Deep Learning for Arabic Sentiment Analysis

Ibrahim Abu Farha

s1758611

Master of Science

Artificial Intelligence

School of Informatics

University of Edinburgh

2018

Abstract

Sentiment analysis is one of the natural language processing applications (NLP), which aims to analyze and identify opinions. Analyzing opinions is very beneficial in many fields such as marketing and social studies.

Despite the advancements in NLP and sentiment analysis in other languages, sentiment analysis in Arabic still lacks behind due to the challenging nature of Arabic compared to other languages like English. While the recent revolution of deep learning improved the performance in many NLP tasks like machine translation and text generation, deep learning has not yet been well studied in Arabic sentiment analysis.

In this dissertation, we propose a wide variety of experiments of using deep learning models for Arabic sentiment analysis. These models include CNNs, LSTMs and different combinations and variations. The models were tested on SemEval's dataset where we achieved an average recall score of 0.60 beating the first place holder. Additionally, the models were tested on the newly published ArSAS dataset, setting a strong baseline for future work on Arabic sentiment analysis.

Furthermore, the experiments showed that the best results were achieved when using word embeddings alone, and that using the hand-engineered features either did not improve the results or sometimes worsened them. Moreover, during the work, a large set of word embeddings was created, which is considered to be the largest set of Arabic social-media-related embeddings.

Acknowledgements

I would like to thank my supervisor, **Dr. Walid Magdy**, for his guidance and continuous care. And I want to express my gratitude to my family; my mother, father and my brother **Yazan**, for always being there for me. And I wouldn't forget my friends, here and back home, for their helpful encouragement and support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Ibrahim Abu Farha*
s1758611)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentiment Analysis (SA) is one of the natural language processing (NLP) applications, which can be defined as the process of analyzing and identifying the polarity/opinion expressed in a piece of text, which might be from different sources such as tweets, products' reviews [Liu, 2012].

The emergence of social media platforms as a medium of communication and expression, and the widespread of their user-base, generated a huge amount of rich information that could be used to understand people's opinions and attitudes. For example, many companies rely on products' reviews in order to asses and plan their marketing and planning strategies. However, analyzing this data would require a large number of employees and would take a lot of time.

Sentiment analysis helps in automating the analysis process and understanding the nature of data and getting what it reflects. This would be very crucial for companies dealing with a large customer base. Additionally, SA helps in understanding people's attitudes towards some events or issues, which, in turn, is very helpful in many social studies. SA as a process can be utilized at different levels based on the target text, which can be a document, a sentence or an aspect/feature of a product/item [Kolkur et al., 2015].

Consequently, SA became under the spotlight and attracted many researchers and companies to work and tackle the problem of opinion/polarity classification. However, most of the work is focused on English as it is the most widely used language, where other languages still lack behind.

In this dissertation, we experimented with deep learning models for Arabic sentiment analysis, the models were compared with the participants of SemEval 2017 task 4-A [Rosenthal et al., 2017], and were tested on the newly published ArSAS dataset.

The best model was Bi-LSTM based one, which achieved an average recall of 0.60 on SemEval 2017 task 4-A datset, and 0.90 on ArSAS dataset.

1 Motivation

Sentiment analysis has many applications that can be very crucial in some cases, such as analyzing customers' opinions and helping to analyze the environment and responses to events. The vast majority of SA research is focused on English, while other languages including Arabic do still need more research.

In recent years, Arabic started getting more attention, due to the increase of Arabic content on the web, the active political situation in the Middle East since 2011, and the corresponding reactions on different social media platforms. This, in turn, attracted more researchers to work on Arabic NLP, including SA, which helped to understand the atmosphere and attitudes to the ongoing events.

Moreover, the world witnessed a strong revolution and a wide adoption of deep learning to solve many problems including NLP ones, which led to many breakthroughs. However, that was not the case for Arabic NLP due to lack of resources, where deep learning models have not been well-studied yet.

2 Problem Statement

As mentioned previously, sentiment analysis aims to analyze the polarity of an opinion provided in a piece of text. The goal of this project is to investigate and use deep learning techniques in order to build an Arabic SA system that targets analyzing data from Twitter. The project is focused on Twitter sentiment analysis and on classifying the polarity into positive, negative, or neutral. The main focus is to achieve state-of-the-art results on SemEval 2017 task 4-A dataset.

3 Contribution

During the work on this project, and through the thorough analysis and experimentation, the following contributions were achieved:

- Building the largest set of Arabic social-media-related word embeddings up to this time.

- Building an Arabic sentiment analysis system that is dialect independent and can handle different dialects.
- Evaluating different deep learning models for Arabic sentiment analysis task.
- Achieving performance results that beat the first place winner in SemEval 2017 task 4-A.

4 Dissertation Outline

The rest of the dissertation is organized as follows: **Chapter 2** gives a background on sentiment analysis and the approaches that are used for such task. **Chapter 3** provides a survey of the related work on this task in Arabic and other languages. **Chapter 4** introduces the detailed methodology that draws the outline for the experiments. **Chapter 5** goes over the resources that were used in the experiments, the details of the models and the evaluation metrics that are used to assess the models. **Chapter 6** introduces the results achieved combined with a thorough analysis of them. Finally, the conclusions and findings are reported in **Chapter 7** with some suggestions of future work.

Chapter 2

Background

This chapter gives background information about sentiment analysis process and approaches. In addition to that, it gives an overview about Arabic language, its features and the challenges it imposes.

1 Arabic Language

Arabic is one of the semitic languages in addition to Hebrew and Amharic. It is considered the national language of 28 countries with around 400 million native speakers [Darwish et al., 2014]. The importance of Arabic comes from the fact that it is one of the 6 formal UN languages and it is the language of Quran, the holy book of around 1.6 billion Muslims around the world.

There are three types of Arabic: Classical Arabic, Modern Standard Arabic (MSA) and Dialectical Arabic. Classical Arabic reassembles the language of the Quran, which is the old Arabic language, with many phrases that are not frequently used these days. MSA is the current unified form of Arabic which is taught in schools and used in media and news [Habash, 2010]. Dialectical Arabic (DA) is the colloquial language which is spoken in the streets, this language differs from one country to another, and even varies inside the country itself. DA differs from MSA in many things as it sometimes does not follow a specific grammar and it has many words that are pronounced differently. It also contains many words that are either borrowed from other languages or specific to that dialect.

Arabic imposes the following challenges for NLP researchers who are willing to work on it:

- **Dialects variety:** As mentioned previously, Arabic has many different dialects

that are considered regional and differ from one region to another. According to [Hamdi et al., 2016, Darwish et al., 2014] Arabic dialects can be put into 5 different groups based on the region as follows:

- Gulf: Saudi Arabia, United Arab Emirates, Oman, Kuwait, Qatar, Bahrain and Yemen.
- Iraqi: Iraq.
- Levantine:Palestine, Jordan, Syria and Lebanon .
- Egyptian: Egypt and Sudan.
- Maghribi: Morocco, Libya, Algeria, and Tunisia.

However, it is also worth mentioning that there are some slight differences between the dialects of countries within the same group [Alhumoud et al., 2015]. For example, the word "زاكي" (zaki) meaning "delicious" is used in Jordan and Palestine; whereas the word "طيب" (tyeb) is used in Lebanon and Syria. Additionally, sometimes dialects might have different ways of expressing negation, which might be different from MSA or other dialects. For example, in the Palestinian dialect a person would say "بعرفش" (baaraf-esh) which means "I don't know", while a Lebanese or a Syrian would say "ما بعرف" (ma baaraf). Negation in the Palestinian dialect is slightly similar to negation in old English as in saying (I know not).

- **Morphological complexity:** Arabic is considered one of the morphologically complex languages. Given a single root, it is possible to derive and inflect many word forms with different meanings [Abdul-Mageed et al., 2011].
- **Ambiguity:** compared to other languages, Arabic has an additional source of ambiguity coming from diacritics. The same word with different diacritics might have a completely different meaning. Since native speakers can easily infer the diacritics from the context, people usually do not write them, which makes the sentiment analysis task harder for an automated system. An example of such ambiguity is the word " كَتَبَ" (k[a]t[a]b[a]) which means "he wrote" and the word " كُتُب" (k[o]t[o]b) which means "books". However, both words are written as "كتب" , without diacritics [Zayyan et al., 2016].

- **Lack of resources:** one of the main challenges that faces researchers when working on Arabic is the lack of resources such as corpora and lexicons. This becomes more prominent when dialects are to be taken into consideration, because they differ from each other, which, in turn, requires specific resources for them in addition to MSA. [Refaee and Rieser, 2014, El-Beltagy and Ali, 2013]

2 Sentiment Analysis Process and Approaches

2.1 Sentiment Analysis Definition

In general, sentiment analysis aims to analyze and identify the polarity of the opinions/emotions in a given piece of text. Thus, sentiment analysis has two main tasks, the first is knowing whether a text has a sentiment, i.e. has a specific polarity; the other is knowing the polarity of the expressed opinion/emotion [Medhat et al., 2014]. Most of the research focused on the later task, specially with the immense growth of social media and other platforms that flood the world with a huge amount of textual data that can be utilized for many applications.

3 Sentiment Analysis Process

Generally, most of the sentiment analysis systems consist of five steps the are shown in Figure 2.1, these steps are common with many other NLP and text related applications.

Figure 2.1: Sentiment analysis steps.

3.1 Data Acquisition

This is the initial step for any data-driven system, it includes getting the data to be processed from the proper sources. For sentiment analysis, since it can be applied on many levels, many sources are very useful, such as social media posts and comments and products' reviews [Alessia et al., 2015].

3.2 Preprocessing

Preprocessing is an intermediate step that aims to prepare the raw data to be fed into the following steps. It seeks to get rid of noise, increase cohesion and to unify some aspects of the data, which might help to increase the performance [Altrabsheh et al., 2013]. Nevertheless, this is not always guaranteed.

There are many different prepossessing steps and some of them depend on the application, the most used ones are:

- **Spelling correction:** generally, data is noisy, specially when collected from social media, which increases the probability of having spelling errors. Additionally, people tend to use repeated characters to emphasize an emotion which can also be considered as a wrong spelling. Microsoft's ATKS [mic, 2013] is one of the available tools for automatic spelling correction.

- **Stop-word removal:** stop-words are connecting words which are used to combine and connect parts of the sentence or different sentences. Usually, they do not play a major role in the sentiment of a sentence, so they can be removed [Alajmi et al., 2012].

- **Punctuation removal:** generally, punctuation does not play a main role in the sentiment of a sentence, thus it can be removed [El-Makky et al., 2014].

- **Tokenization/segmentation:** an initial step to format the raw text and break it up into different sub-structures such as sentences, phrases, words and characters. It is widely used in almost all NLP applications. Some of the tools that can handle segmentation for Arabic such as Farasa [Abdelali et al., 2016], MADAMIRA [Pasha et al., 2014], and the segmenter in Stanford's NLP API [Monroe et al., 2014].

- **Stemming:** it is the process of cleaning the word and getting its root/base form. This is usually done by removing the added letters such as affixes and suffixes. For Arabic, due to the characteristics mentioned previously, it is more complicated to distinguish the original letters from the added ones [Duwairi and El-Orfali, 2014]. Farasa [Abdelali et al., 2016] and MADAMIRA [Pasha et al., 2014] are examples of available Arabic stemmers.

- **Letters' normalization:** Arabic has some letters that are very similar in shape and some people do not use them properly when they are typing and use a

similar-shape one, which leads to having many (wrong) forms for the same word. For example, the letters {إ ، أ ، آ} are usually substituted with { ا }, {ئ ، ىء} with {ي}, and {ة} with {ه} [Ahmed et al., 2013].

3.3 Feature Extraction

This step includes extracting and getting some characteristics and statistics about the data, which can be used as a distinguishing criteria to identify the polarity of a given text. Based on [Aggarwal and Zhai, 2012], some of these features are :

- **N-grams:** a set of features that relates to the words themselves individually or as groups. There are many ways to represent these features such as counts.
- **Part of speech (POS) tag:** POS tagging is the process of finding the type of a word such as nouns, verbs, adjectives, etc.
- **Negation:** the presence of negation in a sentence is very crucial, as it might cause the polarity to be the opposite of what is expected.
- **Opinion words/phrases:** these include the common words/phrases that are used to express emotions and opinions. Usually, it is easy to identify their polarity since they are found in a lexicon.

3.4 Sentiment Classification

This step is the core of any sentiment analysis systems, this section views the available methods and approaches for identifying the sentiment of a given text, Figure 2.2 shows a hierarchy of these approaches as explained in [Medhat et al., 2014].

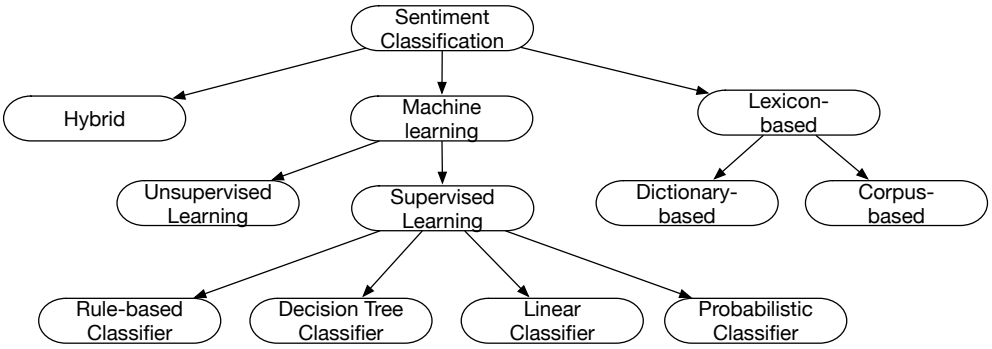

Figure 2.2: Sentiment classification approaches

Machine learning (ML) based methods are the most commonly used in sentiment analysis. They rely on the extracted features to find a relation between the features or a subset of them and the sentiment of the text. There are many methods that come under ML such as supervised and unsupervised learning [Alessia et al., 2015].

In general, ML approaches are supervised and they deal with sentiment analysis as a classification problem, where the goal is to find a proper sentiment label for a given piece of text. The number of labels varies from an application to another. However, people usually tend to have three labels: positive, negative and neutral.

All ML methods and specially the supervised ones heavily rely heavily on the availability of annotated data which will be used in the learning process. On the other hand, unsupervised methods tend to find the labels from the data itself through the correlations and patterns in its characteristics [Medhat et al., 2014]. It is worth mentioning that supervised learning is dominating in the sentiment analysis field where many algorithms are used such as:

- **Rule-based classifiers:** this type of classifiers relies on criteria that represent the set of rules used to distinguish between the different classes/labels. Rule-based classifiers differ from decision trees as they allow overlapping between decision spaces, while decision trees tend to have a hierarchical splitting [Quinlan, 1986].

- **Decision tree classifiers:** they recursively split the decision space hierarchically, based on a set of rules and conditions, the process is repeated until a leaf is reached where the classes are pure or the stop condition is satisfied [Quinlan, 1986].

- **Linear classifiers:** this type of classifiers finds linear decision boundaries in the feature space of the data. They include many algorithms such as support vector machines (SVM) and single layer neural networks. SVMs work through finding a decision boundary, which is the hyper-plane that separates two classes [Cortes and Vapnik, 1995]. Additionally, they can handle non-linear boundaries by using non-linear mapping from a space to another [Aizerman et al., 1964].

- **Probabilistic classifiers:** this type tries to give a specific probability to each given class and thus using it in the prediction process. Naive Bayes, Bayesian Networks and Maximum Entropy are examples of this type of classifiers [Medhat et al., 2014].

Lexicon based approach relies on the availability of annotated lists. These lists contain sentiment words/phrases that have a specific labeled polarity. The disadvantage of this approach is that it is highly language-dependent and will not cover all the possible phrases. The construction of these lists can be done either manually or automatically using one of the following approaches [Alessia et al., 2015]:

- **Dictionary based:** this approach in building lexicons starts by using a seed-list of opinion words that is expanded through adding synonyms of the words using a WordNet [Mohammad et al., 2009].
- **Corpus based:** this approach starts with a list of seed opinion-words and it seeks to expand the list through searching in a corpus for other opinion-words based on some rules [Lafferty et al., 2001].

Hyprid approach combines the aforementioned approaches altogether. It relies on a large variety of features that include lexical based ones, and it relies on ML algorithms to perform the classification [Alessia et al., 2015].

4 Deep Learning

In recent years, the huge leap in the computational power led to a huge development in the machine learning/deep learning methods. Deep learning started solving many problems specially in the computer vision community and many NLP researchers started adopting these models and techniques to tackle NLP related tasks and applications [Manning, 2015].

4.1 Convolutional Neural Networks (CNNs)

These are useful when the task is dependent on or related to the local patterns of the items, they were introduced by [LeCun et al., 1995] and were used for digits recognition tasks. They were introduced to the NLP community in [Collobert et al., 2011] who utilized them for the semantic role labelling task. When using CNNs for NLP related applications, the main concern is the convolution in a single dimension over the input. CNNs are designed to capture the local aspects and features in the larger structure, thus they work as features extractors [Goldberg, 2016].

4.2 Recurrent Neural Networks (RNNs)

Language is sequential and context-dependant, thus the ordering of words has a significant importance. While CNNs can capture and extract features from local patters, recurrent neural networks (RNNs) have the capability to capture more global features that relate to words' ordering. RNNs were first introduced in [Elman, 1990] but the vanilla RNN has the problem of not capturing long dependencies, which led to the appearance of other variants such as long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997]. These days LSTMs and other variations of RNNs are achieving state-of-the-art results in many NLP related tasks [Goldberg, 2016].

Chapter 3

Related work

In this chapter, we summarize the related work that was conducted on Arabic and other languages in order to put the work into its proper context.

1 Sentiment Analysis in Other Languages

Literature is flooding with lots of work on sentiment analysis, however most of the work is focused on English as it is the most widely used language.

Many people started working on sentiment analysis, and in their work they utilized one or more of the previously mentioned approaches. In [Pang et al., 2002], the authors aimed to analyze movies' reviews from IMDB dataset. In their work, they used a set of hand-engineered features and experimented with many classifiers such as Naive Bayes, SVM and Maximum Entropy. Additionally, other researchers started taking into consideration sentiment analysis of social media such as Twitter. For example, in [Go et al., 2009] they used the set of classifiers used by Pang et al. with some additional hand-engineered features that rely on the nature of the data such as URLs, hash-tags and usernames. In their work, SVM was the best model with an accuracy of 82%.

In [Pak and Paroubek, 2010], the authors also utilized a set of features with some linear classifiers. In their work they tried to improve the results by removing the common n-grams, because they will not be informative in the sentiment classification process. In [Kouloumpis et al., 2011], the authors incorporated some lexical and linguistic features in addition to the usual n-grams, URLs, etc. They conducted different experiments with different mixtures of features.

[Batra and Rao, 2010] focused on entity-based sentiment analysis, where the sentiment is analyzed relative to a specific entity. They built their model based on labeled

movies' reviews and used it to analyze tweets. [Di Caro and Grella, 2013] introduced a sentiment analysis model that is context-aware through utilizing a set of rules at syntactic level in the dependency parse tree.

Since 2013, SemEval competition [Nakov et al.,] included different tasks related to sentiment analysis in English, and it had many participating teams. The model by [Mohammad et al., 2013] was the winner in 2013, they utilized a wide variety of features with the use of SVM as a classifier. The task was a message level polarity classification and they achieved an F_1 of 69.02 beating the other 43 teams. In SemEval 2014 [Rosenthal et al., 2014], [Miura et al., 2014] won the first place. In their work, they utilized many features including lexical ones and logistic regression as a classification method, they achieved an average F_1 of 70.96 for the message polarity classification task. The second place winner used a deep learning model that utilizes word embeddings, in addition to hand-engineered features. They achieved an average F_1 score of 70.14 [Tang et al., 2014].

The winner in 2015 [Rosenthal et al., 2015] used an ensemble classifier, which is based on different approaches used by previous winners, they achieved the first place with an F_1 score of 64.84 [Hagen et al., 2015].

The system created by [Deriu et al., 2016] was winner of task 4 in SemEval 2016 [Nakov et al., 2016] for the massage polarity classification sub-task. The authors used a deep learning model, which was based on a 2-layer convolutional neural network (CNN) and achieved an F_1 score of 63.30 on the test set provided by the organizers. Moreover, the second place holder [Rouvier and Favre, 2016] , proposed a CNN-based model, which relies on word embeddings and other features; they achieved F_1 score of 63.0.

[Baziotis et al., 2017] was the winner of SemEval 2017 [Rosenthal et al., 2017] for the English message polarity classification task. They used a deep learning model based on long short-term memory (LSTM) combined with attention mechanism, they achieved an average recall of 68.11. Additionally, [Cliche, 2017] achieved a similar result, they used a model that combines CNNs and LSTMs.

2 Sentiment Analysis in Arabic

Similar to the work on other languages, literature of Arabic sentiment analysis contains many attempts that used a wide variety of the previously mentioned approaches. The most commonly used classifiers for Arabic are SVM and Naive Bayes that rely on

hand-engineered features that are based on statistical calculations and lexicons [Alessia et al., 2015].

In [Al-Smadi et al., 2017], the authors compared the performance of SVM against an RNN-based model in building an aspect-based sentiment analysis system. They tested the model on a dataset for Arabic hotels' reviews. In their approach, they used a combination of lexical, syntactic, semantic and morphological features. Their results showed that SVM , which achieved an accuracy of 95%, was better than the RNN model, which achieved accuracy of 87%, for that specific task.

In [Al-Ayyoub et al., 2015], the authors built a lexicon-based sentiment analysis system that used their own lexicon. The model was tested on a manually collected labelled tweets, they achieved an accuracy of 87%.

The authors of [Soliman et al., 2014] targeted social media and they tried to handle the dialects variety through building their own lexicon, namely slang sentimental words and idioms lexicon (SSWIL). They utilized the lexicon with SVM classifier to perform polarity classification and they achieved an accuracy of 87%. Additionally, [Abdulla et al., 2013] collected their own dataset which consists of 2000 tweets and they experimented with different sentiment analysis approaches, their best model was SVM with an accuracy of 87%.

In [Dahou et al., 2016], the authors proposed a set of Arabic word embeddings to be used for Arabic sentiment analysis, the corpus they used was collected from the web and it contains around 3.4 billion words. Additionally, they built a CNN-based sentiment analysis system that utilized the newly created embeddings and they tested it on LABR book reviews dataset [Aly and Atiya, 2013], Arabic Sentiment Tweets Dataset (ASTD) [Nabil et al., 2015] and other datasets.

Another word embeddings set was proposed in [Altowayan and Tao, 2016]. The authors used the embeddings as features to be fed to the classifier. They experimented with different classifiers and the best was SVM.

In [Alayba et al., 2017], the authors proposed a new dataset for opinions on health services, which was collected from Twitter. They experimented with different sentiment analysis approaches on the new dataset, their experiments included SVM, Naive Bayes and CNNs. The best classifer was SVM with accuracy of 91%.

In [Al Sallab et al., 2015], the authors conducted different experiments on many deep learning models such as recursive auto-encoder (RAE), deep belief networks (DBN) and deep auto-encoder (DAE). In their work, they relied on the bag of words (BoW) representation of text and some lexical features.

In SemEval 2017, Arabic was added to task-4 [Rosenthal et al., 2017] which is about sentiment analysis. For the sub-task A, which is polarity classification, the winner was NileTMRG team [El-Beltagy et al., 2017]. In their work, they used a large set of hand-engineered features that covers a large variety of syntactic, lexical and statistical features. They used a complement Naive Bayes classifier and they achieved an average recall of 0.583. The runner up was SiTAKA team [Jabreel and Moreno, 2017], they used a combination of features such as bag of words and lexical features. Moreover, they introduced some features that were calculated from the word embeddings such as sum, min, max and standard deviation. The classifier of choice was SVM and they achieved an average recall of 0.55.

Recently, [Alayba et al., 2018] experimented with deep learning models for Arabic sentiment analysis. In their work, they built a model that is based on a combination of CNN and LSTM. They tested their model on different datasets such as Twitter dataset (Ar-Twitter) and Arabic Health services dataset. The final model achieved an accuracy of 88.1% and 94.3% on the datasets respectively.

Furthermore, [Al-Smadi et al., 2018] proposed an aspect-based sentiment analysis system, where they created a model based on a character-level Bi-LSTM combined with conditional random field (CRF) that was responsible of extracting the aspect opinion target expression, for the sentiment classification they used an LSTM based model. They tested their models on the Arabic hotels' reviews dataset where they had an enhancement of around 39% with an F-score of roughly 70%.

Chapter 4

Methodology

This chapter provides an overview of the methodology and steps that are used in the different models and the reasoning behind each of them.

In this work, as we are aiming to improve and compare our results on SemEval 2017 task 4-A dataset. We started our work by building a similar system to the one which won the first place, the NileTMRG team from Nile University [El-Beltagy et al., 2017]. In their work, they used a combination of the features and preprocessing steps that were used before in Arabic SA literature. However, it is worth mentioning that we could not build the same model since we could not reach all the resources that were used in the original one. After that we go over our contribution which is the utilization of deep learning models to achieve better results.

1 Data Preprocessing

In general, this step is an initial step that aims to reduce the noise and normalize the data into a consistent form so it can be handled easily. Figure 4.1 shows the steps that are used.

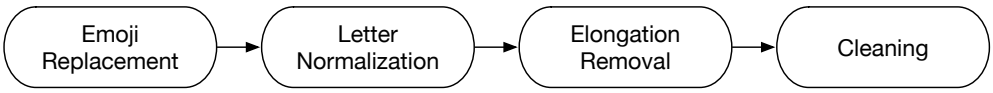

Figure 4.1: Preprocessing steps.

Most of the above steps are commonly used in Arabic SA and Arabic NLP systems in general. The detailed explanation of each one of them is as follows:

- **Emoji replacement:** this step includes matching the input with an available

list of emojis, which are labelled based on their polarity to positive or negative, this list was created by [El-Beltagy et al., 2016]. The list contains 105 negative emojis and 110 positive ones. In the implementation, when an emoji is matched it is replaced by a specific term that is out of the Arabic vocabulary, the term is used to identify if the emoji is positive or negative.

- **Letter normalization:** this step is also widely used in Arabic NLP as it aims to unify the letters that can appear in different forms. In the implementation we replace {إ ، أ ، آ} with {ا}, {ة} with {ه} and {ى} with {ي}.

- **Elongation removal:** sometimes, specially on social media, people tend to repeat a character for emphasis or showing a strong emotion. In this step, these letters are removed and the word is reduced into its standard form. For example, in English people might say "yesssss". In the implementation, all the repeated letters are reduced to one.

- **Cleaning:** this step contains general cleaning of numerical data, URLs, punctuation and diacritics. This step is needed when the aim is to have a representation of the individual words or creating the text representation.

2 Feature Extraction and Text Representation

This section explains the features that are used for the sentiment classification step. Moreover, we go over different word/text representations, in addition to the creation of a new set of word embeddings.

2.1 Text Representation

In this section we go over the possible ways to represent the available text in order to be handled and used for the classification task, these include n-gram features and word embeddings.

2.1.1 N-gram Features

This representation depends on the individual words and some statistical calculations and counts. This representation depends on a specific window of words to deal with. For example, unigram models deal with single words while bigram models deal with 2

words. In the proposed system, we used term frequency-inverse document frequency (TF-IDF) vectors to represent each of the available tweets. Term frequency (TF) [Luhn, 1957] is the frequency of a word in a document, which is the tweet in our case. Inverse document frequency (IDF)[Sparck Jones, 1972] is a weighing term that aims to give a higher weight for the less frequent words. IDF weight for a word/term is calculated using the following equation:

$$idf_i = log(\frac{N}{df_i}) \tag{4.1}$$

where N is the number of documents(tweets) and df_i is the number of tweets that contain the word/term. The TF-IDF representation of a tweet is a vector that contains the TF-IDF weights for the words/terms in the vocabulary where each weight w_i is given by:

$$w_i = tf_i \times idf_i \tag{4.2}$$

In the implementation, we use both unigrams and bigrams to represent of the tweets.

2.1.2 Word Embeddings

The disadvantage of n-gram-based representations is that they deal with words as atomic units. This implies that there is no notion of similarity between words, as these are represented as indices in a vocabulary [Mikolov et al., 2013].

In [Mikolov et al., 2013], the authors introduced an efficient way to create word embeddings which are dense representations of words as vectors. This dense representation manages to capture the meaning and the semantics more robustly, as many tests showed that semantics is encoded by the distance in the embedding space. For example, similar words such as “coffee” and “tea” are mapped to nearby vectors in the embedding space.

In the original paper, the authors introduced two architectures to create the word representation: continuous bag of words (CBOW) and skip-gram, which are shown in Figure 4.2. Both of the models are based on feed-forward neural language models, where the non-linearity is removed and the projection matrix is shared. CBOW model builds a word representation through using the context to predict the word, whereas skip-gram model builds the representation of a word through predicting its context.

In this work, the skip-gram model was used in the process of building a new word embeddings for Arabic, which is based on content collected from Twitter. In the creation process, a large corpus of 100M Arabic tweets was utilized, which is larger than any set used to create Twitter-related word embeddings such as Aravec [Soliman et al.,

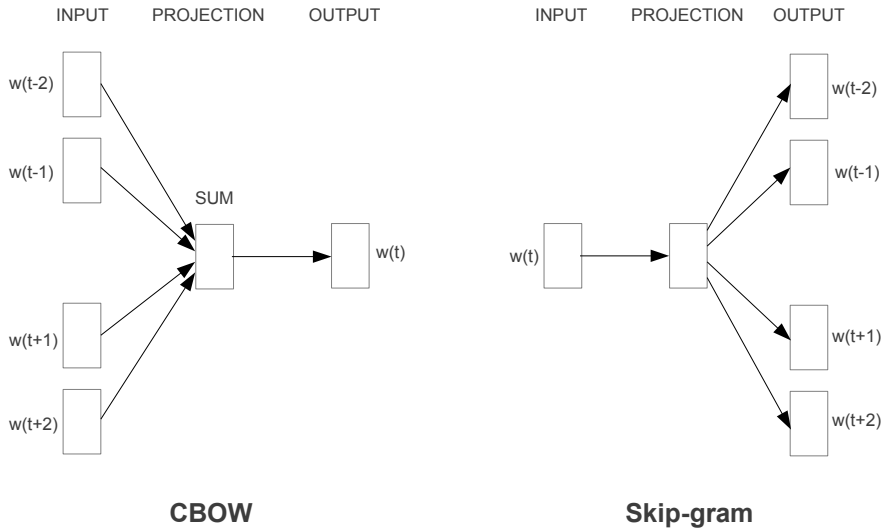

Figure 4.2: word2vec models proposed by Mikolov. et al.

2017], which was built using around 67M tweets. The tweets were collected over different time periods to ensure the variety of topics.

Before creating the embeddings, in order to normalize the tweets and clean them, the following preprocessing steps were applied:

- Duplicate tweets removal.
- Diacritics removal.
- URL removal.
- Punctuation removal.

Table 4.1: Twitter corpus statistics.

#tweets	100,000,000
#words	1,223,242,711
#unique words	4,347,845

Table 4.1 shows the statistics of the corpus used in the embedding creation. The size of the output vectors was set to 300, which was used by many other researchers.

Tweets are represented as NxD matrix, where N is the number of words and D is the dimension of the embeddings (D=300). Each row in this matrix contains the embedding of the corresponding word.

2.2 Hand-Engineered Features

There are many different features that can be extracted and used for the classification process. In this work, the following features are used and they are based on [El-Beltagy et al., 2016, El-Beltagy et al., 2017]:

- **StartsWithLink:** a binary feature that is set to 1 if the tweet starts with a link, 0 otherwise.
- **EndsWithLink:** a binary feature that is set to 1 if the tweet ends with a link, 0 otherwise.
- **Length:** an integer that can take on of the following values $\{0, 1, 2\}$. This feature represents the length of the tweet, where it takes the value $\{0\}$ if the tweet is less than 60 characters, $\{1\}$ if it is less than 100 characters and $\{2\}$ otherwise. The datasets that are used in the experiments were collected before Twitter changed the maximum character count to 280 instead of 140.
- **Number of segments:** an integer that represents the number of segments separated by any of the following characters "?,;!—".
- **StartsWithHash:** a binary feature that is set to 1 if the tweet starts with a hashtag, 0 otherwise.
- **EndsWithQuestion:** a binary feature that is set to 1 if the tweet ends with a question mark (?), 0 otherwise.
- **Number of positive emojis:** represent the count of positive emojis in the tweet.
- **Number of negative emojis:** represent the count of negative emojis in the tweet.
- **Number of positive terms:** the count of the positive terms in the tweet. In order to give more weight to the compound terms, weighting was applied using the equation:

$$numOfPos = \sum_{i=0}^{n} i + \sum_{j=0}^{c} j \times \alpha \tag{4.3}$$

  where n is the number of single-word positive terms, c is the number of positive compound terms and α is a weighting factor and $\alpha > 1$.
- **Number of negative terms:** a real number that represents the score of negative terms, it is calculated using equation 4.3.

- **Ends with positive terms:** a binary feature that is set to 1 if the tweet ends with a positive term, 0 otherwise.
- **Ends with negative terms:** a binary feature that is set to 1 if the tweet ends with a negative term, 0 otherwise.
- **PosPercentage:** a real that number represents the percentage of the positive terms with respect to the total number of terms in the tweet.
- **NegPercentage:** a real that number represents the percentage of the negative terms with respect to the total number of terms in the tweet.

3 Sentiment Classification

This section will go over the different algorithms and methods that were used for the sentiment classification step in the general system. Here, we go over some models that use classical machine learning algorithms, and then we go further to deep learning models, which are the core of this project.

3.1 Classical Machine Learning

This type of models follows the general flow of the SA process, which is shown previously in Figure 2.1. After the preprocessing step, the features explained previously are extracted in addition to the n-gram text representation, in which TF-IDF vectors were used. Both the features and the TF-IDF vectors are concatenated together, which produces a sparse vector representation of the given tweet.

In the implementation, two classifiers were used SVM and Naive Bayes. The reason of this choice is that both algorithms were used extensively in the NLP literature, specially Arabic NLP. SVM is the most commonly used classifier in Arabic SA, for that reason, two variants were used the normal SVM and the SVM combined with non-linear transformation.

3.2 LSTM Model

Language is context dependent and words' ordering is very important, which has to be taken into consideration, because different word orderings might lead to different

meanings. In the previous model, the ordering of the words was not taken into consideration as the tweets were represented as vectors with weights for each word in the vocabulary.

The use of word embeddings with the utilization of long short-term memory (LSTM) networks is very useful in NLP. LSTMs tend to capture long term dependencies between the sequential inputs and thus capturing information that can represent the meaning of the tweet/sentence.

The tweets are fed into the LSTM word by word, and the output of the LSTM is connected to a softmax output layer that produces the output probability of each of the classes, Figure 4.3 shows the model used.

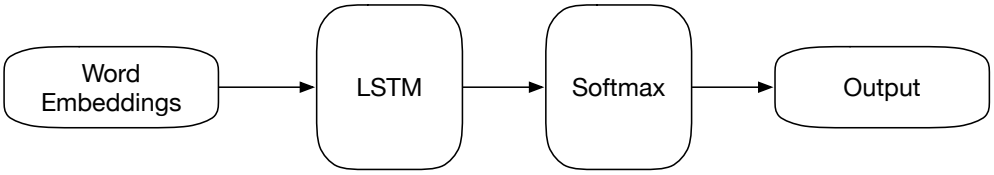

Figure 4.3: LSTM model architecture.

3.3 Bi-LSTM Model

LSTMs tend to capture the dependencies in one direction, and sometimes it might lose important information, here where bidirectional LSTMs (Bi-LSTM) are useful. Bi-LSTMs can be viewed as two LSTMs but each one of them is going over the input in a different direction. This would help because at any point the network would have information about the sequence from the beginning to that point, and from the end of the sequence to that specific point, which represents the entire context. The detailed model is shown in Figure 4.4.

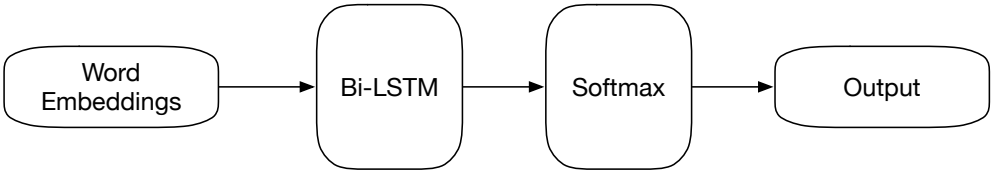

Figure 4.4: Bi-LSTM model architecture.

3.4 CNN Model

CNNs are heavily used in image-related applications such as image classification. The reason to consider CNNs is that they are very good at capturing correlations and pat-

terns in the data, which might be useful, because tweets are represented as concatenation of words' vectors, and related words would have correlated vectors. These correlation might work as features to distinguish between different classes.

Figure 4.5 shows the detailed model. The word embeddings are fed to the 1D convolutional layer which has many filters that work as feature-maps and will be learned during the training, then a max-pooling layer is used to reduce the dimensionality and take the max feature within a specific window. Then a dense layer is used learn from the newly extracted features and at the end there is a softmax layer.

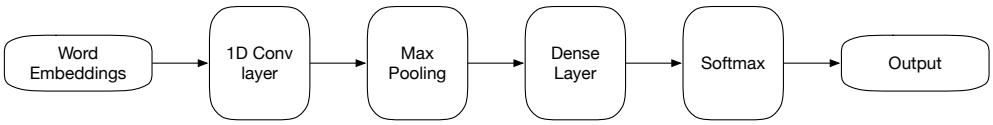

Figure 4.5: CNN model architecture.

3.5 CNN-LSTM Model

This model combines the aforementioned ones, where the model starts with a CNN network followed by an LSTM layer, as shown in Figure 4.6. The motivation to have such a network is that the CNN could learn more features that are not expressed by the embeddings and thus the CNN works as a feature extractor. Consequently, the LSTM will work on the features extracted from the CNN and capture dependencies in the produced sequence.

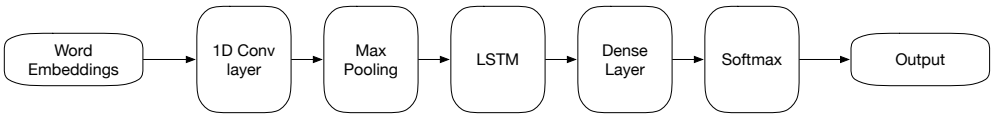

Figure 4.6: CNN-LSTM model architecture.

3.6 LSTM-CNN Model

This model is very similar to the previous one, but here the LSTM comes before the CNN as shown in Figure 4.7. The reason to have such model is that the LSTM might work as a feature extractor and learn things that the CNN in the previous model might not learn. So in this architecture, the LSTM is the feature extractor while the CNN works as a learner and tries to capture correlations between the learned features.

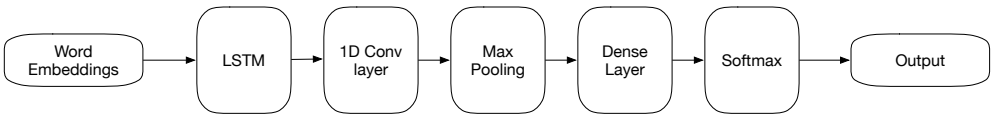

Figure 4.7: LSTM-CNN Model.

3.7 Mixed Model

This model is a mixture of the good of the two worlds; the word embeddings combined with deep learning world and the hand-engineered features world. Figure 4.8 shows the detailed model, this model aims to combine the features and dependencies learned by the Bi-LSTM and combine them with features that were manually extracted from the tweets. Both outputs are concatenated and fed into a dense layer followed by a softmax layer. The reason to have a concatenation is to ensure a combined training of both models and thus some new signals might appear and the model would learn new features.

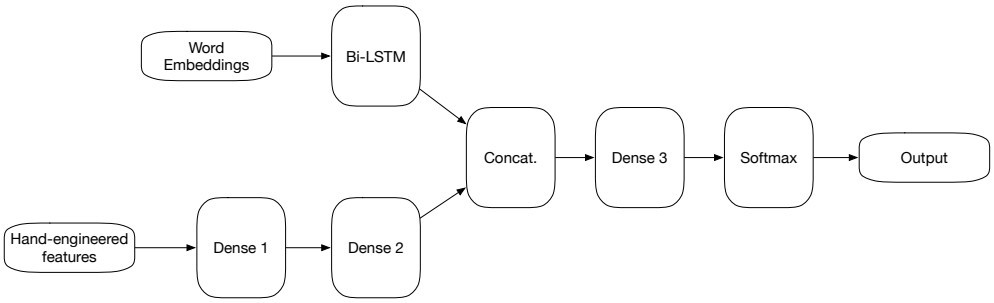

Figure 4.8: Mixture of Bi-LSTM and feed-forward model architecture.

Chapter 5

Experimental Setup and Evaluation

1 Datasets and Lexicons

This subsection provides information about the datasets and lexicons which will be used and utilized in the different experiments.

1.1 SemEval 2017 Task 4-A Dataset

This dataset is one of the datasets provided in SemEval-2017. It was special for task 4-A, which is about predicting the sentiment of tweets and classifying them to positive, negative or neutral [Rosenthal et al., 2017]. The data is provided as two sets, a training set of 3,555 tweets and the test set that contains 6,100 tweets. Moreover, the organizers provided a validation set of 671 tweets, Table 5.1 shows the statistics of the dataset. The training set was collected over the period September-November 2016, while the test set was collected between December 2016 - January 2017. The tweets were collected through specifying some topics that were prominent at the time of collection, and the authors made sure that the topics in the training set are different than the ones in the test set.

Table 5.1: SemEval 2017 Task 4-A dataset statistics.

Set	Positive	Negative	Neutral	Total
Training	743	1,142	1,470	3,555
Validation	222	128	321	671
Testing	1,514	2,222	2,364	6,100

1.2 ArSAS Dataset

ArSAS is a manually annotated datast for Arabic speech-act and sentiment analysis. Currently, it is considered the largest dataset for Arabic SA, as it contains around 21K tweets. Additionally, the tweets cover many different topics and most of them are in dialectical Arabic. The data was manually annotated using CrowdFlower crowdsourcing platform. The annotation scheme for the sentiment analysis task was 4-way classification, as each of the tweets is labelled with one of the following: positive, negative, neutral, or mixed [Elmadany and Hamdy Mubarak, 2018]. The data was collected form Twitter from the 1st to the 15th of November 2017, the authors originally collected around 62,000 tweets and applied some filtering until they had 21,064 tweets at the end. The collected tweets were related to topics that were of importance at that time, they collected the data through specifying some long standing topics, new events that were happening at that time and some entity-related tweets such as celebrities.

Table 5.2 shows the dataset statistics with the specific counts of each class, while Figure 5.1 , which is taken from [Elmadany and Hamdy Mubarak, 2018], shows the percentages of the classes.

Table 5.2: ArSAS dataset statistics.

Positive	Negative	Neutral	Mixed
4,643	7,840	7,279	1,302

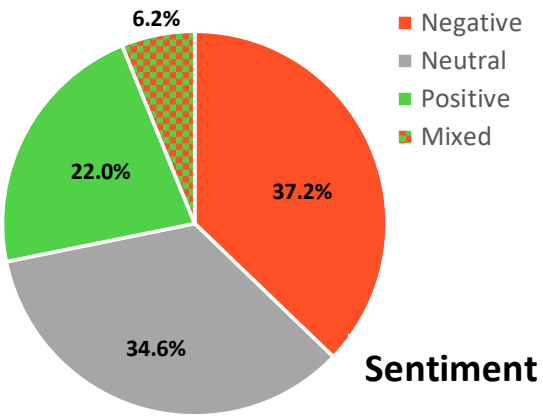

Figure 5.1: ArSAS tweets' sentiment distribution.

1.3 Nile University (NU) Dataset

The dataset consists of 3,436 tweets that are mostly in the Egyptian dialect, and were collected through Twitter's API in December 2014. Each of the tweets were annotated manually by 3 different graduate students at Nile University into six categories: positive, negative, neutral, mixed, sarcastic, and ambiguous. The final labels were chosen through a majority voting between the different annotators. Finally, tweets with positive, negative or neutral tags were kept for the final dataset [Khalil et al., 2015]. The dataset is split into training and testing sets with the statistics shown in Table 5.4.

Table 5.3: NU dataset statistics.

Set	Positive	Negative	Neutral	Total
Training	1,048	974	728	2,750
Testing	264	229	193	686

1.4 NileULex Lexicon

This lexicon contains around 6000 sentiment terms that are taken from the Egyptian dialect and MSA. The lexicon was released in 2013, and in the following two years more terms were added and many were revised. The lexicon contains a large variety of sentiment terms where 55% of them are MSA and the other 45% are from the Egyptian dialect [El-Beltagy, 2016], Table 5.4 shows the lexicon details.

Table 5.4: NileULex statistics.

	Positive	Negative	Total
Single term	1,281	3,693	4,974
Compound term	563	416	979
Total	1,844	4,109	5,953

2 Evaluation Metrics

The sentiment analysis is a classification task. In most of the literature, classification problems are assessed using the following measures: accuracy, precision, recall and F-score. The following equations show how to calculate the metrics based on the confusion matrix shown in Table 5.5.

Based on [Kohavi and Provost, 1998], these metrics are defined as follows:

- True positive: is the proportion of the positive class that was correctly classified.
- True negative: is the proportion of the negative class that was correctly classified.
- False positive: is the proportion of the negative class that was wrongly classified as positive.
- False negative: is the proportion of the positive class that was wrongly classified as negative.
- Precision: is the proportion of the positive items that were correctly classified, from all the items that were classified as positive.
- Recall: is the proportion of the positive items that were correctly classified as positive.
- Accuracy: is the proportion of the items that were correctly classified to their classes.
- F-score: is the harmonic mean of the precision and recall.

Table 5.5: Confusion matrix

Predicted Class			
A	B		
True Positive (TP)	False Negative (FN)	A	Actual Class
False Positive (FP)	True Negative (TN)	B	

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{5.2}$$

$$Precision = \frac{TP}{TP+FP} \tag{5.3}$$

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \tag{5.4}$$

In SemEval-2017, the organizers adopted the average recall (*AvgRec*) as a primary metric for task 4-A, sentiment classification, due to its robustness against the imbalances of the classes, the following equation is used:

$$AvgRec = \frac{1}{3}(R^P + R^N + R^U), \tag{5.5}$$

Where R^P, R^N, R^U are the recall for positive, negative and neutral classes. In addition to that, macro average $F_1^{PN}$was the secondary metric after *AvgRec*, F_1^{PN} is calculated using the following equation:

$$F^{PN} = \frac{1}{2}(F_1^P + F_1^N), \tag{5.6}$$

Where F_1^P, F_1^N are the F_1 with respect to the positive and negative classes.

3 Experimental Setup

In the implementation, Python was the language of choice because it has a large variety of supporting APIs that would make the work flow easier. For the machine learning part, the focus was on *sklearn* API, which provides a variety of machine learning algorithms and methods. For the deep learning experiments, *Keras* was used on top of *Tensorflow* [Abadi et al., 2015] back-end.

In the classical machine learning experiments, the classifiers were used with their automatic configuration provided in *sklearn*. Regarding the deep learning experiments, the hyper-parameters are shown in Table 5.6. For all the experiments, the activation was ReLU and the optimizer was Adam with learning rate of 0.0001.

Table 5.6: Hyper-parameters used for deep learning models, the number after the (/) symbol is the value when the model was trained on the extended SemEval dataset.

Model	***#LSTM cell***	***recurrent dropout***	***output dropout***	***#filters***	***filter size***	***pooling size***	***#hidden units***
LSTM	128/256	0.2	0.2	-	-	-	-
Bi-LSTM	64	0.2	0.2	-	-	-	-
CNN	-	-	-	300/200	3	2	256
CNN-LSTM	128/256	0.2	0.2	300	3	2	128/256
LSTM-CNN	100/128	0.2	0.2	32/64	3	2	-
Mixed	128	0.2	0.2	-	-	-	100

Chapter 6

Results and Discussion

This chapter will go over and analyze the results achieved by the models proposed in the methodology chapter.

In all of the experiments, the goal was primarily to achieve better results on SemEval 2017 task-4A dataset. Additionally, for each of the experiments, the best configuration that achieved the best results on SemEval was used to report the results on ArSAS dataset. In all of the experiments, the main goal was to achieve the highest average recall as a primary metric, but the accuracy and F^{PN} are also reported. Moreover, since the training set for SemEval dataset is considered to be small, all the experiments were conducted in two ways, the first is by using the original training set and the second is by using extra training data from ArSAS and NU datasets.

1 Results on SemEval

Table 6.1 shows the scores achieved by the top three participants in SemEval 2017 task 4-A [Rosenthal et al., 2017]. From the table, the highest *AvgRec* of 0.58 was achieved by the NileTMRG team [El-Beltagy et al., 2017].

Table 6.1: Top three teams in SemEval 2017 task 4-A.

Team	***AvgRec***	***FPN***	***Accuracy***
NileTMRG [El-Beltagy et al., 2017]	**0.58**	**0.61**	**0.58**
SiTAKA [Jabreel and Moreno, 2017]	0.55	0.57	0.56
ELiRF-UPV [González et al., 2017]	0.48	0.47	0.51

Table 6.2 shows the experiments' results on SemEval dataset. As mentioned previously, each of the models was trained twice, one on the original training data and the other is after adding extra data form ArSAS and NU datasets.

From the classical ML experiments, we noticed that the results were best when using Naive Bayes. However, most of the better results in the literature were achieved by SVM.

Regarding the deep learning experiments, a person can notice the large improvement on the results after using the dense word representation, i.e. word embeddings. This means that the embeddings could encode more information compared to the sparse vectors used with the hand-engineered features.

Moreover, most of the models did very well on both datasets, but a person can notice that the peak average recall score on SemEval's test set is around 0.60, and all the models achieved that score or slightly less than that. These scores are considered to be excellent as they already beat or give similar scores to the winner of the first place in the competition. Additionally, these results were achieved only by using the word embeddings and without relying on any lexical features, which means that the model can , to some extent, handle different Arabic dialects.

Table 6.2: Results on SemEval 2017 task 4-A dataset.

Approach	***original training set***			***extended training set***		
	AvgRec	***FPN***	***Accuracy***	***AvgRec***	***FPN***	***Accuracy***
[El-Beltagy et al., 2017]	0.58	0.61	0.58	-	-	-
[Jabreel and Moreno, 2017]	0.55	0.57	0.56	-	-	-
[González et al., 2017]	0.48	0.47	0.51	-	-	-
SVM	0.33	0.0	0.39	0.33	0.0	0.39
Naive Bayes	0.39	0.17	0.44	0.48	0.44	0.39
NuSVC	0.31	0.27	0.34	0.36	0.05	0.40
LSTM	**0.60**	0.61	**0.63**	0.59	**0.63**	0.58
Bi-LSTM	**0.60**	**0.62**	0.62	0.58	0.59	0.61
CNN	0.59	0.61	0.60	0.59	0.59	0.61
CNN-LSTM	0.58	0.59	0.61	**0.60**	0.61	**0.62**
LSTM-CNN	0.58	0.58	0.60	0.59	0.61	0.60
Mixed	**0.60**	0.61	0.62	**0.60**	0.62	**0.62**

Another thing that could be noticed, that with deep learning models, adding more data did not help to improve the results. This phenomenon raises a question about the similarities between the natures of the original training set and the added data.

So as a way of investigation, we decided to check the similarity between the structure of the language in all of the sets. One way to do that is by building a language model on a training data, and reporting the perplexity on the test set. Having lower perplexity value means that the language of the test is closer to the training set [Chen et al.,].

So we built two language models, one based on the original training data, and the other on the extra data that we added. Table 6.3 shows the perplexities of the two models, the perplexity by itself does not mean anything, but here since we are comparing two models it is better to have lower perplexity.

Based on the observation of the perplexities, a person can conclude that there is a difference in the language structure between the dataset, which could be due the collection process as they might have been collected from different locations and thus having different dialects.

Table 6.3: Perplexities of language models on SemEval's test set.

Training corpus	***Unigram LM perplexity***	***Bigram LM perplexity***
SemEval training set	**9,063**	**12,090**
ArSAS+NU	14,880	25,286

Moreover, since the model's performance did not change with the addition of extra data that is from a different dialect, we can conclude that the model can handle different dialects given proper training data.

2 Results on ArSAS

Table 6.4 shows the results achieved on ArSAS dataset. It is clear that the results are higher than the ones achieved on SemEval, which implies that this dataset is easier. Moreover, from these experiments, one can conclude that Naive Bayes is the best classifier to be used with the proposed feature-set, since it gives the highest results on both ArSAS and SemEval.

Additionally, the deep learning models give a huge jump in the results, most of the models achieved an average recall of 0.90. In addition to that, a person might conclude that ArSAS is considered to be an easier dataset compared to SemEval's as most of the deep learning models achieved high results compared to the 0.60 peak on SemEval. However, this is not true as the reason of the large discrepancy between the results on the datasets is the collection process itself. In SemEval's dataset, the organizers made sure that the tweets in the test set are from different time periods and cover different topics than the ones in the training set. But for ArSAS, a split was not provided and the sets were created through random sampling. For this reason, the correlation between the ArSAS's test and training sets is much higher than that for SemEval's, which, in turn, led to higher results.

Moreover, when experimenting with the mixed model that takes the hand-engineered features into consideration, the results dropped on ArSAS. This might imply that the use of word embeddings alone is enough and that they are more representative than the features.

Table 6.4: Results on ArSAS.

Classifier	***Optimized for SemEval***			***Optimized for the extended training set***		
	AvgRec	***FPN***	***Accuracy***	***AvgRec***	***FPN***	***Accuracy***
SVM	0.53	0.47	0.56	-	-	-
Naive Bayes	0.60	0.61	0.59	-	-	-
NuSVC	0.52	0.49	0.53	-	-	-
LSTM	0.89	0.88	0.90	0.88	0.86	0.88
Bi-LSTM	**0.90**	**0.89**	**0.91**	**0.90**	**0.89**	**0.91**
CNN	0.89	0.88	0.90	0.89	0.88	0.90
CNN-LSTM	**0.90**	**0.89**	**0.91**	0.86	0.86	0.90
LSTM-CNN	**0.90**	**0.89**	**0.91**	0.88	0.88	0.91
Mixed	0.72	0.74	0.78	0.72	0.74	0.78

Chapter 7

Conclusion and Future work

1 Conclusion

In this dissertation, we proposed a wide variety of experiments on deep learning models for Arabic sentiment analysis. These experiments included LSTMs, Bi-LSTMs, CNNs and combinations of them. Moreover, we experimented with a mixed model that combines a feed-forward network fed by hand-engineered features with a Bi-LSTM model fed by word embeddings. Most of the deep learning models achieved high scores on SemEval's dataset, as they achieved an average recall score of 0.60 surpassing the first place holder, who got 0.583. Additionally, the models were tested on ArSAS dataset, where they achieved high average recall scores of around 0.90.

Furthermore, the results show that the use of word embeddings with deep learning models is far superior to the use of hand-engineered features with conventional classification methods. Also, experiments showed that combining the word embeddings and hand-engineered features did not improve the results, but in the contrary, resulted in their decrease.

Finally, in this work we created the largest Arabic social-media-related word embeddings set, which helped in achieving these results compared with other available embeddings.

2 Future Work

In the near future, we are aiming to tackle some details that were not discussed or handled in this project, such as negation, its scope and the variations over the dialects. Also, we are willing to experiment with other variations and improve the results.

Finally, we aim to build an online tool that utilizes the new models, which would help researchers in other NLP related applications.

References

[mic, 2013] (2013). Arabic toolkit service (atks).

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Abdelali et al., 2016] Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In *HLT-NAACL Demos*.

[Abdul-Mageed et al., 2011] Abdul-Mageed, M., Diab, M. T., and Korayem, M. (2011). Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 587–591, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

[Aggarwal and Zhai, 2012] Aggarwal, C. C. and Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

[Ahmed et al., 2013] Ahmed, S., Pasquier, M., and Qadah, G. (2013). Key issues in conducting sentiment analysis on arabic social media text. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*, pages 72–77. IEEE.

[Aizerman et al., 1964] Aizerman, M., Braverman, . M., and Rozonoer, L. (1964). Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25:917–936.

[Al-Ayyoub et al., 2015] Al-Ayyoub, M., Essa, S. B., and Alsmadi, I. (2015). Lexicon-based sentiment analysis of arabic tweets. *International Journal of Social Network Mining*, 2(2):101–114.

[Al Sallab et al., 2015] Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 9–17.

[Al-Smadi et al., 2017] Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., and Gupta, B. (2017). Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels reviews. *Journal of Computational Science*.

[Al-Smadi et al., 2018] Al-Smadi, M., Talafha, B., Al-Ayyoub, M., and Jararweh, Y. (2018). Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *International Journal of Machine Learning and Cybernetics*, pages 1–13.

[Alajmi et al., 2012] Alajmi, A., Saad, E., and Darwish, R. (2012). Toward an arabic stop-words list generation. *International Journal of Computer Applications*, 46(8):8–13.

[Alayba et al., 2017] Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2017). Arabic language sentiment analysis on health services. *CoRR*, abs/1702.03197.

[Alayba et al., 2018] Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2018). A combined cnn and lstm model for arabic sentiment analysis. *arXiv preprint arXiv:1807.02911*.

[Alessia et al., 2015] Alessia, D., Ferri, F., Grifoni, P., and Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3).

[Alhumoud et al., 2015] Alhumoud, S. O., Altuwaijri, M. I., Albuhairi, T. M., and Alohaideb, W. M. (2015). Survey on arabic sentiment analysis in twitter. *International Science Index*, 9(1):364–368.

[Altowayan and Tao, 2016] Altowayan, A. A. and Tao, L. (2016). Word embeddings for arabic sentiment analysis. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3820–3825. IEEE.

[Altrabsheh et al., 2013] Altrabsheh, N., Gaber, M., and Cocea, M. (2013). Sa-e: Sentiment analysis for education. Additional Information: Frontiers of Artificial Intelligence and Applications (FAIA) series, IOS Press.

[Aly and Atiya, 2013] Aly, M. and 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)*, volume 2, pages 494–498.

[Batra and Rao, 2010] Batra, S. and Rao, D. (2010). Entity based sentiment analysis on twitter. *Science*, 9(4):1–12.

[Baziotis et al., 2017] Baziotis, C., Pelekis, N., and Doulkeridis, C. (2017). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

[Chen et al.,] Chen, S. F., Beeferman, D., and Rosenfeld, R. Evaluation metrics for language models. Citeseer.

[Cliche, 2017] Cliche, M. (2017). Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. *arXiv preprint arXiv:1704.06125*.

[Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[Dahou et al., 2016] Dahou, A., Xiong, S., Zhou, J., Haddoud, M. H., and Duan, P. (2016). Word embeddings and convolutional neural network for arabic sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2418–2427.

[Darwish et al., 2014] Darwish, K., Magdy, W., et al. (2014). Arabic information retrieval. *Foundations and Trends® in Information Retrieval*, 7(4):239–342.

[Deriu et al., 2016] Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., Luca, V. D., and Jaggi, M. (2016). Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th international workshop on semantic evaluation*, number EPFL-CONF-229234, pages 1124–1128.

[Di Caro and Grella, 2013] Di Caro, L. and Grella, M. (2013). Sentiment analysis via dependency parsing. *Computer Standards & Interfaces*, 35(5):442–453.

[Duwairi and El-Orfali, 2014] Duwairi, R. and El-Orfali, M. (2014). A study of the effects of preprocessing strategies on sentiment analysis for arabic text. *Journal of Information Science*, 40(4):501–513.

[El-Beltagy, 2016] El-Beltagy, S. R. (2016). Nileulex: A phrase and word level sentiment lexicon for egyptian and modern standard arabic. In *LREC*.

[El-Beltagy and Ali, 2013] El-Beltagy, S. R. and Ali, A. (2013). Open issues in the sentiment analysis of arabic social media: A case study. In *Innovations in information technology (iit), 2013 9th international conference on*, pages 215–220. IEEE.

[El-Beltagy et al., 2017] El-Beltagy, S. R., Kalamawy, M. E., and Soliman, A. B. (2017). Niletmrg at semeval-2017 task 4: Arabic sentiment analysis. *arXiv preprint arXiv:1710.08458*.

[El-Beltagy et al., 2016] El-Beltagy, S. R., Khalil, T., Halaby, A., and Hammad, M. (2016). Combining lexical features and a supervised learning approach for arabic sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 307–319. Springer.

[El-Makky et al., 2014] El-Makky, N., Nagi, K., El-Ebshihy, A., Apady, E., Hafez, O., Mostafa, S., and Ibrahim, S. (2014). Sentiment analysis of colloquial arabic tweets. In *ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conference, Harvard University*, pages 1–9.

[Elmadany and Hamdy Mubarak, 2018] Elmadany, A. A. and Hamdy Mubarak, W. M. (2018). Arsas: An arabic speech-act and sentiment corpus of tweets. In *OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, page 20.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

[Go et al., 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

[Goldberg, 2016] Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

[González et al., 2017] González, J.-A., Pla, F., and Hurtado, L.-F. (2017). Elirf-upv at semeval-2017 task 4: Sentiment analysis using deep learning. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 723–727.

[Habash, 2010] Habash, N. Y. (2010). Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

[Hagen et al., 2015] Hagen, M., Potthast, M., Büchner, M., and Stein, B. (2015). Webis: An ensemble for twitter sentiment detection. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 582–589.

[Hamdi et al., 2016] Hamdi, A., Shaban, K. B., and Zainal, A. (2016). A review on challenging issues in arabic sentiment analysis. *JCS*, 12(9):471–481.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Jabreel and Moreno, 2017] Jabreel, M. and Moreno, A. (2017). Sitaka at semeval-2017 task 4: Sentiment analysis in twitter based on a rich set of features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 694–699.

[Khalil et al., 2015] Khalil, T., Halaby, A., Hammad, M., and El-Beltagy, S. R. (2015). Which configuration works best? an experimental study on supervised arabic twitter sentiment analysis. In *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pages 86–93. IEEE.

[Kohavi and Provost, 1998] Kohavi, R. and Provost, F. (1998). Glossary of terms. *Machine Learning*, 30(2-3):271–274.

[Kolkur et al., 2015] Kolkur, S., Dantal, G., and Mahe, R. (2015). Study of Different Levels for Sentiment Analysis. *International Journal of Current Engineering and Technology*, 55(22):2277–4106.

[Kouloumpis et al., 2011] Kouloumpis, E., Wilson, T., and Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[LeCun et al., 1995] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

[Liu, 2012] Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

[Luhn, 1957] Luhn, H. (1957). 1 a statistical approach to mechanized encoding. *IBM journal*.

[Manning, 2015] Manning, C. D. (2015). Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

[Medhat et al., 2014] Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Miura et al., 2014] Miura, Y., Sakaki, S., Hattori, K., and Ohkuma, T. (2014). Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632.

[Mohammad et al., 2009] Mohammad, S., Dunne, C., and Dorr, B. (2009). Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 599–608. Association for Computational Linguistics.

[Mohammad et al., 2013] Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

[Monroe et al., 2014] Monroe, W., Green, S., and Manning, C. D. (2014). Word segmentation of informal arabic with domain adaptation. In *ACL (2)*, pages 206–211.

[Nabil et al., 2015] Nabil, M., Aly, M., and Atiya, A. (2015). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519.

[Nakov et al.,] Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., and Wilson, T. Semeval-2013 task 2: Sentiment analysis in twitter.

[Nakov et al., 2016] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1–18.

[Pak and Paroubek, 2010] Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

[Pasha et al., 2014] Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Refaee and Rieser, 2014] Refaee, E. and Rieser, V. (2014). An arabic twitter corpus for subjectivity and sentiment analysis. In *LREC*, pages 2268–2273.

[Rosenthal et al., 2017] Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

[Rosenthal et al., 2015] Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., and Stoyanov, V. (2015). Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 451–463.

[Rosenthal et al., 2014] Rosenthal, S., Ritter, A., Nakov, P., and Stoyanov, V. (2014). Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

[Rouvier and Favre, 2016] Rouvier, M. and Favre, B. (2016). Sensei-lif at semeval-2016 task 4: Polarity embedding fusion for robust sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 202–208.

[Soliman et al., 2017] Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

[Soliman et al., 2014] Soliman, T. H., Elmasry, M., Hedar, A., and Doss, M. (2014). Sentiment analysis of arabic slang comments on facebook. *International Journal of Computers & Technology*, 12(5):3470–3478.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

[Tang et al., 2014] Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 208–212.

[Zayyan et al., 2016] Zayyan, A. A., Elmahdy, M., binti Husni, H., and Al Jaam, J. M. (2016). Automatic diacritics restoration for dialectal arabic text. *International Journal of Computing & Information Sciences*, 12(2).