.

our ACWE model, where they are concatenated after a multi-group of filters and max-pooling layers. We name this architecture MG-CNN and it is illustrated in Figure 4.2.

- **LSTM:** Long Short-Term Memory (Hochreiter & Schmidhuber, 1997) is an enhanced form of a recurrent neural network. We use the pre-trained embedding models to initialise the weights of the embedding layer. These weights are updated during training to fine-tune them to each task. They are connected to the rest of the layers in the network. Finally, a dense layer with one output is introduced by exploiting a sigmoid activation function. For all the other layers of the network, the ReLU activation function is utilised. We use the Adam optimiser as an optimisation function for the network. LSTM is illustrated in Figure 4.3.

- **CNN-LSTM:** A combination of CNN and LSTM proposed by(X. Wang et al., 2016). This deep learning architecture takes advantage of the most important local features extracted from the text by CNN, then these features are fed as input to LSTM for the downstream tasks. The model starts when ACWE is taken as input of a CNN model in which the convolutional layer and max pooling operations are applied to create most significant features. After these operations,

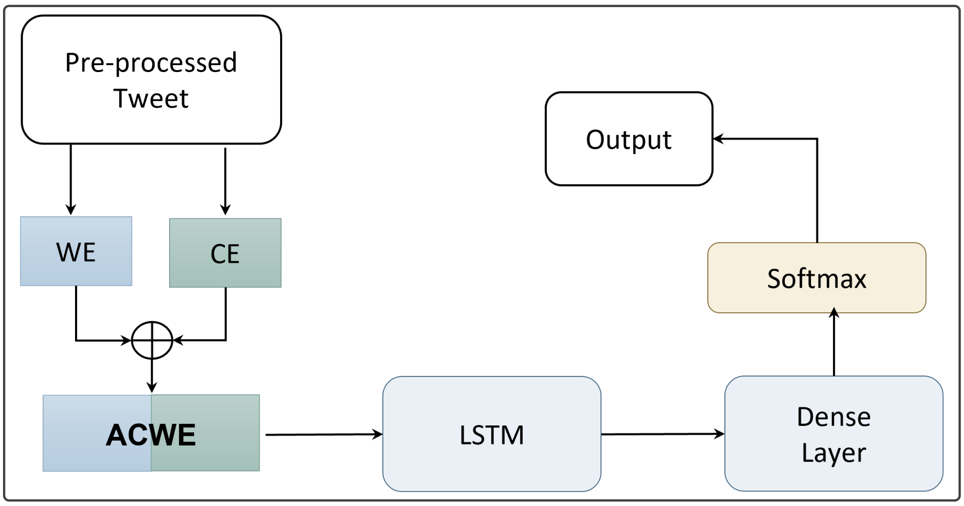


Figure 4.3: The general architecture of LSTM model.

LSTM takes the encoded features as input. The LSTM model is more capable of extracting context-dependent characteristics and generating sentence-level representations. The output text representations are then taken to a fully connected layer to obtain the final output result by using a sigmoid or MSE activation function based on the downstream task.

4.2.3 Contextualised Embedding Model

Contextualised embedding or dynamic language models are trained on a massive amount of unlabelled data in order to obtain context-aware representations, for example, ELMO (Peters et al., 2018), BERT (Devlin et al., 2019), and XLM (Conneau & Lample, 2019). These models are proven to be highly effective in a very wide range of NLP tasks. Two pre-trained multilingual models (mBERT and XLM-RoBERTa) and three monolingual models (AraBERT, MARBERT and ArabicBERT) are used in this study. The following includes a description of each model:

- **mBERT:** Devlin et al. (2019) has released a multilingual model (mBERT) to target languages other than English. This model is essentially just BERT, but instead of training on English only, mBERT has been trained on a large dataset from Wikipedia for about 104 languages, including Arabic. This model uses

BERT-based architecture, which consists of 12 attention headers, 12 encoder blocks, and 768 hidden dimensions. It can process sequences of up to 512 tokens. The number of parameters for this model is 110M.

- **AraBERT:** Antoun et al. (2020) provides a BERT-based model which is specifically built for the Arabic language. AraBERT was the first Arabic-specific BERT model to achieve competitive results on the majority of Arabic NLP tasks. Various versions of this model have been released which differ based on the size of the training corpus and pre-processing methods. AraBERTv1 was pre-trained on a large dataset containing 24 GB of text obtained from Wikipedia and a variety of news sources across the Arab world. However, the authors observed an issue related to WordPiece vocabulary for this model version. AraBERTv2 has been released to solve this problem by inserting a space around punctuation, characters, and numbers, so they would be segmented before the WordPiece vocab learning stage. In addition, they significantly extended the training corpus by three times. We use AraBERTv2 in our study.

- **ArabicBERT:** Another Arabic-specific BERT model proposed by (Safaya et al., 2020) when they participated in the shared task of Arabic offensive language and hate speech identification. The model achieved competitive results compared to AraBERT. They used 95 GB of Arabic text from different sources, including Arabic Wikipedia and the Arabic version of OSCAR, a massive multilingual corpus (Ortiz Suárez et al., 2019). They also used the same BERT-based architecture as mentioned earlier.

- **MARBERT:** Abdul-Mageed et al. (2021) released two Arabic-specific transformer-based models: MARBERT and ARBERT. These are pre-trained models used for transfer learning on Arabic dialects and MSA. Given that the dataset involved in our downstream tasks consisted of multiple Arabic dialects, we chose to use a pre-trained model that was specifically created for Arabic dialect tasks—MARBERT. Pre-trained on a vast dataset containing 6 billion tweets, MARBERT produces state-of-the-art outcomes in many tasks involving Arabic-language NLP.

- **XLM-RoBERTa (XLM-R):** (Conneau & Lample, 2019) proposed an improved model (XLM) of BERT which achieved state-of-the-art results in nine cross-lingual tasks. The XLM model uses byte-pair encoding (BPE), a pre-processing technique which splits the input text into the most frequently occurring sub-words across different languages. The model divides the input into the most frequently occurring sub-words in all languages. In order to learn relationships between words in multiple languages, XLM employs a cross-language training process with BERT in order to learn relationships between words in multiple languages. XLM-R (Conneau et al., 2020) is a model of XLM which was improved by significantly increasing the training data. It used 2.5TB of text in 100 languages from CommonCrawl data. Moreover, instead of using language-specific tokenizers, XLM-R employs a massive shared SentencePiece model to tokenise input text (Kudo & Richardson, 2018).

4.2.4 Combining MARBERT and ACWE

After going over the main components for our proposed method, 1) ACWE with deep learning models and 2) contextual language models, we explained how we integrate these models. Our hypothesis is that the best-performing contextual language model for affect tasks can be enhanced by ACWE, which is rich in Arabic affect-related words. However, we cannot directly concatenate word vectors from ACWE and the contextual language model at the word level, because the BERT and XLM models have their mechanisms to generate their token embeddings based on the context of the sentence. Therefore, we instead integrate representations from these different models at the sentence level.

We follow a similar approach to that proposed by (Peinelt et al., 2020; Alghanmi et al., 2020). However, instead of using topic models or static word-level embeddings, we incorporate our ACWE method for the static word embeddings component. As for the contextual model component, we select the best-performing model in our target tasks, which is the pre-trained MARBERT model. An overview of our proposed approach is illustrated in Figure 4.4.

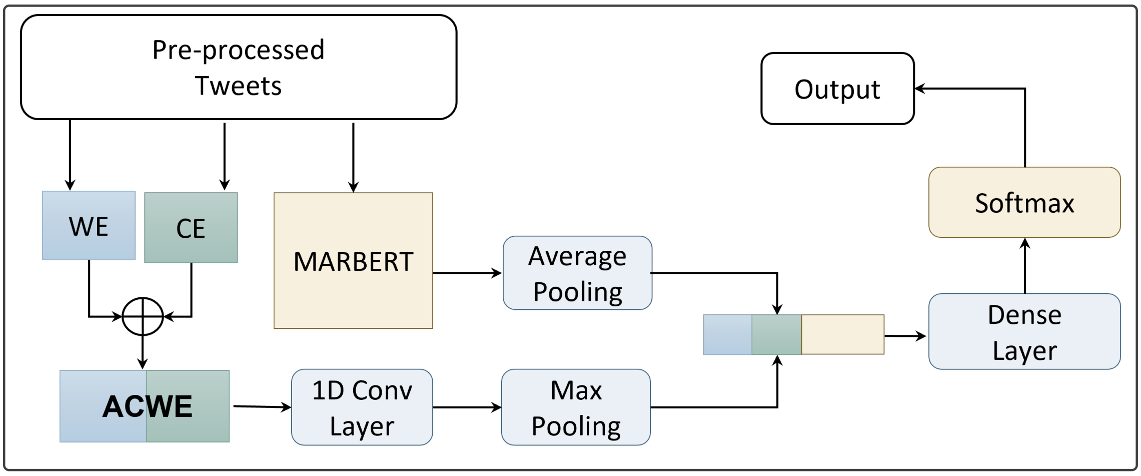


Figure 4.4: The proposed system architecture.

After fine-tuning MARBERT on the training dataset for the downstream task, we retrieve the contextualised vectors (each having an average of 12 hidden layers). For ACWE, we obtain the feature vector after the deep learning model training (we compare different deep learning architectures, as explained in Section 4.2.2). We then concatenate both obtained vectors, which are linked with the remaining network layers. Finally, after applying a dropout, the final concatenated vectors are forwarded to a dense layer with the mean square error activation function for regression tasks and softmax activation for classification tasks.

4.3 Experimental setup

In this section, we describe the datasets used, pre-processing techniques, and training setup to compare deep learning and contextual language models, and to finally evaluate our proposed method.

4.3.1 Datasets

We use emotion intensity regression and classification datasets for our main target task in this study (emotional intensity detection). In addition, to evaluate the robustness of the proposed model, we use several related downstream tasks: sentiment analysis, valence intensity, emotion classification, and sarcasm detection. In total, ten datasets are used in our experiments, which are as follows:

- **Emotion Intensity Regression Task (EI-reg):** The aim is to identify the score value of the emotional intensity expressed within a given tweet. The dataset was described in detail earlier in Section 3.5.1.

- **Emotion Intensity ordinal classification Task (EI-oc):** This task is similar to EI-reg. However, its main objective is to predict one of four intensity classes. The dataset was described in detail earlier in Section 3.5.1.

- **Valence Intensity regression Task (V-reg):** The aim of the task is to use a real-value score to predict the sentiment strength or valance represented within a given tweet. The dataset was described in detail earlier in Section 3.5.1.

- **Valence Intensity ordinal classification Task (V-oc):** The goal of this task is to classify a given tweet using one of seven class labels, which range from -3 (very negative) to +3 (very positive). The dataset was described in detail earlier in Section 3.5.1.

- **ArSentiment (*ArSen*):** (Abu Farha & Magdy, 2020a) combined SemEval's 2017 (Rosenthal et al., 2017) and ASTD (Nabil et al., 2015b) datasets. The tweets have been re-annotated into three sentiment classes: positive, negative, or neutral. The dataset was described in detail earlier in Section 3.5.1.

Table 4.1: Number of tweets in *ArSen-v2* and *ArSarc-v2* datasets and thier distributions of classes.

Task	Label	Train	Test	Total	Class %
ArSen-v2	Positive	2,180	575	2,755	18%
	Negative	4,621	1,677	6,298	41%
	Neutral	5,747	748	6,495	42%
ArSarc-v2	False	10,380	2,179	12,559	81%
	True	2,168	821	2,989	19%

- **Arabic Sarcasm detection (*ArSarc*):** *ArSen* was also used to apply a new annotation that could be used to detect sarcasm. Tweets were labelled with either a sarcasm and not sarcasm. The dataset was described in detail earlier in Section 3.5.1.

- **ArSentiment-v2 (*ArSen-v2*):** Abu Farha et al. (2021) released a new and larger dataset containing 15,548 tweets. They combined tweets from ArSen (10,547 tweets) and DAICT (Abbes et al., 2020), which is a dialectal Arabic irony corpus retrieved from Twitter. They asked the annotators to provide one of the sentiment labels (positive, negative or neutral) for each tweet, as well as which Arabic dialect the tweet belonged to. Table 4.1 presents an overview of the dataset size and label distribution.

- **ArSarc-v2 (*ArSarc-v2*):** The ArSen-v2 dataset was also used to apply a new annotation for the sarcasm detection task. Tweets were labelled with either a 'sarcasm' or 'not sarcasm' label, where 19% were labelled as being sarcastic (2,989 tweets). Every tweet was examined and annotated by three separate annotators, who achieved an 87.3% agreement level. Table 4.1 presents an overview of the dataset size and label distribution.

- **Emotion Classification (*E-c*):** The aim of the task was to identify the emotion label expressed within a tweet. Following the same experimental dataset

Table 4.2: Number of tweets in *E-c* dataset and its distribution of classes.

Emotion	Train	Test	Total	Class %
Anger	821	214	1,035	26%
Fear	728	166	894	23%
Joy	938	249	1,187	30%
Sadness	675	162	837	21%
Total	3,162	791	3,953	100%

setup proposed by (Al-Twairesh, 2021), we used EI-oc dataset for this task. We combined the four emotion sub-datasets and provided each tweet with its corresponding emotion. We removed tweets that were labelled with 'no emotion can be inferred'. As a result, we obtained 3953 tweets labelled with associated emotions (anger, joy, sadness and fear). To make sure both training and test sets had the same proportion of tweets for each emotion, we used the stratified train-test split. Table 4.2 presents an overview of the dataset size and label distribution.

- **Arabic Sentiment Analysis Dataset (*ASAD*):** ASAD, publicly released by (Alharbi et al., 2021), is the largest sentiment analysis of Arabic tweets (95K tweets) to date. Each tweet was annotated with one of the three classes (positive, negative, or neutral). The dataset contains different Arabic dialects, as well as MSA. Tweets were annotated by 69 Arabic speakers from various countries to reduce the possibility of bias in the annotation process. The reliability of the annotators was 0.83, evaluated by the average inter-rater agreement. Table 4.3 presents an overview of the dataset size and label distribution.

4.3.2 Pre-processing

We used pre-processing techniques shown to be more effective in affect tasks, as explained in Section 3.6.1. We started by removing unrecognisable symbols and any character that is not useful or used in Arabic, such as diacritics and punctuation marks.

Table 4.3: Number of tweets in *ASAD* dataset and its distribution of classes.

Label	Train	Test1	Test2	Total	Class %
Positive	8,821	3,150	3,244	15,215	16%
Negative	8,820	3,252	3,195	15,267	16%
Neutral	37,359	13,598	13,561	64,518	68%
Total	55,000	20,000	20,000	95,000	100%

We did not remove the emojis because they are often of value in sentiment and emotion analysis tasks. Moreover, we normalised letters that appeared in different forms and re-rendered them in a single expression. For instance, the 'hamza'on characters (إ , أ) was replaced with the (ا), and the 't marbouta' (ة) was replaced with (ه).

4.3.3 Models Training Setup

We used Google Collab with GPU to conduct all experiments. For deep learning models, we used the Keras library and Tensorflow. The hyper-parameters selected for the experiments were presented in Table 4.4. As for transformer models (BERT and XLM), we employed the implementation designed by the huggingface transformers library. We fine-tuned the models using a learning rate of 5e06 with 4 epochs. The maximum input length was restricted to 64 tokens.

Table 4.4: Hyper-parameters utilised for deep learning models.

Parameters/Models	CNN	MG-CNN	LSTM	CNN-LSTM
No. Filters	300	200	-	300
Filter size	3	3 - 5	-	3
Hidden units	256	64	128	128
Recurrent dropout	-	-	0.2	0.2
Output dropout	-	0.5	0.2	0.2

4.4 Results

In this section, we report our results from three main experiments. First, we compare how employing ACWE changes the results for four different deep learning methods. Second, we compare the differences in the performance of six contextual language models by fine-tuning them on emotional intensity detection datasets. Finally, we present the results of our proposed method (enhancing contextual language models with ACWE) using our main target task (emotional intensity detection) and eight task-related datasets.

4.4.1 ACWE with Deep Learning Algorithms

We conducted experiments to compare the performance of using ACWE as input embeddings into different deep learning approaches. As a baseline, we used ACWE as input features for two machine learning methods: logistic regression and XGBoost. Table 4.5 presents the experiments' results on the downstream datasets using the official metrics for each task.

From the reported results we can observe that machine learning algorithms achieved the highest result in emotional and sentiment intensity tasks. However, this is not the case for *ArSen* and *ArSar* tasks where deep learning models achieve the best results. As seen in Table 4.5, the deep leaning methods performed poorly in the *EI* and *V* tasks compared with machine learning algorithms. These results align with previous studies (Badaro, El Jundi, et al., 2018). The explanation for this limitation is that DL models need more training data to learn than is available from the emotional intensity datasets. In fact, the number of tweets in the training set for *EI* is only 800, compared to 8,400 for *ArSen* and *ArSar*.

Table 4.5: Performance results for using ACWE as an input feature in varied machine and deep learning algorithms cross six datasets.

Algorithm	EI-oc	EI-reg	V-oc	V-reg	ArSen	ArSarc
XGBoost	**0.600**	**0.676**	**0.767**	**0.818**	0.659	0.659
CNN	0.278	0.313	0.458	0.488	0.644	0.676
LSTM	0.568	0.640	0.648	0.691	**0.671**	**0.688**
MG-CNN	0.448	0.505	0.690	0.736	0.636	0.664

4.4.2 Contextual Language Models Comparisons

Since experiments with language models consume time and need powerful GPUs, we only compare those models based on our main task which is emotional intensity detection for both classification and regression tasks. To the best of our knowledge, this is the first work attempting to employ contextual embeddings for emotional intensity tasks. Tables 4.6 and 4.7 show pearson correlation coefficient results by finetuning on two multilingual models and three monolingual models for emotional intensity regression and classification tasks.

Table 4.6: Pearson correlation coefficient results for finetuning six language models on EI-reg dataset.

Pre-trained Language Model	EI-reg				
	anger	fear	joy	sad	avg.
mBERT	0.322	0.385	0.478	0.351	0.384
XLM-r	0.273	0.214	0.350	0.253	0.273
AraBERTv1	0.583	0.681	0.704	0.684	0.663
AraBERTv2	0.604	0.695	0.732	0.707	0.684
ArabicBERT	0.520	0.577	0.629	0.635	0.590
MARBERT	**0.690**	**0.706**	**0.784**	**0.706**	**0.721**

Table 4.7: Pearson correlation coefficient results for finetuning six language models on EI-oc dataset.

Pre-trained Language Model	EI-oc				
	anger	fear	joy	sad	avg.
mBERT	0.327	0.206	0.272	0.312	0.279
XLM-r	0.291	0.153	0.249	0.319	0.253
AraBERTv1	0.372	0.423	0.524	0.547	0.467
AraBERTv2	0.406	0.455	0.544	0.585	0.498
ArabicBERT	0.387	0.373	0.533	0.561	0.464
MARBERT	**0.560**	**0.644**	**0.689**	**0.656**	**0.637**

We can observe from the reported results that monolingual models obtain significantly better results compared to multilingual models. Although mBERT and XLM-r were trained on massive training datasets, the size of vocabulary for Arabic is significantly small compared to monolingual models. Unlike similar languages that can be enhanced by shared representations in multilingual models (Conneau et al., 2020), this is not the case for Arabic as it shares very few representations with other resource-rich languages due to its highly different syntactic and morphological structures. Figure 4.5 presents how the Arabic vocabulary size differs significantly in the language models used in our study.

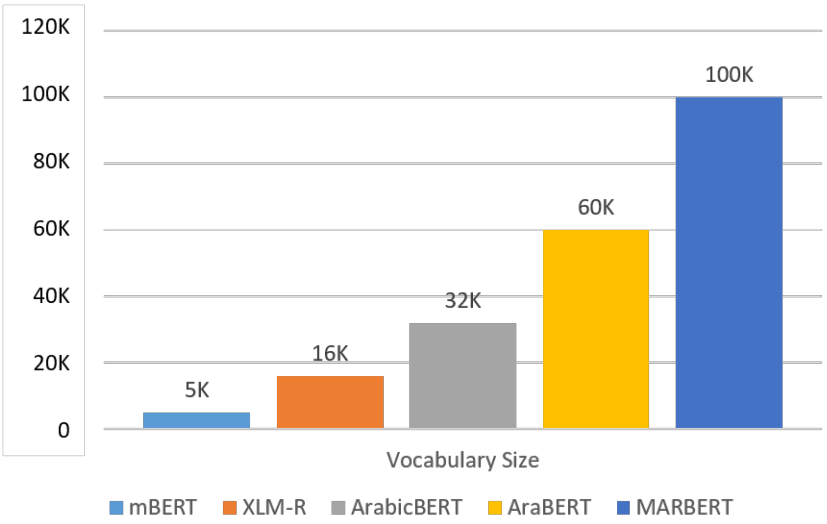


Figure 4.5: The Arabic vocabulary size of the multilingual and monolingual models used in our experiments.

As for monolingual models, MABERT obtained the best result among all emotions for both tasks as can be seen in Tables 4.6 and 4.7. MARBERT outperforms other models by a range from 4% to 20% for the average of all emotions in both tasks. The source and size of the training data have a significant impact on the performance. MARBERT was trained on a massive number of tweets, while AraBERT and ArabicBERT were trained mainly on MSA contents. As a result, these models underperform for our target tasks which are tweets rich with dialectal Arabic.

Additionally, it can be observed that prepossessing and careful tokenising of Arabic words are important to be applied to the training data before generating BERT models. Although AraBERTv1 and AraBERTv2 share the same architecture, AraBERTv2 obtains better results. Antoun et al. (2020) improved AraBERTv2 by splitting some numbers and characters before applying the WordPiece tokeniser. This indicates the impact of preprocessing, particularly the tokenising stage, on generating pre-trained language models.

Comparing the contextual language models and ACWE (reported in Table 4.5), we observe that ACWE outperforms all multilingual and monolingual models except MARBERT. This indicates that although contextual language models are extremely large, the nature of the texts in which models are pre-trained remains the most important factor.

4.4.3 The Proposed Method: Combining ACWE and Contextual Language Models

As a baseline, we used MARBERT to evaluate whether its performance could be improved by incorporating ACWE. We conducted a series of experiments to report the results of these models (the baseline and proposed approaches) by averaging them over five runs for each task. Tables 4.8 and 4.9 present the results for our main target tasks (*EI-reg* and *EI-oc*), while Table 4.10 show the results for the other affect-related tasks to assess the robustness of our proposed method.

As for *EI-reg* and *EI-oc*, we observed that the Pearson correlation result for MARBERT significantly outperformed that of the static ACWE model. Moreover, the proposed method (ACWE+MARBERT) revealed an improvement from about 1% to 3% across all four emotions. This result demonstrates the performance of the proposed method and the importance of leveraging contextual and static word embeddings in emotional intensity tasks within the context of social media microblogs.

Regarding the other affect-related tasks, ACWE+MARBERT showed an improvement and outperformed previous state-of-the-art from about 0.7% to 3.3% in V-reg, V-oc, ArSen, ArSarc and ASAD datasets. For ArSarc-v2, the best result was achieved in our proposed work in Chapter 5, which is an evolution of our proposed method by adding a multi-task learning component. Our proposed method in this chapter obtained the second-best result. As for ASAD, our proposed method obtained a competitive result (second place) with a small difference (F1-score of 0.2%) compared to the best system (Alharbi et al., 2021). In their work, they proposed an ensemble approach that computed the average predictions of five BERT models. For ArSen-v2, our proposed method obtained a competitive result (second place) compared to the best system (El Mahdaouy et al., 2021). They used MARBERT to obtain sentence embeddings, which hence were fed into two attention layers for knowledge sharing between sentiment and sarcasm tasks.

Overall, ACWE+MARBERT improved effectiveness across all four affect tasks. The enhanced results can be explained by the fact that ACWE trained on a large corpus that was built specifically for the domain of affect tasks, whereas MARBERT trained on an enormous dataset for a general domain. Therefore, the combination approach enhances the quality of the final representations with additional information. A recent study (Schick & Schütze, 2020) also found that contextualised language models still struggle to understand rare words even though they are trained with vast data. Thus, leveraging these models with a combination of character and word embeddings can enhance performance across all affect tasks.

Table 4.8: Performance results for EI-reg task using MARBERT alone and the proposed method against state-of-the-art

Model	EI-reg				
	anger	fear	joy	sad	avg.
Previous state-of-the-art	64.7	64.2	75.6	69.4	68.5
MARBERT	69.0	70.6	78.4	70.6	72.1
ACWE+MARBERT	**69.9**	**70.9**	**79.9**	**71.1**	**73.0**

Table 4.9: Performance results for EI-oc task using MARBERT alone and the proposed method against state-of-the-art

Model	EI-oc				
	anger	fear	joy	sad	avg.
Previous state-of-the-art	55.1	55.1	63.1	61.8	58.7
MARBERT	56.0	64.4	68.9	65.6	63.7
ACWE+MARBERT	**57.4**	**67.8**	**70.9**	**66.6**	**65.7**

4.5 Conclusion

Our main aim in this chapter is to improve the performance results of emotional intensity detection. We first start our experiments by investigating the performance of four deep learning models that are fed by ACWE. We find that these sophisticated models cannot boost the performance of emotional intensity tasks compared to ML algorithms. The only explanation is that DL models need more training data to learn, which is not the case for the emotional

Table 4.10: Performance results for eight affect tasks using MARBERT alone and the proposed method against state-of-the-art

Task	Evaluation Metrics	state-of-the-art	MARBERT	ACWE+ MARBERT
V-reg	Pearson	82.81 [1]	85.43	**86.12**
V-oc	Pearson	80.94 [1]	81.21	**83.13**
ArSen	F1-score	71.50 [2]	73.21	**74.72**
ArSarc	F1-score	76.30 [2]	74.17	**76.97**
ArSen-v2	F-PN	**74.80** [3]	72.13	74.33
ArSarc-v2	F1-sarc	**62.25** [4]	58.72	61.34
E-c	F1-score	76.00 [5]	76.13	**77.11**
ASAD	F1-score	**79.24** [6]	78.10	79.03

[1] (Jabreel & Moreno, 2018)
[2] (Abdul-Mageed et al., 2021)
[3] (El Mahdaouy et al., 2021)
[4] Our proposed multi-task method described in Chapter 5
[5] (Al-Twairesh, 2021)
[6] (Alharbi et al., 2021)

intensity datasets. This limitation leads us to explore contextual language models and evaluate them on emotional intensity datasets—the second main experiment. To the best of our knowledge, this is the first study attempting to employ contextual embeddings for emotional intensity tasks. Generally, monolingual models outperform multilingual BERT and XLM models. This outcome confirms that these contextual models can also provide outstanding performance in emotional intensity, as demonstrated in previous studies on sentiment polarity or emotion classification tasks.

We end our experiments proposing a method to enhance the best performed contextual models by incorporating ACWE. The integration of these different text representations indicates an improvement, particularly in our main task (emotional intensity detection). We obtain outstanding results, significantly outperforming previous state-of-the-art methods, in emotional intensity and sentiment strength tasks. We also obtain competitive results in other affect-related tasks. Training BERT on large datasets from scratch is time-consuming and requires high-performance hardware that is not always available. Alternatively, static word embedding algorithms can be trained for the domain and then combined with pre-trained BERT models.

In future work, we will explore more advanced approaches to combine static and dynamic language models. Moreover, given the fact that the training set for emotional intensity is small, we will investigate data augmentation strategies to enable deep neural network learning from more samples during the training stage.

Chapter 5

Offensive Language Detection in Arabic Tweets

In the previous chapter, we presented a combination approach to enhance contextualised language models by integrating our generated static embeddings (ACWE). Our proposed method (BERT-ACWE) achieves state-of-the-art results for emotional intensity detection and other affect-related tasks. In this chapter, we aim to improve the performance of the offensive language task by using transfer learning approaches. First, a combination of pre-trained static word-embedding and contextual-language models is used as a form of transfer learning. Several pre-trained models are compared to find the best-performing candidates for our integration method, BERT-ACWE. Additionally, emotional intensity and other emotion-related tasks are leveraged as a feature transfer learning method. Our proposed transfer learning method achieves state-of-the-art in the offensive language detection task.

5.1 Introduction

While microblogging platforms can be used positively and productively, they may also be employed for destructive purposes, such as circulating offensive messages. Users who wish to spread insults can use these channels to quickly and easily reach millions of people at the click of a button. Such occurrences of online abuse have caused emotional and psychological health concerns for users, leading to varying degrees of reaction that range from account deactivation to instances of self-harm and suicide (Kelly et al., 2018; Hinduja & Patchin, 2019; Kumar et al., 2020). To prevent this spread of negativity, there is a need for systems that can automatically identify messages that contain harmful content.

Generally, content can be considered offensive if it contains either unacceptable language (profanity) or hurtful comments that target specific individuals or groups of people (Zampieri et al., 2019). Recent research has been published on associated forms of offensive language such as cyberbullying, hate speech, abuse and aggression. The majority of studies and datasets created to date are focused on English, and few studies have been conducted on offensive language detection in Arabic (Mubarak & Darwish, 2019a).

To automatically detect offensive language, the NLP community has experimented with a variety of techniques. One approach to identifying harmful content relies on filtering texts based on offensive lexicons. This method is ineffective due to the complexity of Arabic morphology, especially in various dialects. Moreover, the offensiveness of a word is highly dependent on context. This has encouraged researchers to explore more advanced techniques, such as the use of word embeddings with deep neural networks, character-level deep learning classifiers (Mubarak & Darwish, 2019b), and more recently, transformer-based models. The majority of proposed methods target offensive language identification as a single task. A few studies attempt to take advantage of similar tasks such as hate speech and offensive language. Other works propose methods to involve sentiment and emotion classification (Plaza-del Arco et al., 2021; Rajamanickam et al., 2020) in a multi-task learning framework to detect abusive content. However, to the best of our knowledge, there is no existing research that aims to incorporate emotional intensity into the offensive language detection task.

In this Chapter, we hypothesise that transferring features from anger intensity and other affect-related tasks can be useful for the offensive language task. Moreover, there is a need to incorporate multi-level pre-trained language models to include varied forms of words used in different dialects. Thus, we will integrate different levels of word embedding models and transfer learning from anger intensity and other sources of affect-related tasks. Our proposed ensemble transfer-learning approach enhances the performance of offensive language detection and achieves state-of-the-art results.

The remainder of this chapter is organized as follows: Section 5.2 describes the offensive language dataset and explores the correlation between offensive content and affect-related tasks. Section 5.3 presents the details of the proposed method. Section 5.4 includes the presentation and discussion of the results. Finally, Section 5.5 summarises our work and suggests directions for future research.

5.2 Data

In this section, we describe the dataset used for offensive language detection. We then analyse the correlation between offensive language and sentiment analysis, as well as between emotion classification and emotional intensity.

5.2.1 Dataset Description

The dataset released by (Mubarak et al., 2021) constitutes 10,000 tweets that have been retrieved using the Twitter API and filtered by setting the language to Arabic. To ensure that a larger number of offensive tweets was collected, they used an Arabic vocative particle (yA) vastly observed in offensive content on social media. When users seek to offend an individual or group, they often use (yA) before the offensive words. Unlike collecting tweets based on a specific abusive word list, this method can generalise and expand topics, targets, and dialects. This is the largest dataset currently publicly accessible for an Arabic offensive language task. The distribution of the targeted classes was unbalanced: 19% of the tweets were labelled as offensive *OFF*, while the remaining were labelled as inoffensive *Not-OFF*. Native speakers who are familiar with several Arabic dialects carefully annotated the tweets. Using Fleiss's Kappa coefficient, the Inter-Annotator Agreement (IIA) between the annotators was 0.92. An overview of the dataset is provided in Table 5.1.

Table 5.1: Number of tweets in offensive language dataset and distribution of classes.

Label	Train	Test	Total	Class %
OFF	1,589	402	1,991	20%
Not-OFF	6,411	1,598	8,009	80%
Total	8,000	2,000	10,000	100%

5.2.2 Analysis of Offensive-Affect Correlation

In this study, we assume that users who write offensive language content tend to use emotional expressions, especially negative ones. Therefore, in this section, we first analyse the tweets in the offensive language task and their relationship to different emotions. Due to the very high cost of manually re-annotating these tweets with their corresponding emotion or sentiment, we use our proposed method (BERT-ACWE) in Chapter 4 to automatically classify the offensive language tweets.

Figures 5.1, 5.2, and 5.3 illustrate the sentiment, emotion, and anger intensity distribution in the offensive language detection dataset. In general, the negative emotion and high anger intensity constitute the majority for the offensive class (*OFF*), while neutral emotion and low anger intensity are assigned for the non-offensive class (*Not-OFF*). It can be observed that around 70% of tweets from *OFF* are assigned as negative sentiment, while only 8% of *Not-OFF* constitutes negative sentiment. Likewise, 80% and 76% of *OFF* tweets are labelled as anger and high anger intensity respectively, compared to only 10% and 6% of *Not-OFF* consisting of anger and high anger intensity. These results reveal a strong association between using emotional intensity and offensive language expressed in tweets. Therefore, exploiting this correlation is more likely to improve offensive language detection performance.

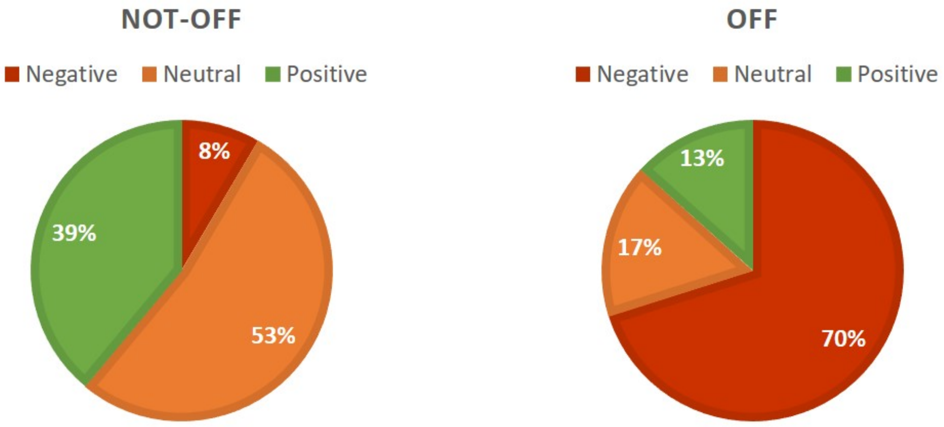


Figure 5.1: Distribution of sentiment in the offensive language detection dataset.

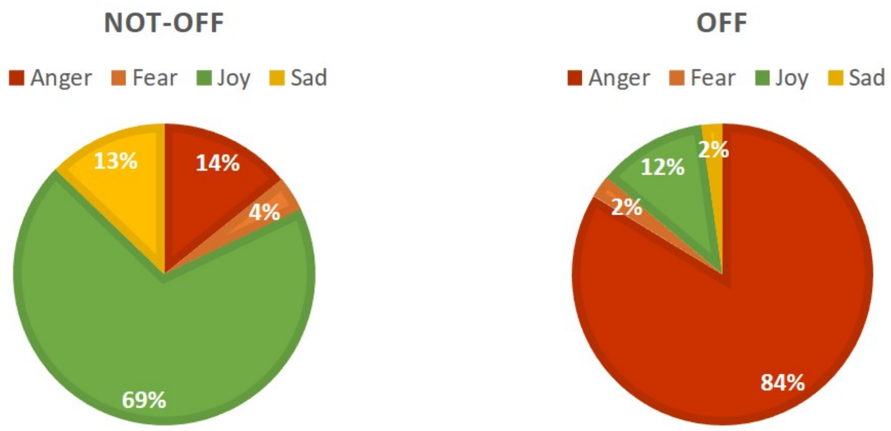


Figure 5.2: Distribution of emotion in the offensive language detection dataset.

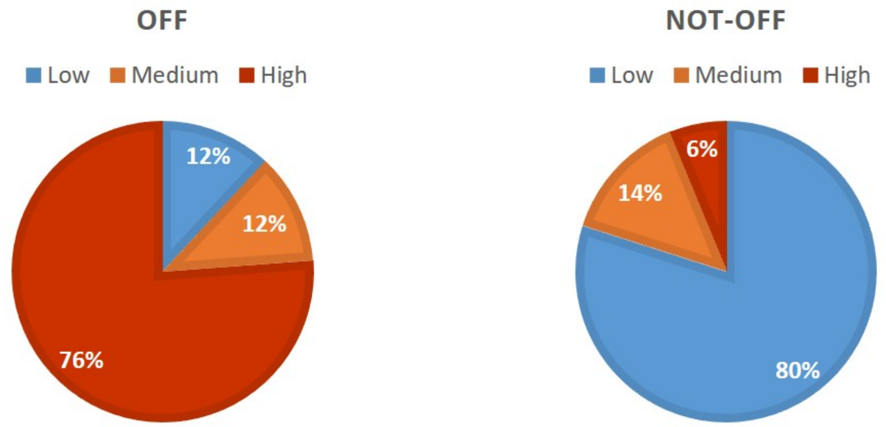


Figure 5.3: Distribution of anger intensity in the offensive language detection dataset.

Table 5.2 lists some examples of *OFF* tweets that are labelled as positive emotions (first three tweets), and the opposite when *Not-OFF* tweets are labelled as having negative emotions (last three tweets). It can be observed clearly from examples 1, 2, 3, and 6 that sarcasm has a significant impact leading to classification errors. Sarcasm is a challenging task for classifiers in many NLP tasks (Abu Farha et al., 2021). In addition, tweet 4 shows an example of annotation error where it should be labelled as *OFF* class.

5.3 Methodology

The proposed ensemble system consists of three different classifier models. We pre-processed the raw tweets as inputs (see Section 3.1) to the models. Sections 5.3.2, 5.3.3, and 5.3.4

No.	Tweet	SA	EC	EI	OFF
1	أنت يا عنصرى يا قذر يا قمر كل سنة و أنت طيب O racist O dirty O moon every year and you are good!	pos	joy	low	OFF
2	لما الاكس تنزل صوره حلوه ايه الحلاوه يا سكره يا بتلوسخه When my ex uploads a nice picture what is the sweetness O sweety O dirty girl	pos	joy	low	OFF
3	تكلم عن اليوم من الأغنى يا حلاق يا راعي كباب Talk about today who is the richest O barber O kebab man	pos	joy	low	OFF
4	يا حمد الله يا حمد الله والله انك اخس خلق الله O Hamdallah O Hamdallah by God you are the worst of God's creation	neg	anger	high	Not OFF
5	اللهم يا الله انتقم ممن سمح وسهل وحمى هؤلاء الابالسه Oh God, O God, take revenge on those who allowed facilitated and protected these demons	neg	anger	high	Not OFF
6	لما تاكل الجزر الصوت اللي يطلع ترا صوت عظامها هي تتكسر يا قاسي يا شرير يا قاتل الجزرات When you eat carrots the sound that comes out is the sound of their bones breaking O cruel O evil O killer of carrots	neg	anger	high	Not OFF

Table 5.2: Examples of sentimental and emotional classification errors

describe the three different models, which are then combined using an ensemble technique, as described in Section 5.3.5.

5.3.1 Preprocessing

Pre-processing was undertaken following a procedure used previously by several researchers (Abu Farha & Magdy, 2019; Duwairi & El-Orfali, 2014). Firstly, any unknown symbols or other characters were removed (e.g., letters from other languages, punctuation, diacritics, etc.). However, emojis were retained. We also normalised several letters which appeared in different forms in the original tweets; these were rendered into a single form. For example, the 'hamza' on characters (أ,أ) was replaced with the (ا), and the 't marbouta' (ة) was replaced with (ه).

In addition, we noted that one of the most frequent ways of using offensive words in Arabic is to begin a phrase with (يا - ya), followed by the offensive word. Many social media posters do not insert a space inside this phrase, so it can be identified as a single word. This is a problem that even the most state-of-the-art tools, such as Farasa (Abdelali et al., 2016) and MADAMIRA (Pasha et al., 2014), cannot treat. We dealt with this problem by employing RegEx to split any strings beginning with (ya) into two words. This method requires further improvement in order to deal with words like 'Yasmine' or 'Yafa'.

5.3.2 Static Word Embeddings Model (SWE)

One central method that has been recently applied to NLP tasks is word embeddings (Devlin et al., 2014; J. Zhang et al., 2014; Lin et al., 2015; Bordes et al., 2014). Word embeddings are dense vector representations of text, which capture semantic similarity between words as proximity within the vector space. We examined four pre-trained word embedding models, which are detailed in the following subsections. A summary of the important information about each of these models, including their sizes and pre-trained corpus, is presented in Table 5.3.

Model	Level	Corpus	Size	Dimensions
Ara2Vec	Word	General	77M tweets	300
Mazajak	Word	Sentiment	250M tweets	300
WE	Word	Emotional Intensity	10M tweets	300
CE	Character	Emotional Intensity	10M tweets	300

Table 5.3: Different pre-trained Arabic word embeddings used for our system

- **Ara2Vec (Soliman et al., 2017):** Ara2Vec consists of six different word embedding models derived from different sources. These are word-level models that aim to learn word representations for general NLP tasks. We selected the model that was trained on Twitter data as our target tasks contained tweets.

- **Mazajak (Abu Farha & Magdy, 2019):** Mazajak is a word-level model, and it is considered the largest Arabic word-level embedding. A total of 250 million Arabic tweets were used to build this language model.

- **WE:** This model was trained on 10M Arabic tweets that are rich in affect-related words from a variety of Arabic dialects. It was built by using the word2vec algorithm (Mikolov et al., 2013) to learn individual words and their representations. WE was discussed in detail in Chapter 3.

- **CE:** Since a significant number of different dialects are spoken in Arabic, more significant OOV issues can develop with this language than with other languages. Therefore, we employ a pre-trained CE model, which has been proven to perform well in Arabic affect tasks. CE was produced by training fastText (Bojanowski et al., 2017) on a dataset containing 10 million tweets. This corpus contains a variety of emotional and sentimental words in different Arabic dialects. CE was discussed in detail in Chapter 3.

Table 5.4 shows an example of offensive Arabic words, where the similarity of these words is mostly based on morphology for the character-level and semantics for the word-level models. This supports our intuition that while word-level embeddings seems to give more importance to the semantic similarity, character-level embeddings are more likely to encode all variants of a word's morphology closer in the embedded space. Therefore, combining these two different levels of embeddings into a supervised learning framework for the task of detecting offensive tweets can improve the results.

To this end, we use aforementioned word embbedings models separately or by combining them as multi-level embbedings (character- and word- level models). These pre-trained embbedings are fed then into four machine and deep learning algorithms (namely SVM, XGBoost, CNN and LSTM) to detect offensive language detection.

5.3.3 Contextualised Language Models (CLM)

Because the meaning of some offensive words depends on the context, contextualised language models would be helpful in this task. Transformer models such as BERT (Devlin et al.,

Example of an offensive query term: (mnHT)منحط	
Mazajak	Our char-level model
(wmnHT) ومنحط	(wmnHT) ومنحط
(q*r) قذر	(mnHTh) منحطه
(wq*r) وقذر	(wmnHTh) ومنحطه
(mtxlf) متخلف	(AlmnHT) المنحط
(wwqH) ووقح	(mnHTyn) منحطين

Table 5.4: The top five most similar words to a given query term using char and word level embeddings.

2019) have yielded excellent results for a wide variety of NLP tasks. We used four Arabic-specific language models that achieved state-of-the-art results on the majority of Arabic NLP tasks. We fine-tuned each of these models on offensive language detection to compare their performance. The models that we use were as follows:

- **AraBERT:** Antoun et al. (2020) provides a BERT-based model which is specifically built for the Arabic language. AraBERT was the first Arabic-specific BERT model to achieve competitive results on the majority of Arabic NLP tasks. This model was pre-trained on a large dataset containing 24 GB of text obtained from Wikipedia and a variety of news sources across the Arab world.

- **ArabicBERT:** Another Arabic-specific BERT model proposed by (Safaya et al., 2020). They used 95 GB of Arabic text from different sources, including Arabic Wikipedia and the Arabic version of OSCAR, a massive multilingual corpus (Ortiz Suárez et al., 2019).

- **MARBERT:** Abdul-Mageed et al. (2021) released for transfer learning on Arabic dialects. MARBERT Pre-trained on a vast dataset containing 6 billion tweets, MARBERT produces state-of-the-art outcomes in many tasks involving Arabic-language NLP.

- **QARiB:** Proposed by Chowdhury et al. (2020), this model was trained using a variety of MSA and dialects sources. They used around 420M tweets and 180M sentences from article news. From the author's observation, using such mixture sources (MSA and dialects) to pre-train a language model is more likely to improve the performance in classification tasks.

5.3.4 Affect Transfer Learning

While some existing studies exploit sentiment analysis, no research, to the best of our knowledge, leverages emotional intensity for the specific purpose of enhancing the detection of offensive language. We hypothesise that the task of offensive language detection can be further enhanced by transferring learning from other sources of emotion-related tasks.

We used our method in Chapter 4 (BERT-ACWE) to classify the offensive language dataset into different intensity levels of sentiment and emotion classes. BERT-ACWE integrates BERT and ACWE (proposed in Chapter 3) at a sentence level. Once BERT is fine-tuned using the training dataset, the contextualised vectors are retrieved. For ACWE, the CNN-LSTM model is fed by a combination of CE and WE to obtain sentence vectors. Both obtained vectors are connected and linked with the remaining layers of the neural network. Lastly, in order to produce the final output, softmax plays the role of an activation function to predict the class or the probability distribution for each of the following affect tasks:

- **Sentiment Model:** We first train BERT-ACWE on one of the largest sentiment analysis dataset ASAD, described in Section 4.3.1. Then, the trained model is used to classify tweets from offensive language datasets into one of the sentiment classes: negative, neutral, and positive. This prediction step is used to initially explore the correlation between offensive tweets and negative sentiment, as presented in Section 5.2.2. Consequently, the trained model is used also to obtain the final probability distribution for the three sentiment classes. These probabilities will be used as input vectors beside other affect tasks, as will be explained later. We refer to this trained model as *Sent*.

- **Sentiment Intensity Model:** Similar to *Sent*, we first train BERT-ACWE on the sentiment intensity dataset V-oc, described in Section 3.5.1, to predict one of seven classes: three classes from the lowest to the highest level of negative, similarly three classes for positive, and a neutral class. Additionally, the trained model is used to obtain the final probability distribution for the seventh class. We refer to this trained model as *Sent-I*.

- **Emotion Model:** Similar to the above models, we first train BERT-ACWE on the emotion dataset E-c, described in Section 4.3.1, to predict one of four emotions: anger, fear, joy, and sadness. Additionally, the trained model is used to obtain the final probability distribution for the four emotion classes. We refer to this trained model as *Emo*.

- **Anger Intensity Model:** Similar to the aforementioned models, we first train BERT-ACWE on the anger sub dataset from the emotional intensity task EI-oc, described in Section 3.5.1, to predict the intensity level of anger tweets: three classes from the lowest to the highest level of anger. Additionally, the trained model is used to obtain the final probability distribution for the three classes. We refer to this trained model as *Anger-I*.

To this end, we concatenated all probability distributions obtained from Sent, Sent-I, Emo, and Anger-I models. This provided us with an affect representation, a feature vector of 18 dimensions. The representation was then passed to machine learning models as an input feature to predict given tweets whether they are offensive or inoffensive based on their affect representation. We refer to this final model and its output as *Affect-TL* (Affect Transfer Learning).

5.3.5 Multi-task Learning for Sarcasm

From our observation, we found several offensive tweets that were mis-classified as positive sentiment or joy due to sarcasm. Therefore, we assume that using a state-of-the-art model to detect sarcasm can further improve the performance of offensive language detection besides Affect-TL. We used our multi-task learning method that achieved the best result in the

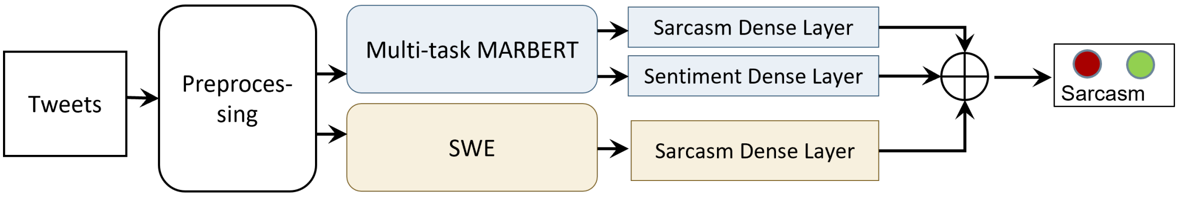


Figure 5.4: *Sarc-MTL* architecture.

sarcasm detection shared-task (Abu Farha et al., 2021). We refer to our proposed model as *Sarc-MTL*.

Multi-Task Learning (MTL) is an approach to inductive transfer that enhances generalisation by using the domain knowledge found in the training signals of similar tasks as an inductive bias. The intuition behind MTL is that a useful feature for one task will be useful and thus predictive for other, similar tasks (Caruana, 1997). There has been little research into the idea that these two tasks influence each other (Majumder et al., 2019; C. Zhang & Abdul-Mageed, 2019a). We exploit this relationship since the dataset released by Abu Farha et al. (2021) provides both labels (sentiment/sarcasm) for each tweet.

Similar to affect models, *Sarc-MTL* used a combination of the SWE and BERT to detect sarcasm. However, our system enhanced this combination of representations by additionally sharing contextual sentiment vectors after fine-tuning the BERT model on the sentiment task. In other words, instead of using only contextual sarcasm vectors to detect sarcasm, it is concatenated with a similar vector but trained on sentiment. Figure 5.4 illustrates the architecture of *Sarc-MTL*.

5.3.6 The proposed Method: Ensemble Multi-task learning

Integrating multiple kinds of classifiers can help overcome the weaknesses and realise the advantages of each. We used an ensemble technique to combine the classifiers via a majority voting method. We selected the two best-performing CLM models—MARBERT and QARiB—as candidates for the ensemble method. Additionally, we selected CNN-ACWE as a candidate from the SWE models, Affect-TL, and Sarc-MTL classifiers. Each of these five classifiers had a vote (class). We compared ensembling three classifiers, MARBERT and QARiB (CLMs) with one of the remaining candidates, and ensembling the five classifiers (ALL).

5.4 Results

In this section, we report our results from three experiments. First, we compare four pre-trained SWE models by using them individually and a combination of character and word levels. Second, we examine the impact of using affect transfer learning models. Finally, we present the results of our proposed ensemble models. We use the same evaluation metrics provided by the authors releasing the datasets: F1-measure, precision, recall and accuracy. For all experiments, we use the dataset OSACT 2020 described in Section 5.2.1. For the third experiment, we use an additional dataset (OSACT 2022) that has been released very recently (Mubarak et al., 2022) to evaluate the robustness of our proposed ensemble method. Similar to OSACT 2020, the distribution of the targeted classes for OSACT 2022 was unbalanced: 4,463 tweets were labelled as offensive, while the remaining (8,235) were labelled as inoffensive.

5.4.1 Experiment 1: SWE Models Comparison

We evaluated the use of four pre-trained word embeddings: two open-source word-level models (Mazajak and Ara2vec) and our generated models (WE and CE). We compared the performance of these models individually and by combining CE with each of the word-level embeddings to detect offensive language tweets. Table 5.5 compares the performance of using single or multi-levels of SWEs in three different machine and deep learning classifiers. In general, the combination of character- and word-level embeddings outperforms using individual SWE models across all three machine and deep learning algorithms. This result indicates the importance of using multi-level SWE to improve the performance of offensive language classifiers. It can be observed that the integration of CE and Mazajak obtained the best results, compared to combining CE with Ara2vec or WE. This can be interpreted, when we note the performance of each word-level model individually, as solo Mazajak outperformed WE and Ara2vec. Although CE trained only on 10 million tweets, it achieved a better result than Mazajak, which trained on 250 million tweets. However, combining different levels of models improved the results from 0.5% to 1%, compared to solo CE. We believe that this combination takes advantage of large word-level embeddings (Mazajak) and overcomes their limitation by incorporating CE to deal with the OOV problem. An example of OOV taken from the dataset is an offensive word (الكلبوبه - Alklbwbh), meaning the small female dog, and it could not be realised by all aforementioned word-level models. However, the CE model was able to capture its meaning by encoding this word close to other related words that either have the same meaning or are mostly a different form of this word.

Table 5.5: Performance results (F-score) for using SWEs as an input feature in machine and deep learning algorithms.

SWE	Level	CNN	SVC	XGB	avg.
Mazajak	Word	89.4	85.0	85.7	86.7
Ara2vec	Word	88.9	84.3	84.8	85.6
WE	Word	88.8	84.1	85.5	86.1
CE	Character	89.8	86.0	86.2	87.3
CE + Mazajak	Character	**90.3**	**86.8**	**87.2**	**88.1**
CE + Ara2vec	+	89.6	86.6	86.5	87.6
CE + WE	word	89.4	86.3	86.8	87.5

5.4.2 Experiment 1: CLM Models Comparison Results

We conducted experiments to compare four CLMs developed specifically for Arabic: ArabicBERT, AraBERT, QARiB and MARBERT. We fine-tuned each of these models on the offensive language training set and report results on the test set using the main evaluation metrics (F1-score) and other metrics (accuracy, precision and recall). It can be observed that MARBERT obtained the best result (F1-score of 92.51), followed by QARiB, which achieved a competitive result (92.12). When it comes to correctly predicting offensive tweets (precision), QARiB slightly outperforms MARBERT. Additionally, we note that the performance of ArabicBERT and AraBERT fell behind the combination of multi-level SWE with CNN.

Table 5.6: Performance results for fine-tuning four CLMs on the offensive language task

Model	Accuracy	Precision	Recall	F1-score
ArabicBERT	92.14	87.91	87.50	87.70
AraBERT	93.21	89.29	89.69	89.48
QARiB	95.05	**93.11**	91.22	92.12
MARBERT	**95.25**	93.05	**92.00**	**92.51**

5.4.3 Experiment 2: Impact of Affect Transfer Learning

In this experiment, we investigate the effectiveness of transferring affect learning for the offensive language detection task. We explore the effect of utilising a range of affect-related tasks, namely sentiment strength, emotion classification and anger intensity. We compare performance when using each one of the affect classifiers individually versus combining them. The evaluation results are reported in Table 5.7.

As for solo *Affect-TL* classifiers, *Anger-I* obtained the best result (F1-score of 84.79), outperforming other affect models by a range from 1.23% to 5.59% based on the F1-score metric. However, this range becomes noticeably lower (from 0.18% to 2.48%) when we compare the performance of these individual models using precision metrics. This indicates that these models perform better when it comes to correctly predicting offensive tweets. Moreover, *Sent-I* is least effective, with a noticeable difference ranging between 2% and 6% compared to other models. We believe that the low performance for Sent-I is due to the small training set size (1800 tweets) further complicated by the tweets being divided into seven imbalanced classes.

As for *Affect-TL* incorporation, combining emotion classifiers (*Emo* and *Anger-I*) had a higher impact than sentiment classifiers (*Sent* and *Sent-I*), with a difference of about 2.5%. From error analysis observation, we found that most of the false positives[1] of Sent are labelled as negative, while *Emo* labelled them as fear or sadness. For example, the tweet (💔 يا التاكسي خذني لها يا التاكسي - Oh taxi, take me to her, oh taxi 💔) was incorrectly predicted as 'OFF' by the Sent models, while it was predicted correctly by Emo models as it was labelled as sadness with no anger intensity. Finally, we found that incorporating all models provide the best result, even if we add the lowest solo performing classifier (*Sent-I*). This indicates the importance of involving a range of different affect-related tasks as transfer learning features for the offensive language detection task.

[1] *Not-OFF* tweets that were incorrectly classified as *OFF*

Table 5.7: Performance results for using each one of affect classifiers individually and by combining them

Affect-TL Model	Accuracy	Precision	Recall	F1-score
Sent	88.80	84.68	78.28	80.86
Sent-I	87.70	82.76	76.29	78.84
Emo	89.85	85.06	82.29	83.56
Anger-I	**90.35**	**85.24**	**84.37**	**84.79**
Sent + Sent-I	90.6	87.78	81.46	84.08
Emo + Anger-I	92.05	89.27	85.06	86.94
All except Sent-I	92.85	91.00	85.94	88.16
All	**93.20**	**91.67**	**86.44**	**88.72**

5.4.4 Experiment 3: The proposed Method: Ensemble Multi-task learning Results

We conducted experiments to compare the performance of four ensemble models that described in Section 5.3.6. We used MARBERT (the best solo performing model) as a baseline to investigate if transferring features from anger intensity and other affect-tasks can enhance the results of offensive language detection. In addition to OSACT 2020 dataset, we used an additional dataset (OSACT 2022) to evaluate the robustness of our proposed ensemble method. Tables 5.8 and 5.9 present the results for the previous state-of-the-art methods, MARBERT and ensemble models. In general, it can be noted that using multi-classifiers (ensemble) outperformed a single MARBERT model and the previous SOTA for both datasets.

As for ensemble methods, combining the five classifiers (*ALL*) and *CLMs+Affect-TL* obtained the best results, improving state-of-the-art by about 2% to 3% in both datasets. We found that each classifier had advantages and weaknesses after analysing the predictions on the test set. For instance, the tweet (طيب مين احلى صوته ولا صوتي لما احكي يا سيسي يا سيسي - Ok, who has the best voice or my voice when I speak, O Sisi, O Sisi?) is incorrectly predicted as 'OFF' by the solo ACWE model and 'NOT_OFF' by the CLM models. Because the word (Sisi) appears in multiple 'OFF' tweets, static ACWE was unable to determine the context. The contextual language model (MARBERT), on the other hand, was better able to capture (Sisi). Moreover, for OSACT 2020, we found that *ALL* slightly outperformed *CLMs+Affect-TL* in F1 metrics, *CLMs+Affect-TL* had better results in precision metrics.

This result indicates the importance of using anger intensity and other related emotion tasks when predicting offensive tweets.

Table 5.8: Performance results for OSACT 2020 using different ensemble classifiers against the previous state-of-the-art (Hassan et al., 2020)

Model	C	Accuracy	Precision	Recall	F1-score
previous state-of-the-art	1	93.85	90.18	90.85	90.51
MARBERT	1	95.25	93.05	92.00	92.51
CLMs+ACWE	3	95.6	95.01	91.01	92.84
CLMs+Affect-TL	3	**95.85**	**94.66**	92.19	93.36
CLMs+Sarc	3	95.3	92.75	92.59	92.67
ALL	5	**95.85**	93.94	**93.03**	**93.47**

Table 5.9: Performance results for OSACT 2022 using different ensemble classifiers against the previous state-of-the-art (Mubarak et al., 2022)

Model	C	Accuracy	Precision	Recall	F1-score
previous state-of-the-art	1	84.02	82.53	82.11	82.31
MARBERT	1	85.66	83.36	83.87	83.61
CLMs+ACWE	3	85.98	83.80	83.80	83.88
CLMs+Affect-TL	3	85.98	84.70	84.29	84.49
CLMs+Sarc	3	85.87	84.67	83.91	84.28
ALL	5	**86.34**	**85.32**	**84.28**	**84.75**

Table 5.10 presents other examples where the proposed ensemble method was able to correctly predict OFF (TP), incorrectly predict OFF (FP), correctly predict NOT_OFF (TN) and incorrectly predict NOT_OFF (FN). We selected samples where the best-performing solo classifies MARBERT (C1) and QARiB (C2) disagreed in their final prediction (except for the last example). It is clear that the majority of these examples are complex due to sarcasm or tweets that need more understanding. Involving different resources contributes to the improved performance of our proposed method. It is worth mentioning that both C1 and C2 predicted the last offensive tweet (No. 8) as NOT_OFF (0), while ACWE (C3) considered it as an OFF tweet (1). Affect-TL (C4) considered it positive in terms of sentiment and emotion, because all words convey feelings of respect and love except for the last word, which is a swear word. The last offensive word is actually a combination of two words used in

some Arabic dialects. This context of using positive words with a complex morphology in an offensive word resulted in a false prediction for our proposed method.

	PC	Tweet	C1	C2	C3	C4	C5	L
1	TP	يا ربي عليك يا محمود يا غباء انت Oh my God, oh Mahmoud, you are stupid	0	1	0	1	1	1
2		يا عيني عالتبديل يا قاتوزو يا ابن الغوريلا Oh my eyes, oh Gattuso, son of the gorilla	1	0	1	1	0	1
3	FP	الاتحاد كبير اسيا وافضل من انجبت الكوره الله اكبر يا عميد على راسي كل اتحادي Itihad is big team in Asia, and the best of those that gave birth to football. Allah Akbar, O Dean. Over my head, every fan is great	1	0	1	0	1	0
4		يا رباه مستحيل يا اللهي ما الذي يحدث تبا للفار وتقنيه الفار وحكام الفار Oh God, it is impossible, what is happening, loser VAR and tech and VAR's referee.	0	1	1	0	1	0
5	TN	يا مجنونه يا بنت المجنونه يا كوره Oh crazy oh crazy girl oh football	0	1	0	0	1	0
6		انت مش قادر تفوز على الوصل تبي تلعب بلقوه يعني يا العب يا ما اتفرج كبر عقلك الله يرضى عليك You are not able to win against Al Wasl, you want to play with strength, either you play or watch. Think again.	1	0	1	0	0	0
7	FN	يا كذبك يا عطوي خاف الله Oh your lie, oh Atwi, fear God	1	0	1	0	0	1
8		بنحبك وبنحترمك جدا يا دكتور علاء يا ابونـ**** We love you and respect you very much, Dr. father-f*****	0	0	1	0	0	1

Table 5.10: Examples of correct and incorrect predictions from five models (Pc= Predicted condition, C1= MARBERT, C2= QARiB, C3= CNN-Mazajak+CE, C4= Affect-TL, C5= Sarcasm and L= Actual Label where 1 is *OFF* and 0 is *NOT_OFF*).

5.5 Conclusion

In this chapter, different transfer learning approaches were used to enhance the performance of offensive language detection. Four static word embeddings were first compared using pre-trained models individually and combining character- and word-level models. We found that integrating our generated character models with the largest pre-trained model (Mazajak) produced highly competitive results with the previous state-of-the-art in the task of offensive language. Then, four pre-trained Arabic-specific BERT models were compared and fine-tuned on the offensive language dataset. We found that MARBERT and QARiB are the best-performing models, outperforming the previous state-of-the-art that used the AraBERT model.

Some prior studies have used sentiment and emotion classification, but to our knowledge, no research has employed anger intensity to improve the identification of offensive language.

Several experiments were conducted to investigate our hypothesis that transferring anger intensity and other affect-related tasks (Affect-TL) can enhance offensive language detection. We found that incorporating the best-performing BERT models with Affect-TL in an ensemble method improved the results and achieved state-of-the-art.

In future studies, we hope to enhance the performance of our model by applying additional pre-processing techniques and more effectively exploiting a list of offensive words. In fact, some Arabic words are considered offensive regardless of their context. Additionally, we aim to investigate various methods by which data can be augmented into our training dataset to enrich samples of offensive content so that algorithm learning can be more effective.

Chapter 6

Conclusions

In this dissertation, we studied two main problems: emotional intensity and offensive language detection in Arabic microblogs. We first proposed a novel combination of static character- and word-level embeddings to improve the detection of emotional intensity. Then, we enhanced contextualised language models by incorporating our proposed static embeddings into the emotional intensity task. Finally, we transferred learning from emotional intensity and other emotion-related tasks to offensive language detection.

This chapter concludes by briefly summarising the contributions and presenting future research directions.

6.1 Summary of Contributions

In this dissertation, we proposed and examined several approaches to detecting emotional intensity using static word embeddings and contextual language models. We then exploited emotional intensity and other emotion-related features to improve the performance of offensive language detection. The essential contributions of this dissertation can be briefly summarised as follows.

In Chapter 3, We proposed a novel method combining character- and word-level embeddings (ACWE) to improve the performance of emotional intensity detection. While most research work is concerned with building word-level embedding models, to our knowledge there are no models available for use at the word- and character-level designed to detect emotion intensity in Arabic. Our generated models have been released to be used as pre-trained word embeddings for applications and research relying on Arabic sentiment and emotion analysis. Additionally, we conducted a systematic analysis to investigate the role of applying pre-processing techniques to a large training corpus prior to generating word embeddings, a study that has not previously been conducted for noisy, user-generated text.

Our generated character-level model significantly outperformed state-of-the-art pre-trained Arabic word embeddings in emotional intensity tasks. We evaluated the robustness of ACWE by using four affect-related tasks (in addition to emotional intensity detection). ACWE obtained better results by 1.3% to 5% across all tasks. ACWE achieved a state-of-the-art result for the task of emotional intensity classification, outperforming the top-performing systems. This answers **RQ1** and confirms the importance of leveraging character-level and word-level embeddings in emotional intensity detection. Finally, our experiments also showed the importance of applying simple methods to clean noisy data, keep emojis and segment hashtag words.

In Chapter 4, we focused more in improving the performance of emotional intensity detection by investigating advanced deep learning algorithms and contextualised language models. First, we used ACWE (proposed in Chapter 3) as input to various DL approaches in order to examine the impact of employing such advanced learning on emotional intensity detection. Then, we investigated and evaluated the performance of two multilingual and four monolingual language models in emotional intensity detection. To the best of our knowledge, this is the first study attempting to employ contextual embeddings for emotional intensity tasks. Finally, we proposed a novel method for enhancing the best-performing language models by incorporating ACWE in emotional intensity tasks.

We first showed the experimental results for four deep learning models that used ACWE as input. We found that these sophisticated models could not boost the performance of emotional intensity tasks. We believe this was due to the small size of the training dataset. This led us to **RQ2** for exploring contextual language models and evaluating them on emotional intensity datasets. We found that the monolingual language models obtained significantly better results than the multilingual models. More specifically, MARBERT obtained the best result among all emotions for both EI-reg and EI-oc tasks. Finally, we showed the importance of leveraging contextual and static word embeddings in emotional intensity tasks. Our proposed method revealed an improvement from about 1% to 3% across all four emotions. To evaluate the robustness of our proposed method, we used eight affect-related tasks, for which we obtained state-of-the-art results in seven tasks, including emotional intensity tasks.

In Chapter 5, we aimed to exploit emotional intensity and other affect-related tasks in the offensive language task by using transfer learning approaches. We first analysed the correlation between offensive language and anger intensity, emotion and sentiment classification. Then, we compared the performance of four pre-trained word-embedding models individually and by combining character-level (CE) with one of the word-level embeddings to detect offensive language tweets. Additionally, we compared four pre-trained contextual language models to find the best-performing models in offensive language detection. Finally, anger intensity and other emotion-related tasks were leveraged as a feature transfer learning method.

We found that the combination of character- and word-level embeddings outperformed using individual models. Specifically, the integration of CE and Mazajak obtained the best results compared to combining other models. We believe that this combination took advantage of large word-level embeddings (Mazajak) and overcame their limitation by incorporating CE to deal with the OOV problem. In terms of contextual language models, we found MARBERT obtained the best result, followed by QARiB, which achieved a competitive result. It is worth mentioning that the performance of the other language models fell behind the combination of static word embeddings.

We showed the relation between offensive language and a range of affect tasks. We found that the negative emotion and high anger intensity constituted the majority for the offensive class (70%–80%), while neutral emotion and low anger intensity were assigned to the non-offensive class. These results showed a strong association between using emotional intensity and offensive language in tweets. Then, we assessed this correlation by transferring affect learning to the offensive language detection task. We compared performance when using each of the affect classifiers individually versus combining them. We found that anger intensity obtained the best result, outperforming other affect models by 1.23% to 5.59%. We finally conducted experiments to compare the performance of five ensemble models: MARBERT, QARiB, CNN-ACWE, Affect-TL and Sarc-MTL classifiers. Our proposed transfer learning method achieved state-of-the-art result in the offensive language detection task.

6.2 Future Directions

In this dissertation, we developed effective word embeddings, after which we used these generated models to enhance pre-trained language models and improve emotional intensity detection. Finally, we employed emotional intensity enhance the effectiveness of offensive language detection. Although the research questions of our study have been addressed, there are potential directions that can be further examined in the future.

6.2.1 Annotated Dataset Creation

One of the limitations of this study is the shortage of annotated datasets for the emotional intensity task. The available datasets (S. Mohammad et al., 2018) can be expanded by increasing the number of tweets and basic emotions analysed. Chapter 4 explains how this scarcity of annotated data negatively affected performance of the deep learning algorithms used in this research; the algorithms required more training data to learn. The dataset used in this work the intensity of the four emotions from the Plutchik emotion model (Plutchik, 1980) (anger, fear, joy and sadness). Other emotions, such as disgust, surprise, anticipation and trust, can be explored in future research. For example, detecting the intensity of disgust can be useful in identifying offensive language along with anger intensity (Mubarak et al., 2022).

6.2.2 Semi-supervised Learning

As mentioned earlier, supervised deep learning models require numerous annotated training data. Accordingly, semi-supervised learning approaches are potential avenues through which to automatically increase the volume of training data for emotional intensity detection. One of these approaches is self-training, which has revealed its usefulness in a variety of text classification tasks (Pavlinek & Podgorelec, 2017; C. Zhang & Abdul-Mageed, 2019b; Ligthart et al., 2021). This simple process starts after a few labelled data are classified, followed by the use of the learned classifier to predict unlabelled datasets. High-confidence predictions are then selected for gradual augmentation and incorporation into labelled datasets. This process is performed iteratively until a threshold is reached. Enlarging the training datasets by this automated approach is a potential work direction to overcome the lack of large datasets in emotional intensity detection.

6.2.3 Alternative Resources for Enhancing Language Models

In our study, we presented how integrating static word embeddings improves the performance of contextual language models such as BERT. Incorporating other knowledge resources into BERT are directions that remain for future work (Roy & Pan, 2020). Topic models such as LDA (Blei et al., 2003) and short text topic model GSDMM (Yin & Wang, 2014) can be combined with language models. Such integration has shown promising results in semantic similarity detection (Peinelt et al., 2020). Other potential resources for inclusion in BERT are informative lexicons, which have been successfully applied to the detection of abusive language (Koufakou et al., 2020).

6.2.4 Advanced Ensemble Multi-task Methods

In Chapter 5, we used a simple multi-task ensemble method (majority voting) to combine several classifiers for the offensive language detection. Exploring more sophisticated ensemble techniques can be a direction for future work. For instance, stacked ensembles are interesting innovations for training new models to learn how to effectively integrate contributions from each classifier. This strategy has also been successfully used in sentiment classification (Subba & Kumari, 2021). Employing such advanced ensembles creates opportunities to address issues regarding detection tasks that involve emotional intensity and offensive language.

6.2.5 The Role of Emojis

We explained the importance of retaining emojis in order to detect the intensity of the emotion (Chapter 3), but questions remain about proposed methods for treating these emojis and their effects on downstream tasks. An example is the issue of converting emojis in a way that ensures correspondence with their descriptions or the categories to which they belong. A useful approach is to use lexicons constructed specifically for Arabic affect tasks. Researchers can also incorporate such lexicons into pre-trained emoji-embedding models that universally learn from textual descriptions (Eisner et al., 2016) for the detection of offensive language and emotional intensity.

6.2.6 Exploring Multi-task Learning for Affect Tasks

In Chapter 5, we proposed a multi-task learning framework that involves sentiment representations in sarcasm detection given the availability of annotated datasets for these tasks (Chapter 5). However, the absence of annotated datasets on the same tweets for other affect tasks prevented us from evaluating the proposed multi-task method of identifying emotional intensity, emotion classification and sarcasm. This study can be extended by creating comprehensive Arabic affect datasets that share tweets with those belonging to affective classes. For the English language, there has been an attempt to study multi-task learning methods for different sentiment and emotion analysis tasks owing to the availability of corresponding datasets (Akhtar et al., 2022).

References

Abbes, I., Zaghouani, W., El-Hardlo, O., & Ashour, F. (2020, May). DAICT: A dialectal Arabic irony corpus extracted from Twitter. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 6265–6271). Marseille, France: European Language Resources Association.

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016, June). Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 11–16). San Diego, California: Association for Computational Linguistics.

Abdou, M., Kulmizev, A., & Ginés i Ametllé, J. (2018). AffecThor at SemEval-2018 task 1: A cross-linguistic approach to sentiment intensity quantification in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 210–217). New Orleans, Louisiana: Association for Computational Linguistics.

Abdullah, M., Hadzikadicy, M., & Shaikhz, S. (2018). Sedat: Sentiment and emotion detection in Arabic text using cnn-lstm deep learning. In *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (p. 835-840).

Abdullah, M., & Shaikh, S. (2018). TeamUNCC at SemEval-2018 task 1: Emotion detection in English and Arabic tweets using deep learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 350–357). New Orleans, Louisiana: Association for Computational Linguistics.

Abdul-Mageed, M., AlHuzli, H., & Duaa'Abu Elhija, M. D. (2016). DINA: A multi-dialect dataset for Arabic emotion analysis. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media* (p. 29).

Abdul-Mageed, M., & Diab, M. (2011). Subjectivity and sentiment annotation of modern standard Arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop* (pp. 110–118).

Abdul-Mageed, M., & Diab, M. (2012, May). AWATIF: A multi-genre corpus for Modern Standard Arabic subjectivity and sentiment analysis. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)* (pp. 3907–3914). Istanbul, Turkey: European Language Resources Association (ELRA).

Abdul-Mageed, M., Elmadany, A., & Nagoudi, E. M. B. (2021, August). ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 7088–7105). Online: Association for Computational Linguistics.

Abdul-Mageed, M., Zhang, C., Hashemi, A., & Nagoudi, E. M. B. (2020, May). AraNet: A deep learning toolkit for Arabic social media. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 16–23). Marseille, France: European Language Resource Association.

Abu Farha, I., & Magdy, W. (2021). A comparative study of effective approaches for Arabic sentiment analysis. *Information Processing Management*, *58*(2), 102438.

Abu Farha, I., & Magdy, W. (2019). Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop* (pp. 192–198). Florence, Italy: Association for Computational Linguistics.

Abu Farha, I., & Magdy, W. (2020a, May). From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 32–39). Marseille, France: European Language Resource Association.

Abu Farha, I., & Magdy, W. (2020b, May). Multitask learning for Arabic offensive language and hate-speech detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 86–90). Marseille, France: European Language Resource Association.

Abu Farha, I., Zaghouani, W., & Magdy, W. (2021, April). Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 296–305). Kyiv, Ukraine (Virtual): Association for Computational Linguistics.

Akhtar, M. S., Ghosal, D., Ekbal, A., Bhattacharyya, P., & Kurohashi, S. (2018). A multi-task ensemble framework for emotion, sentiment and intensity prediction. *arXiv preprint arXiv:1808.01216*.

Akhtar, M. S., Ghosal, D., Ekbal, A., Bhattacharyya, P., & Kurohashi, S. (2022). All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE Transactions on Affective Computing*, *13*(1), 285-297.

Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Towards accurate detection of offensive language in online communication in Arabic. *Procedia computer science*, *142*, 315–320.

Alami, H., Ouatik El Alaoui, S., Benlahbib, A., & En-nahnahi, N. (2020, December). LISAC FSDM-USMBA team at SemEval-2020 task 12: Overcoming AraBERT's pretrain-finetune discrepancy for Arabic offensive language identification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 2080–2085). Barcelona (online): International Committee for Computational Linguistics.

Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., & Al-Kabi, M. N. (2019). A comprehensive survey of Arabic sentiment analysis. *Information Processing & Management*, *56*(2), 320–342.

Alghanmi, I., Espinosa Anke, L., & Schockaert, S. (2020, November). Combining BERT with static word embeddings for categorizing social media. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)* (pp. 28–33). Online: Association for Computational Linguistics.

Alharbi, B., Alamro, H., Alshehri, M., Khayyat, Z., Kalkatawi, M., Jaber, I. I., & Zhang, X. (2021). *Asad: A twitter-based benchmark A sentiment analysis dataset.*

Al-Khatib, A., & El-Beltagy, S. R. (2018). Emotional tone detection in Arabic tweets. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (pp. 105–114). Cham: Springer International Publishing.

Alomari, K. M., ElSherif, H. M., & Shaalan, K. (2017). Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 602–610).

Alshutayri, A., & Atwell, E. (2019, July). Classifying Arabic dialect text in the social media Arabic dialect corpus (SMADC). In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics* (pp. 51–59). Cardiff, United Kingdom: Association for Computational Linguistics.

Altowayan, A. A., & Elnagar, A. (2017). Improving Arabic sentiment analysis with sentiment-specific embeddings. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)* (pp. 4314–4320). Boston.

Altowayan, A. A., & Tao, L. (2016). Word embeddings for Arabic sentiment analysis. In *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data)* (pp. 3820–3825). Washington.

Al-Twairesh, N. (2021). The evolution of language models applied to emotion analysis of Arabic tweets. *Information*, *12*(2), 84.

Al-Twairesh, N., Al-Khalifa, H., & Al-Salman, A. (2016). Arasenti: large-scale twitter-specific Arabic sentiment lexicons. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 1: Long papers)* (pp. 697–705).

Al-Twairesh, N., Al-Khalifa, H., Al-Salman, A., & Al-Ohali, Y. (2017). Arasenti-tweet: A corpus for Arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, *117*, 63-72. (Arabic Computational Linguistics)

Aly, M., & Atiya, A. (2013). Labr: A large scale Arabic book reviews dataset. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics (volume 2: Short papers)* (pp. 494–498).

AlZoubi, O., Tawalbeh, S. K., & Mohammad, A.-S. (2020). Affect detection from Arabic tweets using ensemble and deep learning techniques. *Journal of King Saud University-Computer and Information Sciences*.

Antoun, W., Baly, F., & Hajj, H. (2020, May). AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source A Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 9–15). Marseille, France: European Language Resource Association.

Arslan, Y., Küçük, D., & Birturk, A. (2018). Twitter sentiment analysis experiments using word embeddings on datasets of various scales. In M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, & F. Meziane (Eds.), *Natural Language Processing and Information Systems* (pp. 40–47). Cham: Springer International Publishing.

Babanejad, N., Agrawal, A., An, A., & Papagelis, M. (2020, July). A comprehensive analysis of preprocessing for word representation learning in affective tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5799–5810). Online: Association for Computational Linguistics.

Badaro, G., Baly, R., Hajj, H., Habash, N., & El-Hajj, W. (2014). A large scale Arabic sentiment lexicon for Arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)* (pp. 165–173).

Badaro, G., El Jundi, O., Khaddaj, A., Maarouf, A., Kain, R., Hajj, H., & El-Hajj, W. (2018, June). EMA at SemEval-2018 task 1: Emotion mining for Arabic. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 236–244). New Orleans, Louisiana: Association for Computational Linguistics.

Badaro, G., Jundi, H., Hajj, H., El-Hajj, W., & Habash, N. (2018). Arsel: A large scale Arabic sentiment and emotion lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bordes, A., Chopra, S., & Weston, J. (2014). Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 615–620). Doha, Qatar: Association for Computational Linguistics.

Borod, J. C. (2000). *The neuropsychology of emotion.* Oxford University Press.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., . . . Oflazer, K. (2018, May). The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA).

Bouazizi, M., & Otsuki Ohtsuki, T. (2016). A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, *4*, 5477-5488.

Buechel, S., & Hahn, U. (2017, April). EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics: Volume 2, short papers* (pp. 578–585). Valencia, Spain: Association for Computational Linguistics.

Calvo, R. A., & Mac Kim, S. (2013). Emotions in text: dimensional and categorical models. *Computational Intelligence*, *29*(3), 527–543.

Çano, E., & Morisio, M. (2017). Quality of word embeddings on sentiment analysis tasks. In *Natural Language Processing and Information Systems* (pp. 332–338). Cham: Springer International Publishing.

Caruana, R. (1997). Multitask Learning. *Machine Learning*. doi: 10.1023/A:1007379606734

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 785–794). New York, NY, USA: Association for Computing Machinery.

Chowdhury, S. A., Abdelali, A., Darwish, K., Soon-Gyo, J., Salminen, J., & Jansen, B. J. (2020, December). Improving Arabic text categorization using transformer training diversification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop* (pp. 226–236). Barcelona, Spain (Online): Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2020, July). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451). Online: Association for Computational Linguistics.

Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Darwish, K., Habash, N., Abbas, M., Al-Khalifa, H., Al-Natsheh, H. T., Bouamor, H., ... Mubarak, H. (2021, mar). A Panoramic survey of Natural Language

Processing in the Arab world. *Commun. ACM*, *64*(4), 72–81.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media.*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Minneapolis: Association for Computational Linguistics.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., & Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1370–1380). Baltimore, Maryland.

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (pp. 231–240).

Djandji, M., Baly, F., Antoun, W., & Hajj, H. (2020, May). Multi-task learning using AraBert for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 97–101). Marseille, France: European Language Resource Association.

Dos Santos, C., & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 69–78). Dublin, Ireland: Dublin City University and Association for Computational Linguistics.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing*

Systems (pp. 155–161).

Duwairi, R., & El-Orfali, M. (2014). A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *J. Inf. Sci.*, *40*(4), 501–513.

Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., & Riedel, S. (2016, November). emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media* (pp. 48–54). Austin, TX, USA: Association for Computational Linguistics.

Ekman, P. (1992). An Argument for Basic Emotions. *Cognition and Emotion*, *6*(3-4), 169–200.

El-Beltagy, S. R. (2016). Nileulex: A phrase and word level sentiment lexicon for egyptian and modern standard Arabic. In *Proceedings of the Tenth Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 2900–2905).

El Gohary, A. F., Sultan, T. I., Hana, M. A., & Dosoky, M. M. (2013). A computational approach for analyzing and detecting emotions in Arabic text. *International Journal of Engineering Research and Applications (IJERA)*, *3*(3), 100–107.

Elmadany, A., Zhang, C., Abdul-Mageed, M., & Hashemi, A. (2020, May). Leveraging affective bidirectional transformers for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 102–108). Marseille, France: European Language Resource Association.

El Mahdaouy, A., El Mekki, A., Essefar, K., El Mamoun, N., Berrada, I., & Khoumsi, A. (2021). Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 334–339).

Elnagar, A., & Einea, O. (2016). Brad 1.0: Book reviews in Arabic dataset. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications*

(AICCSA) (p. 1-8).

Elnagar, A., Khalifa, Y. S., & Einea, A. (2018). Hotel Arabic-reviews dataset construction for sentiment analysis applications. In K. Shaalan, A. E. Hassanien, & F. Tolba (Eds.), *Intelligent natural language processing: Trends and applications* (pp. 35–52). Cham: Springer International Publishing.

Elnagar, A., Yagi, S. M., Nassif, A. B., Shahin, I., & Salloum, S. A. (2021). Systematic literature review of dialectal Arabic: Identification and detection. *IEEE Access*, *9*, 31010-31042.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017, September). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1615–1625). Copenhagen, Denmark: Association for Computational Linguistics.

Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, *49*(2), 28.

Guellil, I., Saâdane, H., Azouaou, F., Gueni, B., & Nouvel, D. (2021). A natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*, *33*(5), 497–507.

Habash, N. Y. (2010). Introduction to Arabic Natural Language Processing. *Synthesis lectures on human language technologies*, *3*(1), 1–187.

Hassan, S., Samih, Y., Mubarak, H., Abdelali, A., Rashed, A., & Chowdhury, S. A. (2020, May). ALT submission for OSACT shared task on offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 61–65). Marseille, France: European Language Resource Association.

Hinduja, S., & Patchin, J. W. (2019). Connecting adolescent suicide to the severity of bullying and cyberbullying. *Journal of School Violence*, *18*(3), 333-346.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computa-*

tion, *9*(8), 1735–1780.

Husain, F. (2020, May). OSACT4 shared task on offensive language detection: Intensive preprocessing-based approach. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 53–60). Marseille, France: European Language Resource Association.

Husain, F., & Uzuner, O. (2021, mar). A survey of offensive language detection for the arabic language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, *20*(1).

Ibrahim, H. S., Abdou, S. M., & Gheith, M. (2015). Automatic expandable large-scale sentiment lexicon of modern standard Arabic and colloquial. In *First International Conference on Arabic Computational Linguistics (ACLing)* (p. 94-99).

Jabreel, M., & Moreno, A. (2018). EiTAKA at SemEval-2018 task 1: an ensemble of n-channels ConvNet and XGboost regressors for emotion analysis of tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 193–199). New Orleans, Louisiana: Association for Computational Linguistics.

Keleg, A., El-Beltagy, S. R., & Khalil, M. (2020, May). ASU_OPTO at OSACT4 - offensive language detection for Arabic text. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 66–70). Marseille, France: European Language Resource Association.

Kelly, Y., Zilanawala, A., Booker, C., & Sacker, A. (2018). Social media use and adolescent mental health: Findings from the uk millennium cohort study. *EClinicalMedicine*, *6*, 59-68.

Kim, Y. (2014, October). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751). Doha, Qatar: Association for Computational Linguistics.

Koufakou, A., Pamungkas, E. W., Basile, V., & Patti, V. (2020, November). Hurt-

BERT: Incorporating lexical features with BERT for the detection of abusive language. In *Proceedings of the Fourth Workshop on Online Abuse and Harms* (pp. 34–43). Online: Association for Computational Linguistics.

Kralj Novak, P., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of emojis. *PLoS ONE*, *10*(12), 1-22.

Kudo, T., & Richardson, J. (2018, November). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 66–71). Brussels, Belgium: Association for Computational Linguistics.

Kumar, R., et al. (Eds.). (2020, May). *Proceedings of the second workshop on trolling, aggression and cyberbullying.* Marseille, France: European Language Resources Association (ELRA).

Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018, August). Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (pp. 1–11). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Lee, D.-H., et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML* (Vol. 3, p. 896).

Lei, Z., Yang, Y., Yang, M., & Liu, Y. (2018, July). A multi-sentiment-resource enhanced attention network for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 758–763). Melbourne, Australia: Association for Computational Linguistics.

Li, Q., Shah, S., Liu, X., & Nourbakhsh, A. (2017). Data sets: word embeddings learned from tweets and general data. In *Proceedings of the Eleventh International Conference on Web and Social Media (ICWSM-17)* (pp. 428–436). Montréal, Canada: AAAI Press.

Li, X., Rao, Y., Xie, H., Liu, X., Wong, T.-L., & Wang, F. L. (2019). Social emotion classification based on noise-aware training. *Data Knowledge Engineering*, *123*, 101605.

Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. *Applied Soft Computing*, *101*, 107023.

Lin, C.-C., Ammar, W., Dyer, C., & Levin, L. (2015). Unsupervised POS induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1311–1316). Denver, Colorado: Association for Computational Linguistics.

Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge university press.

Liu, R., Shi, Y., Ji, C., & Jia, M. (2019). A survey of sentiment analysis based on transfer learning. *IEEE Access*, *7*, 85401-85412. doi: 10.1109/ACCESS.2019.2925059

Mahmoud, A., & Zrigui, M. (2019). Deep neural network models for paraphrased text classification in the Arabic language. In E. Métais, F. Meziane, S. Vadera, V. Sugumaran, & M. Saraee (Eds.), *Natural Language Processing and Information Systems* (pp. 3–16). Cham: Springer International Publishing.

Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., & Gelbukh, A. (2019). Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, *34*(3), 38-43. doi: 10.1109/MIS.2019.2904691

Malmasi, S., & Zampieri, M. (2017, September). Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017* (pp. 467–472). Varna, Bulgaria: INCOMA Ltd.

Mehrabian, A. (1980). Basic dimensions for a general psychological theory implications for personality, social, environmental, and developmental studies.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (p. 3111–3119). Red Hook, NY, USA: Curran Associates Inc.

Mohammad, S. (2016). Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text. In *Emotion Measurement* (pp. 201–237). Elsevier.

Mohammad, S., & Bravo-Marquez, F. (2017, August). Emotion intensities in tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)* (pp. 65–77). Vancouver, Canada: Association for Computational Linguistics.

Mohammad, S., Bravo-Marquez, F., Salameh, M., & Kiritchenko, S. (2018). SemEval-2018 task 1: Affect in Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 1–17). New Orleans, Louisiana: Association for Computational Linguistics.

Mohammad, S. M. (2018). Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018).* Miyazaki, Japan.

Mubarak, H., & Darwish, K. (2019a). Arabic offensive language classification on twitter. In *Social Informatics: 11th International Conference (SocInfo 2019)* (p. 269–276). Berlin, Heidelberg: Springer-Verlag.

Mubarak, H., & Darwish, K. (2019b). Arabic offensive language classification on twitter. In I. Weber et al. (Eds.), *Social Informatics* (pp. 269–276). Cham: Springer International Publishing.

Mubarak, H., Darwish, K., & Magdy, W. (2017, August). Abusive language detection on Arabic social media. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 52–56). Vancouver, BC, Canada: Association for Computational Linguistics.

Mubarak, H., Darwish, K., Magdy, W., Elsayed, T., & Al-Khalifa, H. (2020, May). Overview of OSACT4 Arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection* (pp. 48–52). Marseille, France: European Language Resource Association.

Mubarak, H., Hassan, S., & Chowdhury, S. A. (2022). *Emojis as anchors to detect Arabic offensive language and hate speech.* arXiv. Retrieved from `https://arxiv.org/abs/2201.06723` doi: 10.48550/ARXIV.2201.06723

Mubarak, H., Rashed, A., Darwish, K., Samih, Y., & Abdelali, A. (2021, April). Arabic offensive language on Twitter: Analysis and experiments. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 126–135). Kyiv, Ukraine (Virtual): Association for Computational Linguistics.

Nabil, M., Aly, M., & Atiya, A. (2015a). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2515–2519).

Nabil, M., Aly, M., & Atiya, A. (2015b, September). ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2515–2519). Lisbon, Portugal: Association for Computational Linguistics.

Obeid, O., Zalmout, N., Khalifa, S., Taji, D., Oudah, M., Alhafni, B., … Habash, N. (2020, May). CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 7022–7032). Marseille, France: European Language Resources Association.

Ortiz Suárez, P. J., Sagot, B., & Romary, L. (2019, July). Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In P. Bański et al. (Eds.), *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7).* Cardiff, United Kingdom: Leibniz-Institut für

Deutsche Sprache.

Orzechowski, P., La Cava, W., & Moore, J. H. (2018). Where are we now? a large benchmark study of recent symbolic regression methods. In *Proceedings of the Genetic and Evolutionary Computation Conference* (p. 1183–1190). New York, NY, USA: Association for Computing Machinery.

Parker, R., Graff, D., Chen, K., Kong, J., & Maeda, K. (n.d.). *Arabic gigaword fifth edition robert parker, david graff, ke chen, junbo kong, kazuaki maeda.* Retrieved from `https://catalog.ldc.upenn.edu/LDC2011T11`

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., ... Roth, R. (2014, May). MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 1094–1101). Reykjavik, Iceland: European Language Resources Association (ELRA).

Pavlinek, M., & Podgorelec, V. (2017). Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, *80*, 83-93.

Peinelt, N., Nguyen, D., & Liakata, M. (2020, July). tBERT: Topic models and BERT joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7047–7055). Online: Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. (2014, October). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics.

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019, November). Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 43–54). Hong Kong, China: Association for Computational Linguistics.

Plaza-del Arco, F. M., Halat, S., Pad, S., & Klinger, R. (2021). Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. *CoRR*, *abs/2109.10255*.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion* (pp. 3–33). Elsevier.

Plutchik, R. (2001). The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, *89*(4), 344–350.

Preoţiuc-Pietro, D., Schwartz, H. A., Park, G., Eichstaedt, J., Kern, M., Ungar, L., & Shulman, E. (2016, June). Modelling valence and arousal in Facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 9–15). San Diego, California: Association for Computational Linguistics.

Qu, L., Ferraro, G., Zhou, L., Hou, W., Schneider, N., & Baldwin, T. (2015, July). Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (pp. 83–93). Beijing, China: Association for Computational Linguistics.

Rabie, O., & Sturm, C. (2014). Feel the heat: Emotion detection in Arabic social media content. In *The International Conference on Data Mining, Internet Computing, and Big Data (BigData2014)* (pp. 37–49).

Rajamanickam, S., Mishra, P., Yannakoudakis, H., & Shutova, E. (2020, July). Joint modelling of emotion and abusive language detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4270–4279). Online: Association for Computational Linguistics.

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

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 704–714).

Rosenthal, S., Farra, N., & Nakov, P. (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 502–518). Vancouver, Canada: Association for Computational Linguistics.

Roy, A., & Pan, S. (2020, 7). Incorporating extra knowledge to enhance word embedding. In C. Bessiere (Ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (pp. 4929–4935). International Joint Conferences on Artificial Intelligence Organization.

Ruder, S. (2019). *Neural transfer learning for natural language processing* (Unpublished doctoral dissertation). NUI Galway.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161–1178.

Safaya, A., Abdullatif, M., & Yuret, D. (2020, December). KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 2054–2059). Barcelona (online): International Committee for Computational Linguistics.

Schick, T., & Schütze, H. (2020). Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, pp. 8766–8774).

Singh, T., & Kumari, M. (2016). Role of text pre-processing in Twitter sentiment analysis. *Procedia Computer Science*, *89*, 549-554.

Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). AraVec: A set of Arabic word embedding models for use in Arabic NLP. *Procedia Computer Science*, *117*, 256–265.

Subba, B., & Kumari, S. (2021). A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Computational Intelligence*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*(2), 267–307.

Top ten internet languages in the world - internet statistics. (2020). Retrieved from `https://www.internetworldstats.com/stats7.htm`

Wang, S., Liu, J., Ouyang, X., & Sun, Y. (2020, December). Galileo at SemEval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 1448–1455). Barcelona (online): International Committee for Computational Linguistics.

Wang, X., Jiang, W., & Luo, Z. (2016, December). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 2428–2437). Osaka, Japan: The COLING 2016 Organizing Committee.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 606–615).

Waseem, Z., Davidson, T., Warmsley, D., & Weber, I. (2017, August). Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 78–84). Vancouver, BC, Canada: Association for Computational Linguistics.

Yin, J., & Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 233–242).

Zahran, M. A., Magooda, A., Mahgoub, A. Y., Raafat, H., Rashwan, M., & Atyia, A. (2015). Word representations in vector space and their applications for Arabic. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (pp. 430–443). Cham: Springer International Publishing.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019, June). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 75–86). Minneapolis, Minnesota, USA: Association for Computational Linguistics.

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., ... Çöltekin, c. (2020). SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval.*

Zhang, Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), 12–53.

Zhang, C., & Abdul-Mageed, M. (2019a). *Multi-task bidirectional transformer representations for irony detection.*

Zhang, C., & Abdul-Mageed, M. (2019b, August). No army, no navy: BERT semi-supervised learning of Arabic dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop* (pp. 279–284). Florence, Italy: Association for Computational Linguistics.

Zhang, J., Liu, S., Li, M., Zhou, M., & Zong, C. (2014). Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 111–121). Baltimore, Maryland: Association for Computational Linguistics.

Zhang, Y., Roller, S., & Wallace, B. C. (2016, June). MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1522–1527). San Diego, California: Association for Computational Linguistics.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019, July). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441–1451). Florence, Italy: Association for Computational Linguistics.