A Comparative Study of Effective Approaches for Arabic Sentiment Analysis

A Comparative Study of Effective Approaches for Arabic Sentiment Analysis

Ibrahim Abu Farha[a], Walid Magdy[a,b]

[a] *School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom*
[b] *The Alan Turing Institute, London, United Kingdom*

Abstract

Sentiment analysis (SA) is a natural language processing (NLP) application that aims to analyse and identify sentiment within a piece of text. Arabic SA started to receive more attention in the last decade with many approaches showing some effectiveness for detecting sentiment on multiple datasets. While there have been some surveys summarising some of the approaches for Arabic SA in literature, most of these approaches are reported on different datasets, which makes it difficult to identify the most effective approaches among those. In addition, those approaches do not cover the recent advances in NLP that use transformers. This paper presents a comprehensive comparative study on the most effective approaches used for Arabic sentiment analysis. We re-implement most of the existing approaches for Arabic SA and test their effectiveness on three of the most popular benchmark datasets for Arabic SA. Further, we examine the use of transformer-based language models for Arabic SA and show their superior performance compared to the existing approaches, where the best model achieves F-score scores of 0.69, 0.76, and 0.92 on the SemEval, ASTD, and ArSAS benchmark datasets. We also apply an extensive analysis of the possible reasons for failures, which show the limitations of the existing annotated Arabic SA datasets, and the challenge of sarcasm that is prominent in Arabic dialects. Finally, we highlight the main gaps in Arabic sentiment analysis research and suggest the most in-need future research directions in this area.

Keywords: Arabic, Sentiment Analysis, Sarcasm

1. Introduction

Sentiment Analysis (SA) is a natural language processing (NLP) application, it can be defined as the process of analysing and identifying the polarity/sentiment expressed in a piece of text, which can be from different sources such as social media posts or product reviews [1].

The emergence of social media platforms as a medium of communication and the growing size of their user-base, led to the generation of massive amounts of rich data that could be used to measure people's opinions and attitudes. For example, many companies rely on products' reviews in order to assess and readjust their marketing and planning strategies. However, analysing this data would require much manual effort and consumes a lot of time. SA automates analysing large amounts of text, which helps in understanding people's attitudes/opinions towards products, events or issues. SA has attracted NLP researchers' interest, who started exploring the applications of SA at different levels based on the target text, which can be a document, a sentence, or an aspect/feature of a product/item [2]. However, most of the work has focused on English, while other languages have lagged behind.

In recent years, the increase of Arabic web content, particularly on social media, and the transformative political developments in the Middle East have attracted more interest to Arabic NLP applications, including SA. Furthermore, recent advances in deep learning (DL) have led to breakthroughs in many NLP

Email addresses: i.abufarha@ed.ac.uk (Ibrahim Abu Farha), wmagdy@inf.ed.ac.uk (Walid Magdy)

applications. However, due to lack of resources for Arabic SA, the application of DL in Arabic SA has been limited.

Multiple studies, such as [3, 4, 5, 6], have surveyed the current state in SA including approaches, challenges, and applications. Additionally, many surveys, such as [7, 8, 9, 10], have focused on Arabic SA. Although these surveys, especially those focusing on Arabic, are reasonably informative concerning the current state-of-the-art, they do not provide a thorough comparison of the relative effectiveness of the different approaches. Most of the surveys focus on providing a high-level view of the state of research in a specific area. Arabic SA surveys are not an exception, they tend to cover SA research from different aspects such as lexicons, corpora and the different ways of constructing them. Also, they tend to categorise the different approaches and methods that could be used to approach SA with coverage of what has been done so far, without deep analysis or one-to-one comparison between different methods. These surveys are useful to give a glimpse about the topic, but, unfortunately, they do not provide deep insights to the intricacies of the field. More importantly, they do not identify which of the surveyed approaches are most effective, especially when most of those approaches for Arabic SA are tested on different datasets with no direct comparison on the same benchmark dataset.

The main objective of this work is to fill the gap of the absence of a comparative empirical study on Arabic SA approaches. Our aim is to have a clear understanding of the most effective approaches for Arabic SA compared directly on the same datasets. In addition, we aim to explore the recently introduced NLP approaches that utilise the new advances in language models, such as bidirectional encoder representations from transformers (BERT). We achieve these objectives through the following contributions:

- We survey state-of-the-art methods for Arabic sentiment analysis.
- We replicate the most effective methods for Arabic SA reported in the literature, and compare their performance on the most popular publicly available Arabic SA datasets.
- We built the largest Arabic word-embeddings trained on 250 million unique tweets, covering multiple Arabic dialects that exist on social media.
- We apply BERT-based models for Arabic SA and compare their performance with all the existing state-of-the-art Arabic SA approaches, showing the superiority of using Arabic specific BERT.
- We conduct an extensive error analysis for the different approaches, which include the reannotation of the existing Arabic SA datasets to assess the subjectivity of the task and the presence of sarcasm.
- We empirically show the challenges that sarcasm imposes on sentiment analysis systems.

In this study, we conduct a rigorous comparison of different machine learning approaches for Arabic SA including Naive Bayes, SVM, CNN, LSTM and a variety of the recently introduced language models. We test the effectiveness of these approaches on the most popular benchmark datasets for Arabic SA, namely SemEval [11], ASTD [12], and ArSAS [13]. Our results show that BiLSTM and CNN-LSTM are the most effective approaches among those reported in the literature, where the BiLSTM achieves F-scores of 0.63, 0,72, 0.89 and the CNN-LSTM achieves 0.63, 0.72, 0.90 on the SemEval, ASTD, and ArSAS datasets respectively. However, those approaches do not compare to transformer-based models, where we show that using BERT trained on Arabic corpus can achieve significantly better performance beating all existing state-of-the-art results, reaching F-score of 0.69, 0.76, 0.92 on the SemEval, ASTD, and ArSAS datasets respectively. Further, we provide an analysis of the performance of the models and the factors that affected their performance in each of the experiments. A phenomenon that is noticed is the high subjectivity of the sentiment and the complexity of choosing a label over another, where when we reannotated portions of the SemEval and ASTD datasets, the mismatch between the new and original labels was 35%. We also labelled the presence of sarcasm among the tweets, and noticed that 16% of the tweets were seen as sarcastic by annotators, which imposes a further challenge on detecting sentiment.

The findings of our study should have several implications on future directions of Arabic SA and Arabic NLP in general. We show that there should be clear guidelines for data annotation for Arabic SA to avoid

the large inconsistency in data labelling. We also grab the attention to the importance of developing methods for Arabic sarcasm detection. Finally, we show the importance of building further Arabic language resources that can have a better impact on Arabic NLP tasks in general.

The rest of the paper is organised as follows: Section 2 provides background about Arabic and a literature survey of the state-of-the-art in Arabic SA. Section 3 provides a detailed description of the various approaches used in Arabic SA. Section 4 explains the experimental setup including the datasets, resources, libraries and parameters used for the different models. Section 5 reports the performance of each of the experimented models on our datasets. Section 6 shows a demo of the deployment of the best model. Section 7 provides a deep analysis of the tested models with an error analysis of their performance. Section 8 discusses the main findings of our study, the gaps in Arabic sentiment analysis we spotted, and suggest the most in-need future research directions in this area. Finally, Section 9 concludes the paper and provides some suggestions for future directions.

2. Background

This section gives background on Arabic language and surveys the literature on Arabic sentiment analysis.

2.1. Arabic Language

Arabic is the most widely spoken Semitic language and is an official language in 28 countries with around 400 million native speakers [14]. Furthermore, Arabic has a particular religious importance, since 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 Dialectal Arabic (DA). 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 [15]. DA is the colloquial language which is spoken in everyday life, this language differs from one country to another, and even varies inside the country itself. DA differs from MSA in many aspects 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 [14].

Arabic imposes many challenges for NLP tasks in general, these include dialectal variation, morphological complexity [16], ambiguity and lack of resources [17].

2.2. Sentiment Analysis

There has been a lot of work on sentiment analysis. However, most of it is focused on English as it is the most widely used language.

One of the initial works on SA is [18], where the authors aimed to analyse 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 [19] they used the set of classifiers used by [18] 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 [20], the authors utilised a set of features with some linear classifiers. In their work, they tried to improve the results by removing the common n-grams, assuming that they are not informative for sentiment classification. In [21], the authors incorporated some lexical and linguistic features in addition to the commonly used n-grams, URLs, etc. They conducted different experiments with different mixtures of features.

Batra et al. [22] focused on entity-based sentiment analysis, where the sentiment is analysed relative to a specific entity. Their model is based on labelled movies' reviews, they used the same model to analyse

tweets. The authors of [23] introduced a context-aware[1] sentiment analysis model, which utilises a set of rules at syntactic level in the dependency parse tree.

Since 2013, SemEval competition [24] included different tasks related to English SA. In 2013, the task was a message level polarity classification. The model by [25] was the winner, they achieved an F_1 score of 69.02. They used SVM with a large variety of hand-engineered features. In SemEval 2014 [26], the authors of [27] won the first place. They used logistic regression and utilised many features including lexical ones, they achieved an average F_1 score of 70.96 in the message polarity classification task. The second place winner used a deep learning model that utilises word embeddings and hand-engineered features. They achieved an average F_1 score of 70.14 [28].

The winner in 2015 [29] 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 [30]. The system created by [31] was the winner of task 4 in SemEval 2016 [32] 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 organisers. Moreover, the second place holder [33] proposed a CNN-based model, which relies on word embeddings and other features; they achieved F_1 score of 63.0.

Baziotis et al. [34] were the winners of SemEval 2017 [11] 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, the authors of [35] achieved a similar result, they used a model that combines CNNs and LSTMs.

The recent advancements in NLP and the emergence of pre-trained language models such as ELMo [36], ULMFiT [37], and BERT [38], led to the advancement of sentiment analysis research through the utilisation of the representational power such models provide. Each of these models was tested on different sentiment datasets and they achieve much higher results than other models.

2.3. Arabic Sentiment Analysis

Similar to the work on other languages, the literature on Arabic sentiment analysis contains many attempts that utilise a wide variety of ML approaches. The most commonly used classifiers for Arabic SA are SVM and Naive Bayes. These classifiers are usually used with hand-engineered features that are based on statistical calculations and lexicons [39].

One of the early works on Arabic SA started with [40], where the authors worked on Standard Arabic and proposed methods to detect subjectivity and sentiment. In [41], the authors proposed a corpus for subjectivity and sentiment analysis. In another work [42], they proposed an SA system for social media. In their work, they experimented with a large set of features. Moreover, In [43, 44], the authors studied different possible ways of handling the morphological richness of Arabic for the task of SA.

In [45], the authors compared the performance of SVM against an RNN-based model in building an aspect-based[2] sentiment analysis system. They tested the model on a dataset for Arabic hotels' reviews, which was part of SemEval 2016 [46]. 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 an accuracy of 87%, for that specific task.

In [47], the authors built a lexicon-based sentiment analysis system that utilises their own lexicon. The model was tested on a manually collected and labelled tweets, they achieved an accuracy of 87%. The authors of [48] targeted social media where they tried to handle the dialects variation through building their own lexicon, namely slang sentimental words and idioms lexicon (SSWIL). They utilised the lexicon and an SVM classifier, which achieved an accuracy of 87%. In [49], the authors collected their own dataset which consists of 2000 tweets. They experimented with different sentiment analysis approaches, their best model was an SVM which achieved an accuracy of 87%.

[1] Context-awareness means considering the context in which a word appears to identify its polarity. For example, in a hotel review, the word "hot" in "hot room" is negative, while it is positive in "hot water".

[2] Aspect-based sentiment analysis works through identifying the aspects/features of a product/service and then finding the sentiment related to each of them.

In [50], the authors proposed a set of Arabic word embeddings to be used for Arabic sentiment analysis. In order to build the embeddings, they used a corpus of around 3.4 billion words. A CNN-based model, which utilises the newly created embeddings, was used to perform sentiment analysis on LABR book reviews dataset [51], Arabic Sentiment Tweets Dataset (ASTD) [12] and other datasets. Another word embeddings set was proposed in [52], where the authors used the embeddings as features to be fed to the classifier. In their experiments, SVM was the best classifier.

In [53], 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 classifier was SVM with an accuracy of 91%. In [54], 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 one of the sentiment analysis tasks [11]. The winner was NileTMRG team [55], where 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 which achieved an average recall of 0.583 and F^{PN} score[3] of 0.61 . The runner up was SiTAKA team [56], they used a combination of features such as bag of words and lexical features. Moreover, they introduced some features that are based on the word embedding vectors such as sum, min, max and standard deviation. The classifier of choice was SVM which achieved an average recall of 0.55 and F^{PN} score of 0.571.

Additionally, the authors of [57] 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, which they introduced in a previous work. The final model achieved an accuracy of 88.1% and 94.3% on the datasets respectively.

Furthermore, Al-Smadi et al. [58] proposed an aspect-based sentiment analysis system, where they created a model based on a character-level BiLSTM combined with conditional random field (CRF) that was responsible for 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 improvement of around 39% with an F-score of roughly 70%.

Recently, the authors of [59] proposed to learn sentiment-specific word embeddings. They used the new embeddings to test and compare their effectiveness against generic embeddings. In their experiments, they tested different models including deep learning. They found that generic embeddings outperforms the sentiment-specific ones. In [60], the authors experimented with different deep learning approaches on a corpus that they manually collected from multiple resources such as Twitter, YouTube, Facebook. They also propose a framework that provides text preprocessing and sentiment classification capabilities.

Table 1 provides a summary of the previous approaches in Arabic SA. There have been many approaches introduced for Arabic SA. However, as could be noticed, each of these approaches is tested on a different dataset. This makes it difficult to identify the best approach among them. In our study, we compare most of the approaches discussed in the literature on standardised benchmark datasets to have a deep comparative analysis to the effectiveness of these approaches on multiple Arabic datasets.

3. Methodology

In this section, we discuss the preprocessing steps, the features, language resources, and the multiple ML approaches inspired from literature that we use for our comparative experimentation.

3.1. Data Preprocessing

Generally, data preprocessing is an initial step that is applied on the input data, which aims to modify the data into a normalised consistent form. It includes various operations and processes that vary based

[3] F^{PN} is the macro average F-score for the positive and negative classes only. Details are in Section 4.3.

Table 1: Summary of previous approaches in Arabic sentiment Analysis. The polarities are positive (POS), negative (NEG), or neutral (NEU).

Article	*Dataset/source*	*Features*	*Approach*	*Polarity*
[40]	Penn Arabic Treeback	Domain, unique words, n-grams, lexicon-based features	SVM	POS/NEG
[42]	Tweets, Wikipedia, Forums	Domain, unique words, n-grams, POS tags, lexicon-based features	SVM	POS/NEG
[43]	Arabic Treebank	Domain, unique words, n-grams, POS tags, lexicon-based features	SVM	POS/NEG
[45]	Arabic hotels' reviews (SemEval-ABSA16)	POS tags, NER feature, morphological features, n-grams, lemmas, stems, word embeddings	SVM, RNN	POS/NEG/NEU
[47]	Tweets	-	Lexicon-based	POS/NEG/NEU
[48]	Comments from Facebook and news websites	N-grams, lexicon-based features	SVM	POS/NEG
[49]	Tweets	N-grams	SVM, NB, KNN, D-tree, unsupervised (lexicon-based)	POS/NEG
[50]	LABR, ASTD, Arabic Gold-Standard Twitter Sentiment Corpus	Word embeddings	CNN	POS/NEG
[52]	LABR, MPQA, ASTD, ArTwitter	Word embeddings	SVM	POS/NEG
[53]	Tweets about health service	N-grams	SVM, NB, Logistic regression	POS/NEG
[54]	Arabic Treebank	N-grams, lexicon-based features, word embeddings	Neural networks, deep auto-encoder, deep belief network	POS/NEG
[55]	SemEval-2017	N-grams, lexical features, lexicon-based features	Complement NB	POS/NEG/NEU
[56]	SemEval-2017	Word embeddings, syntactic features, n-grams, lexicon-based features	SVM	POS/NEG/NEU
[57]	Arabic Health Services dataset, ArTwitter, ASTD	Word embeddings, character embeddings	CNN-LSTM	POS/NEG
[58]	Arabic hotels' reviews (SemEval-ABSA16)	Word embeddings	LSTM	POS/NEG/NEU

on the data form and application. Figure 1 shows the steps that are used in this work, which are mainly based on the work in [55]. These steps are widely used in Arabic SA, but they slightly vary from system to another.

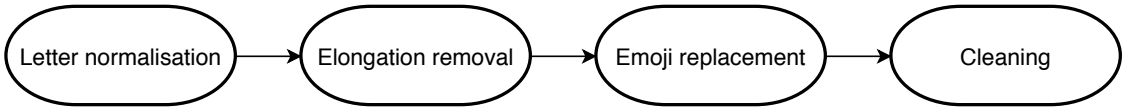

Figure 1: Preprocessing steps.

According to Figure 1, we apply the following preprocessing steps:

- Letter normalisation: 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} [14].
- Elongation removal: sometimes, especially 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. In the implementation, we keep at most two repeated consecutive letters [61].
- 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 collected and annotated by [62]. 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. In the implementation, we used the same terms used in [62] where the positive emojis were replaced by "ايموشنموجب" , the negative emojis were replaced by "ايموشنسالب"

- 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.

3.2. Text Representation

This section details the possible ways to represent the available text in order to be handled and used for SA. We apply two methods for text representation: 1) n-gram word-representation, and 2) word embeddings.

In n-gram representation, the text is represented using statistics from the text itself. There are different ways of doing this such as term-frequency (TF) and inverse document frequency (IDF). In the implementation, we used unigram and bigram word representation as the features and TF-IDF as the feature value. Those were created after applying the previously mentioned preprocessing steps, which reduces the sparsity of these vectors.

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 set [63].

In [63], the authors introduced an efficient way to create word embeddings which are dense representations of words as vectors. This dense representation captures the meaning and the semantics more robustly. Many tests showed that semantics are 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 [63], two architectures were introduced to create the word representation: continuous bag of words (CBOW) and skip-gram. Both 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. In skip-gram, which is used in this work, the representation of a word is learnt through predicting words within a certain range before and after the current word, i.e. context. Other methods for creating dense word representation include GloVe [64] and fastText [65]. Word embeddings have been utilised in multiple Arabic NLP tasks including SA [55, 56, 59, 66, 67]. The largest dataset used for building the Arabic word embeddings (AraVec) was the one by [68], which was built using around 67M tweets.

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 250M unique Arabic tweets was utilised, which is larger than any set used to create Twitter-related word embeddings. The tweets were collected over different time periods between 2013 and 2016 to ensure topic diversity. The same preprocessing steps discussed earlier were applied for the tweets before building the embeddings.

Table 2 shows the statistics of the corpus used in the embedding creation. We generated embeddings vectors of length 300.

Table 2: Twitter corpus statistics.

#tweets	249,941,286
#words	3,057,189,052
#unique words	8,916,818

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. Our embeddings are available freely for public for research purposes[4]. The embeddings were published along with the work in [67].

[4] `http://mazajak.inf.ed.ac.uk:8000/#embedding-page`

3.3. Hand-engineered Features

In addition to the word representation of the text, there are many different features that can be extracted and used for sentiment classification, which have been explored in the literature [62, 55]. In this work, the following features are extracted and examined for Arabic SA:

- 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 one of the following values {0, 1, 2}. This feature represents the length of the tweet, where it takes the value {0} if the tweet has less than 60 characters, {1} if it has between 60 and 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. Segments are identified using 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: this feature represents the count of positive emojis in the tweet, which were provided by [62, 55].
- Number of negative emojis: this feature represents the count of negative emojis in the tweet [62, 55].
- Number of positive terms: the count of the positive terms in the tweet, where the terms were taken from [69]. This score is calculated through counting the number of positive terms. However, a weighing factor was applied to give more weight for the compound terms as shown in the equation:

$$numOfPos = \sum_{i=0}^{n} i + \sum_{j=0}^{c} j \times \alpha, \tag{1}$$

  where n is the number of single-word positive terms, c is the number of positive compound terms and α is a weighting factor such that $\alpha > 1$. In the implementation we set α to 1.5 similar to [62].
- Number of negative terms: a real number that represents the score of negative terms [69], calculated similar to $numOfPos$ in equation 1.
- 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 number that represents the percentage of the positive terms with respect to the total number of terms in the tweet.
- NegPercentage: a real number that represents the percentage of the negative terms with respect to the total number of terms in the tweet.

The above features along with the text representation were used to train multiple classifiers, which are discussed in the following section.

3.4. Machine Learning Approaches

This section goes over the different algorithms and methods that we used for the sentiment classification. These include classical classifiers and deep learning models that have been used in literature in addition to new architectures proposed by us.

3.4.1. Classical Machine Learning Models

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 this work, SVM and Naive Bayes were used . The reason for this choice is that both algorithms are used extensively in the literature of Arabic SA. For SVM, we examined both linear and non-linear kernels.

3.4.2. LSTM Model

Language is context-dependent and word order is extremely important. As different word order might lead to a different meaning, word order has to be taken into consideration. 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 utilisation of long short-term memory (LSTM) networks can overcome this issue. 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 after the last word is fed to a softmax output layer that produces the output probability of each of the classes, Figure 2 shows the model used.

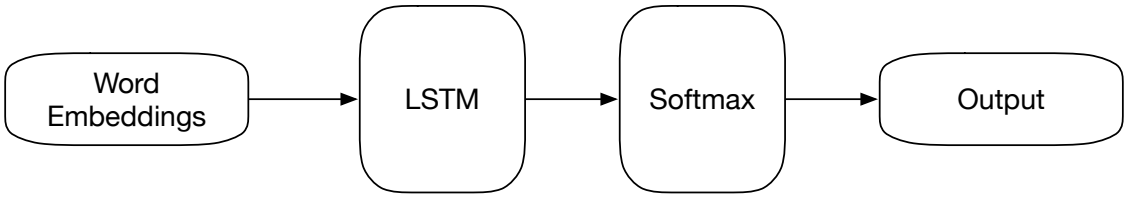

Figure 2: LSTM model architecture.

3.4.3. BiLSTM Model

LSTMs tend to capture the dependencies in one direction, and sometimes they might lose important information. This has been tackled using bidirectional LSTMs (BiLSTM). BiLSTMs 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 3.

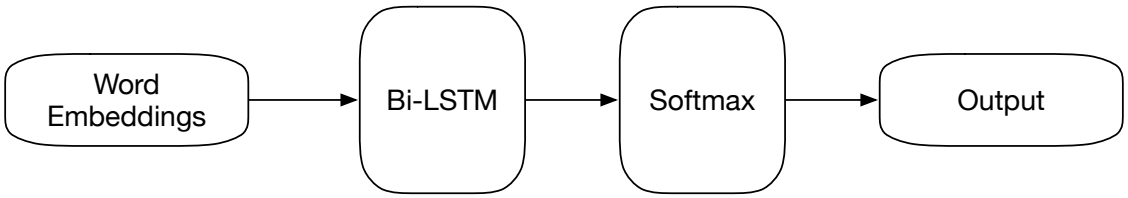

Figure 3: BiLSTM model architecture.

Additionally, the architecture shown in Figure 4 was used. It is based on the work in [66], where they achieved good results on ASTD dataset [12]. The model is similar to the one above but there is a dropout and dense layers before the softmax layer.

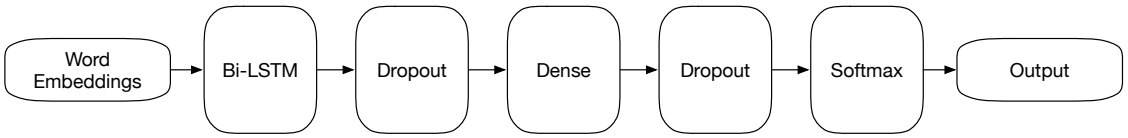

Figure 4: Modified BiLSTM model architecture.

3.4.4. CNN Model

Convolutional Neural Networks (CNNs) have demonstrated high performance at capturing correlations and patterns in data, which might be useful, because the text is represented as a concatenation of words' vectors, and related words would have correlated vectors. These correlations might work as features to distinguish between different classes.

Figure 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 are 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 to learn from the newly extracted features and at the end there is a softmax layer.

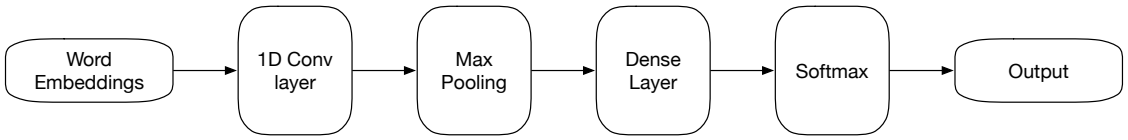

Figure 5: CNN model architecture.

Moreover, we used the architecture in Figure 6, which is based on the work in [35, 66]. The model consists of three parallel CNN layers where each of them has a different filter size, and each of them is followed by a max-pooling layer. After the max-pooling, the outputs are concatenated and fed into a dense layer. A dropout layer is used to provide a regularizer effect. Finally, the output is taken from the softmax layer.

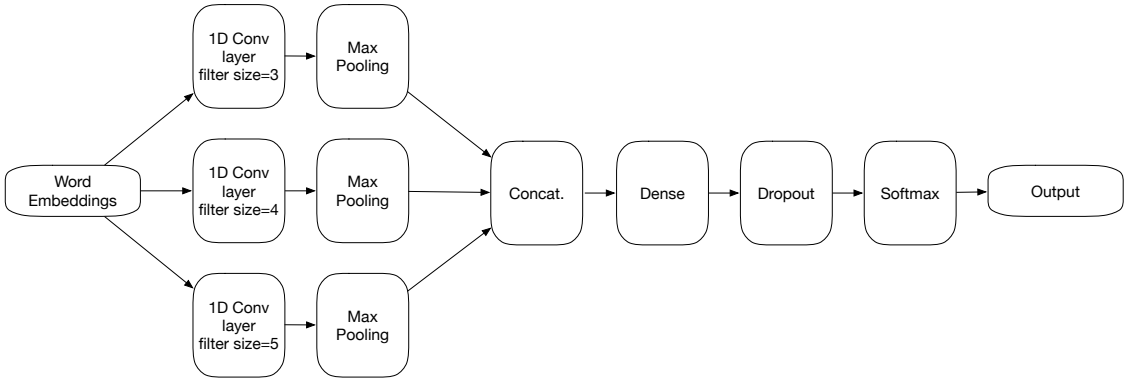

Figure 6: Modified CNN model architecture.

3.4.5. CNN/LSTM mixed Models

A combination between CNN and LSTM has also been studied for Arabic SA [57]. The model starts with a CNN network followed by an LSTM layer, as shown in Figure 7. 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.

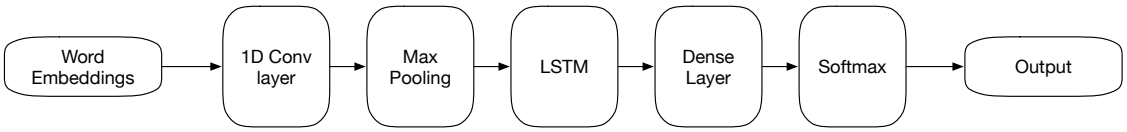

Figure 7: CNN-LSTM model architecture.

In addition to the previous model, we apply a similar combination to the previous one, but here the LSTM comes before the CNN as shown in Figure 8. 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.

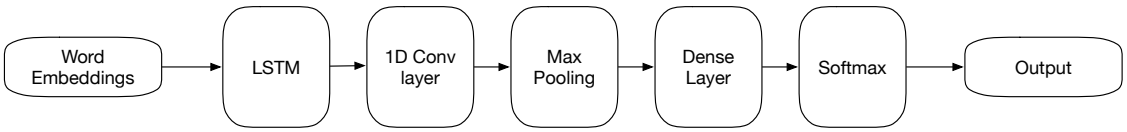

Figure 8: LSTM-CNN Model.

3.4.6. Ensemble Models

In this part, we examine the usage of ensemble models. The advantage of ensemble models is that they utilise multiple smaller models, where each of them can provide some improvement to the overall performance. We experiment with two ensemble architectures. The first is based on the work in [66]. It uses both of the models in Figures 6 and 4, it takes an average of their outputs to achieve the final output. This idea will assure a combined training of the models, where signals might propagate from one to another which would help utilising the advantages of both models, we will refer to this model as Ensemble (CNN/LSTM).

The other ensemble model combines the architectures of the word-embedding-based deep learning model with the models based on hand-engineered features. Figure 9 shows the detailed model, this model aims to combine the features and dependencies learned by the BiLSTM 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, we refer to this model as Ensemble(BiLSTM/features).

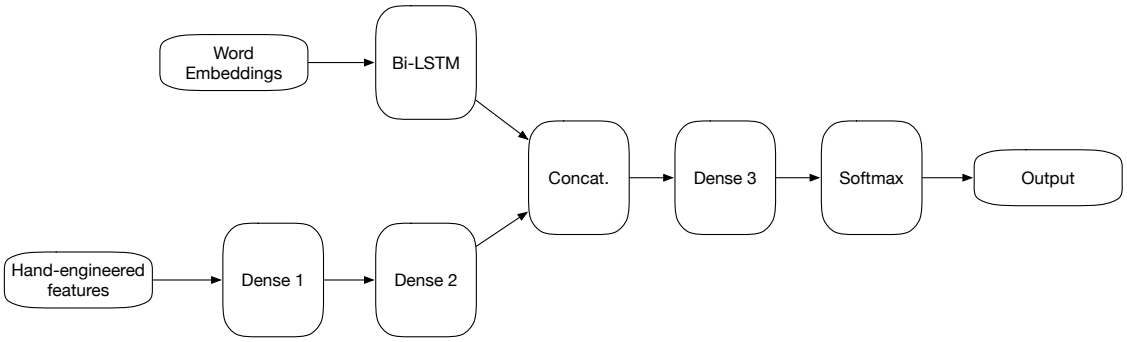

Figure 9: Ensemble of BiLSTM and feed-forward models.

3.4.7. Language-Models Based Models

In this part, we utilise the available Arabic language models to perform the task of sentiment analysis.

The first model is hULMonA [70], an Arabic language model that is based on Universal Language Model Fine-Tuning (ULMFiT) [37]. This architecture relies on using a pre-trained language model and fine tune it for a specific task, i.e. transfer learning. Since hULMonA is based on ULMFiT, it uses the same architecture of three layers of AWD-LSTM [71]. The new model was trained on 600K Wikipedia articles, and they used MADAMIRA [72] for preprocessing and tokenization.

The introduction of Bidirectional Encoder Representation from Transformers (BERT) [38], led to a revolution in the NLP world. The proposed architecture is also reliant on transfer learning and fine-tuning. In the experiments, we use two variants of BERT-based models. The first is the multilingual BERT which was trained on 104 languages and relies on 110K shared WordPiece vocabulary. BERT consists of 12 layers with 768 hidden units in each of them, and 12 attention heads. The multilingual BERT was trained on the entire Wikipedia dump for each language.

The other model is AraBERT [73], which was built using the same architecture as BERT-base [38]. AraBERT was trained using a combination of different Arabic news corpora. The authors utilised Farasa [74] for the preprocessing and segmentation, then they trained a SentencePiece tokenizer [75] on the segmented text with a vocabulary of 60K subword tokens.

In the experiments, we fine-tune these models through adding a fully connected layer and a softmax layer after the pre-trained model. Then the model is trained for a small number of epochs to adjust the weights for the specific task.

4. Experimental Setup

4.1. Arabic Sentiment Benchmark Datasets

This section provides information about the datasets which are used in our experiments.

4.1.1. SemEval 2017 Task 4-A Dataset

The SemEval Arabic SA dataset is one of the most popular benchmark datasets for this task. The data was provided as part of SemEval-2017, Task 4-A, which is about predicting the sentiment of tweets and classifying them to positive, negative or neutral [11]. The data is provided as two sets, a training set of 3,355 Arabic tweets and the test set that contains 6,100 tweets. Moreover, the organisers provided a validation set of 671 tweets, Table 3 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 by 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. The annotation was performed using CrowdFlower[5] crowdsourcing platform.

Table 3: SemEval 2017 Task 4-A dataset statistics.

Set	***Positive***	***Negative***	***Neutral***	***Total***
Training	743	1,142	1,470	3,355
Validation	222	128	321	671
Testing	1,514	2,222	2,364	6,100
Total	2,479	3,492	4,155	10,126

4.1.2. ArSAS Dataset

ArSAS is the most recently released manually annotated dataset for Arabic speech-act and sentiment analysis. Currently, it is considered the largest human-annotated dataset for Arabic SA, as it contains around 21K tweets. Additionally, the tweets cover many different topics and most of them are in dialectal Arabic. The data was manually annotated using CrowdFlower[5]. 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 [13]. The data was collected from 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 controversial topics that were of importance at that time. The topics include some long-standing topics, events that were happening at that time, and some entity-related tweets such as celebrities.

[5]Currently known as Appen

In the implementation, we ignore the mixed class, since it has the smallest number of samples (see Figure 10). Additionally, ArSAS has a confidence value for each label, which was used to eliminate low-confidence labels; and only the tweets with confidence level over 50% were kept. After this step, we end up with 18,819 tweets in the ArSAS dataset labelled with three sentiment labels. Since no specific split was provided, an 80/20 split was applied randomly to create the train and tests sets respectively[6]. Table 4 shows the dataset statistics and the details of the splits, while Figure 10 shows the percentages of the classes.

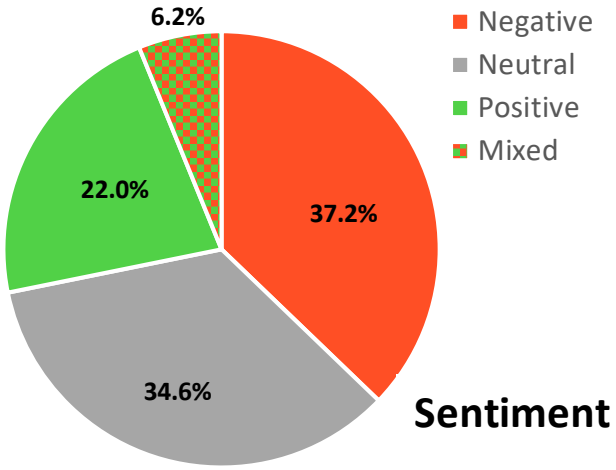

Figure 10: ArSAS tweets' sentiment distribution [13].

Table 4: ArSAS dataset statistics. The first row is the statistics of the original dataset. The other rows are the statistics of the splits used in the experiments after filtration.

Set	***Positive***	***Negative***	***Neutral***	***Mixed***	***Total***
ArSAS	4,643	7,840	7,279	1,302	21,064
Training	2,916	4,816	4,487	-	12,219
Validation	680	1,204	1,116	-	3,000
Testing	724	1,433	1,443	-	3,600

4.1.3. ASTD Dataset

The Arabic Sentiment Tweets Dataset (ASTD) dataset contains 10,006 tweets, mainly in Egyptian dialect [12]. It is distributed over 4 classes as shown in Table 5. The tweets were collected over the period between 2013 and 2015, based on the most trending topics at that time. The authors did not provide any specific splitting. However, they provided a code sample that can generate either balanced or unbalanced splits [12]. Since we are focusing on sentiment analysis in our experiments, we are mainly interested in the subjective part of the dataset, thus the objective class was eliminated leaving us with a set of 3,315 tweets. The resultant set consists of only subjective tweets that belong to one of the class: positive, negative or neutral as shown in Table 5. A balanced 80/20 split was applied into the resultant set to create the training and testing sets.

[6] Tweets with 100% confidence were used to prepare the test set (all annotators agreed on the same label).

Table 5: ASTD dataset statistics. The first row is the statistics of the original dataset. The other rows are the statistics of the splits used in the experiments.

Set	***Positive***	***Negative***	***Neutral***	***Objective***	***Total***
ASTD	799	1,684	832	6,691	10,006
Training	489	1,086	546	-	2,121
Validation	141	263	127	-	531
Testing	169	335	159	-	663

4.2. Arabic Sentiment Lexicon

In some of our experiments, we also used some Arabic sentiment lexicon. We used the NileULex lexicon [69], which contains around 6000 sentiment terms that are taken from the Egyptian dialect and MSA. The original lexicon was released in 2013, and in the following two years more terms were added and many were revised. At the time of writing this paper, the lexicon contains a large variety of sentiment terms, where 55% of them are MSA and the other 45% are from the Egyptian dialect, Table 6 shows the lexicon details.

Table 6: NileULex statistics.

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

4.3. Evaluation Metrics

For evaluation, we adopt the same metrics used at the SemEval-2017 task [11]. The organisers adopted the average recall (*AvgRec*) as a primary metric for sentiment classification, due to its robustness against the imbalances of the classes, the following equation is used:

$$AvgRec = \frac{1}{3}(R^P + R^N + R^U), \tag{2}$$

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

$$F^{PN} = \frac{1}{2}(F_1^P + F_1^N), \tag{3}$$

Where F_1^P, F_1^N are the F_1 with respect to the positive and negative classes respectively, while the neutral class is ignored.

4.4. Environment Setup and Parameters Optimisation

In our implementation, Python was the language of choice because it has a large variety of supporting APIs that would make the workflow easier. For the machine learning part, we used the *scikit-learn* library [76], which provides a variety of machine learning algorithms. For the deep learning experiments, *Keras* [77] was used on top of a *Tensorflow* [78] back-end[7].

All hyper-parameters were selected through grid-search to maximise the performance on SemEval's dataset. The hyper-parameters for the deep learning experiments are shown in Table 7. For all experiments, we used Rectified Linear Unit (*ReLU*) and *Adam* [79] optimiser with a learning rate of 0.0001. The best value for the SVM regularization parameter C is 1000.

[7]The experiments were conducted on a machine with an 8-core CPU and 64GB RAM. The transfer learning experiments were conducted using Google Colab, which provides Nvidia P100 GPU.

Table 7: Hyper-parameters used for deep learning models.

Model	***#LSTM cell***	***recurrent dropout***	***output dropout***	***#filters***	***filter size***	***pooling size***	***#hidden units***
LSTM	128	0.2	0.2	-	-	-	-
BiLSTM	64	0.2	0.2	-	-	-	-
CNN	-	-	-	300	3	2	256
CNN-LSTM	128	0.2	0.2	300	3	2	128
LSTM-CNN	100	0.2	0.2	32	3	2	-
Ensemble (BiLSTM/features)	128	0.2	0.2	-	-	-	100
Modified CNN	-	-	0.5	200	[3,4,5]	-	30
Modified BiLSTM	200	-	0.5	-	-	-	30
Ensemble (CNN/LSTM)	200	-	0.5	200	[3,4,5]	-	30

Regarding the experiments with BERT, we relied on the implementation provided by HuggingFace's Transformers library [80]. We used the provided *BertForSequenceClassification* implementation along with BertAdam optimiser. We trained the models for 4 epochs with a learning rate of $1e{-}5$. The maximum sequence length was set to the maximum length seen in the training set. For AraBERT experiments, we used Farasa [74] for the preprocessing and segmentation, similar to the original paper.

For the experiment with ULMFiT, we relied on *fast.ai*[8] library. The fine-tuning was done in a similar way to BERT, however, here we used gradual unfreezing of the layers and tuned each of them using learning rates that range from $2e{-}5$ to $1e{-}2$. Finally, the whole model was trained for 3 epochs. Also, it is worth mentioning that we relied on MADAMIRA [72] for the preprocessing and tokenization.

5. Results

In this section, we provide the results of the experiments. In general, a total of 15 different methods were tested, these methods cover most of the sentiment analysis approaches available in the literature.

Table 8 shows the experiments' results on all the three datasets, where the $AvgRec$, F^{PN} and the accuracy are reported. The first three rows show the state-of-the-art top three teams in SemEval, while the other three are the results on ASTD dataset from [66]. The highest score achieved previously on SemEval's dataset was $AvgRec$ of 0.58, while the highest on ASTD is an $AvgRec$ of 0.613.

As can be seen in Table 8, conventional classifiers perform poorly compared to deep learning models. The performance of these classifiers varies depending on the dataset. The non-linear SVM (NuSVC) has the best performance on all three datasets. Using deep learning models has a significant effect on the performance of Arabic SA. All of these models perform better than the conventional ones, which indicates that the use of word-embeddings combined with deep learning models is better suited to handle the Arabic sentiment analysis problem. These results show the power of word-embeddings, where the meanings of the words are encoded within the representation. Consequently, deep learning models have the capability to correlate and learn meaning-dependent relations, which is more logical than using sparse n-gram representations, which does not include any information about the meaning of a word. Additionally, we can see that the models which utilise LSTMs are performing better than others. This is because LSTMs are capable of capturing long term relations and dependencies over the input text.

Regarding the transfer learning experiments, BERT and hULMonA, these models were not as effective as the other deep learning models. This could be attributed to the limited Arabic vocabulary in BERT. Regarding hULMonA, the fact that it is trained on standard Arabic rather than dialectal is probably the reason for its weak performance. Furthermore, we experimented on BERT with and without preprocessing, the results without preprocessing were relatively higher and thus we report them.

AraBERT shows a significant improvement over the other models. This improvement can be attributed to the fact that AraBERT was trained using a SentencePiece toknizer, which was trained on the text

[8] https://www.fast.ai

segmented by Farasa. This way the learned vocabulary would be more representative of Arabic morphology, which would enable the model to handle and learn a better contextual representation of Arabic morphemes and subword tokens.

Table 8: Results on all datasets along with the top results reported in the literature. The abbreviations in the features column are hand-engineered (HE), static embeddings (SE), and contextualised embeddings (CE).

Approach	*Features*	*SemEval*			*ArSAS*			*ASTD*		
		AvgRec	$\mathbf{F^{PN}}$	*Acc*	*AvgRec*	$\mathbf{F^{PN}}$	*Acc*	*AvgRec*	$\mathbf{F^{PN}}$	*Acc*
[55]	HE	0.58	0.61	0.58	-	-	-	-	-	-
[56]	HE	0.55	0.57	0.56	-	-	-	-	-	-
[81]	SE	0.48	0.47	0.51	-	-	-	-	-	-
BiLSTM [66]*	SE	-	-	-	-	-	-	0.593	0.72	0.648
CNN [66]*	SE	-	-	-	-	-	-	0.61	0.71	0.643
Ensemble(CNN/LSTM) [66]*	SE	-	-	-	-	-	-	0.613	0.71	0.651
SVM	HE	0.47	0.43	0.49	0.56	0.54	0.61	0.33	0.34	0.50
Naive Bayes	HE	0.45	0.37	0.48	0.49	0.34	0.58	0.33	0.34	0.50
NuSVC	HE	0.47	0.44	0.47	0.60	0.60	0.58	0.47	0.63	0.59
LSTM	SE	0.60	0.61	0.63	0.89	0.89	0.91	0.57	0.73	0.65
BiLSTM	SE	0.61	0.63	0.63	0.89	0.89	0.91	0.60	0.72	0.66
Modified BiLSTM	SE	0.60	0.61	0.63	0.89	0.88	0.90	0.60	0.73	0.67
CNN	SE	0.57	0.57	0.59	0.90	0.89	0.91	0.56	0.70	0.64
Modified CNN	SE	0.57	0.57	0.59	0.89	0.88	0.90	0.56	0.69	0.63
CNN-LSTM	SE	0.61	0.63	0.62	0.90	0.90	0.92	0.62	0.72	0.66
LSTM-CNN	SE	0.60	0.62	0.63	0.90	0.89	0.91	0.59	0.73	0.66
Ensemble(CNN/LSTM)	SE	0.56	0.55	0.59	0.88	0.87	0.90	0.56	0.73	0.65
Ensemble(BiLSTM/features)	SE + HE	0.50	0.42	0.49	0.90	0.89	0.91	0.62	0.73	0.66
BERT	CE	0.51	0.48	0.54	0.88	0.87	0.89	0.55	0.67	0.60
hULMonA	CE	0.41	0.34	0.45	0.88	0.87	0.90	0.47	0.60	0.57
AraBERT	CE	**0.67**	**0.69**	**0.68**	**0.93**	**0.92**	**0.93**	**0.66**	**0.76**	**0.69**

The best performing models over all the three datasets are AraBERT, BiLSTM and CNN-LSTM models. In order to test how these models generalise, we trained them on the combined training sets from the three datasets and tested the trained models on each of the testing sets. Results of this experiment are reported in Table 9.

Table 9: Results of the best performing models.

Approach	*SemEval*			*ArSAS*			*ASTD*		
	AvgRec	$\mathbf{F^{PN}}$	*Acc*	*AvgRec*	$\mathbf{F^{PN}}$	*Acc*	*AvgRec*	$\mathbf{F^{PN}}$	*Acc*
BiLSTM	0.61	0.62	0.63	0.89	0.89	0.91	0.57	0.73	0.64
CNN-LSTM	0.62	0.63	0.63	0.90	0.90	0.91	0.56	0.73	0.63
AraBERT	**0.63**	**0.65**	**0.65**	**0.93**	**0.92**	**0.93**	**0.67**	**0.78**	**0.70**

The results in Table 9 demonstrate an overall excellent performance across the three test sets. It is noticeable that the performance of these models is close to their performance when they were trained for a specific dataset, which indicates that they can generalise.

6. Mazajak: An Online Arabic Sentiment Analyser

Based on the experiments in the previous section, the CNN-LSTM model was demonstrated to be among the most effective models for Arabic SA. This model was used to create the first online Arabic sentiment

*We followed the same steps as [66], where they ignored the objective class. We also used the same split ratios, which were generated using the code provided by [12]. However, there is no guarantee that it generated the exact test split of [66].

analyser, Mazajak [67], which is trained on the combination of the aforementioned datasets, including both the training and test sets of these datasets. Mazajak[9] provides different functionalities for Arabic sentiment analysis, including submitting text in a box to get its sentiment, and submitting a Twitter account to analyse the sentiment of the latest 3,200 Arabic tweets in the timeline of the accounts. The latter feature is quite helpful in cases such as studying the reactions of public figures to events that take place in a time frame. Figure 11 shows an example of the output of the timeline analysis feature provided in Mazajak. Majazak also provides two modes for text bulk processing, including the batch mode and the online API. These modes are targeted for the research community where sentiment information can be useful as an input or a tool of study. Mazajak API is considered the first free open-source Arabic sentiment analysis API.

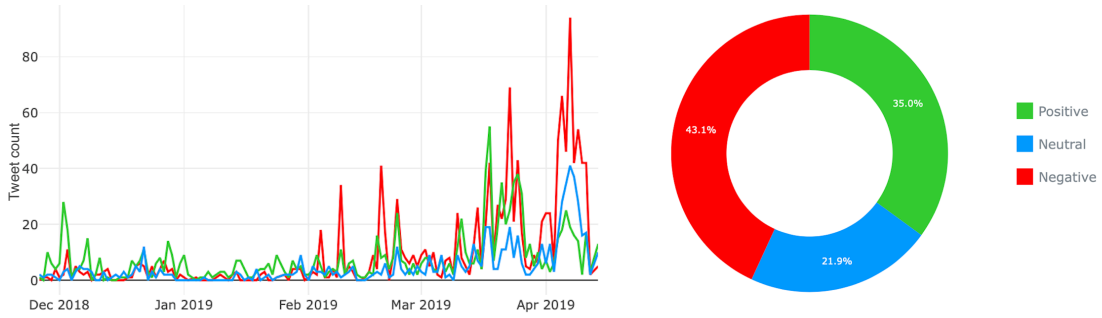

Figure 11: Sample outputs of the timeline feature in Mazajak. (Left) A time-series showing the tweet-count over time for each of the sentiment classes. (Right) The sentiment percentages of the analysed tweets.

7. Analysis

From our extensive comparison of multiple machine learning models for the task of Arabic SA, it can be noticed that the performance on the SemEval and ASTD datasets was considerably lower than the ArSAS dataset. The high performance on ArSAS may stem from the nature of the topics covered in the dataset, which are controversial [13], and thus the sentiment could be clear in the text. In order to understand the reason for the low performance on the SemEval and ASTD datasets, we studied the nature of the tweets in these datasets and the cases where the classifiers fail.

Through our investigation, we found that there is some inconsistency in labelling tweets reporting news. We found that some annotators would sometimes label a news tweet as neutral, considering the objectivity of the source (the news agency). However, sometimes they tended to give the label based on the content itself. We believe that this confusion affected the classifiers and their performance.

7.1. Subjectivity of the Sentiment Labelling Task

To better understand the nature of the annotation, we decided to redo it from scratch. We started a new annotation process, where we annotated portions of SemEval and ASTD dataset. The total was 10,547 tweets, the majority (8,075 tweets) were taken from SemEval's dataset. To preserve consistency, we used the same guidelines for sentiment labelling used to annotate SemEval's dataset, where the track organisers were kind enough to share the guidelines with us. The annotation was conducted using Appen crowd-sourcing platform[10]. In addition to the sentiment label, we asked participants to provide labels for the dialect of the tweet and the presence of sarcasm.

Each tweet was annotated by at least three different annotators. Only annotators who have Arabic language in their profiles and located in an Arab country were allowed to participate. The overall agreement

[9]available at: "`http://Mazajak.inf.ed.ac.uk:8000`"
[10]https://www.appen.com/

among annotators for the tweets was 80.7%, 86.7%, and 89.3% for the sentiment, dialect, and sarcasm labels respectively.

Figure 12 shows the agreement between the original and the new sentiment labels of the dataset for each of the sentiment classes, where the labels above the charts are the original labels. As shown, a huge shift has occurred to the final sentiment labels of the tweets reaching more than 50% for the positive sentiment class. It was astonishing to get this significant mismatch in the sentiment labels assigned to the tweets compared to the original ones. This demonstrates the high subjectivity of the task and large variations among annotators, despite being provided with the same labelling guidelines.

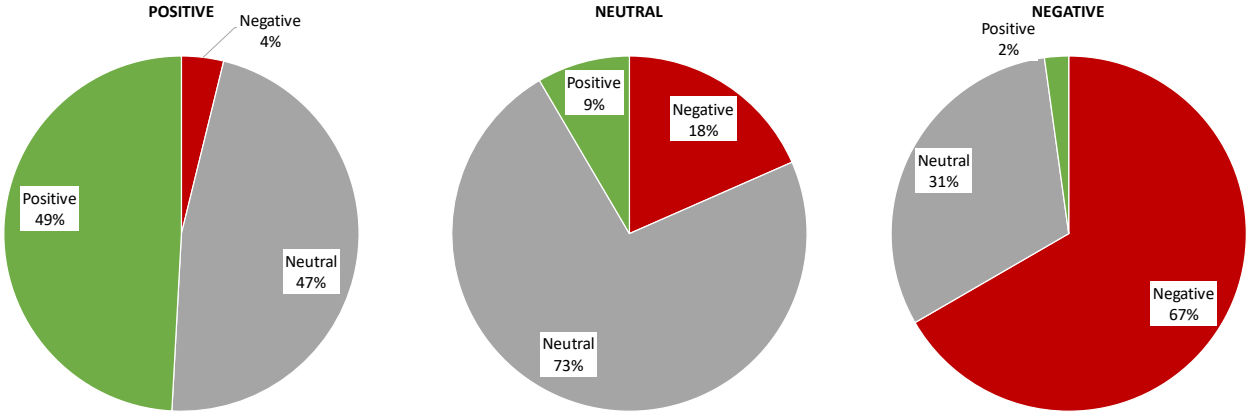

Figure 12: Sentiment labels change after the new annotation. The labels above the charts are the original labels.

ID	*Tweet*	*Original sentiment*	*New sentiment*
1	أبل تكسب معركة قضائية ضد سامسونغ (Apple wins a lawsuit against Samsung)	Positive	Neutral
2	ضبط ٨٩ طن سكر فى مخزن بتملكه تاجر بقصدالاحتكار فى العاشر من رمضان (In the tenth of Ramadan, 89 tons of sugar were seized in a store owned by a merchant for the purpose of monopolising)	Positive	Negative
3	انا اللي غلطان يا سيدي،كنت عايز اقف جنبك (It is my mistake, I wanted to support you)	Neutral	Negative
4	لا ده نتكاتف فيه رغم اى خلافات أو أخطاء حدثت (No, we will support each other in this, regardless of any disagreement or mistakes that happened)	Neutral	Positive
5	سمعة أبل على المحك.. مشكلة حقيقية في آيفون ٧ (Apple's reputation is on the line ... A real problem in iPhone 7)	Negative	Neutral
6	طقس كاذب يقولو ثلوج ويطلع حر (deceitful weather, they say it will snow and it is warm)	Negative	Positive

Table 10: Examples of some tweet that have its labels changed.

Table 10 shows some examples of these cases, where the sentiment label has changed. From the examples, it is noticeable that the shift in labels might stem from the different perspectives that the annotators might consider. For example, in the case of a news content, some of the annotators might provide the label as neutral since it is an objective piece of news, while others consider the sentiment of the news itself. Some of the annotators, especially when the content is political or related to rivalling parties, they provide the label based on their preference. The first example demonstrates that, where the original annotator considered the information from Apple's perspective, while the new annotator considered it as news coming from a neutral agency, thus labelled it as neutral.

7.2. The Challenge of Sarcasm

While reannotating the tweets for sentiment, we added a couple of questions for annotators on each tweet; one about the dialect of the tweet, where we provided six options {MSA, Egypt, Levant, Gulf, Maghreb} [14];

and the other about if the tweet is sarcastic or not. Our main purpose behind the second additional question is to measure how sarcasm might be disruptive to sentiment analysis systems.

Our annotation task showed that 1,682 tweets (16% of the tweets) have been labelled as sarcastic. The majority (47.5%) of sarcastic tweets are in Egyptian dialect (799 tweets), followed by MSA (631 tweets), Gulf (122 tweets), Levant (118 tweets), then Maghrebi (12 tweets). Figure 13 shows the percentage of sarcasm with each dialect. 34% of the Egyptian tweets in our collection are sarcastic. The lowest percentage of sarcasm are within the MSA, where less than 10% of the tweets are sarcastic, which is expected, since MSA is mainly used for official communication [15]. The percentage of sarcasm in Maghrebi dialect is highest (38%), however, this can be an outlier since the collection has only 32 Maghrebi tweets.

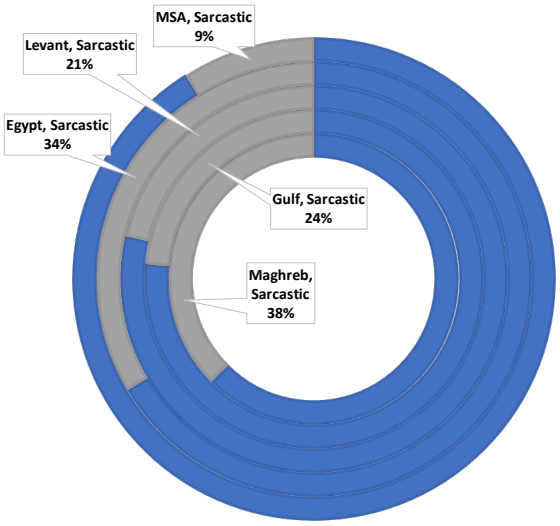

Figure 13: Sarcasm ratio over the dialects.

ID	*Tweet*	*Sentiment*	*Dialect*
1	لن تصبحي جستن بيبر (You won't become Justin Bieber)	Neutral	MSA
2	بوكيمون ايه اللي اقوم اضور عليه .. ده انا بكسل اقوم اطفي نور الاوضه ..!! (What Pokemon that I would be looking for! I am too lazy to turn off my room's lighting)	Negative	Egypt
3	بالصيفيات الحلوه محد يقرر ينزلني على لبنان لما وصلت درجه الحراره تحت الصفر امي تقول نفكر نروح لا شكرا (When it is summer, no one suggests going to Lebanon. Now, when it is below zero, my mother considers going there. No, thanks)	Negative	Levant
4	الناس المؤمنين بالسحر كان لازم نوضحلهم ان هاري بوتر مو فلم وثائقي (We should have explained for those who believe in magic that Harry Potter is not a documentary)	Negative	Gulf

Table 11: Examples of some sarcastic tweets from different dialects.

Table 11 provides examples of sarcastic tweets from different dialects. Those examples demonstrate different attributes of a sarcastic expression. In general, most of the sarcastic tweets rely on referencing some known figures or world knowledge. Also, these expressions are highly contextual, and they could be interpreted differently based on the context [82].

We also analysed the distribution of sentiment within the 1,682 sarcastic tweets. The analysis shows that sarcasm is mostly used to convey negative sentiment, where 88% of the sarcastic tweets have negative sentiment, 9% neutral and only 3% positive. The presence of neutral and positive sentiment within sarcastic

context could be due to the inclusion of other forms of figurative language within the sarcasm definition. An example of that is humble bragging such as "I am exhausted from my two-week vacation to Hawaii"

Finally, to measure the effect of sarcasm on sentiment analysis, we used Mazajak [67] on both the sarcastic and non-sarcastic tweets of the combined data of SemEval and ASTD. Since Mazajak was trained on portions of these datasets, we used the original sentiment labels as a reference. Table 12 shows the performance gap between sarcastic and non-sarcastic tweets, where the F-score performance on non-sarcastic tweets is 0.63, while it is only 0.5 on the sarcastic tweets. These results demonstrate how sarcasm can be highly disruptive for sentiment analysis systems, which is another limitation of the existing work on Arabic SA that needs to be tackled in future research work.

Table 12: Mazajak's performance on Sarcastic vs. non-sarcastic tweets.

Group	***AvgRec***	$\mathbf{F^{PN}}$
Sarcastic	0.46	0.50
Non-Sarcastic	0.61	0.63

8. Discussion

The aim of our paper was to put all existing algorithms in literature for Arabic sentiment analysis into a direct comparison to get a clear analysis of the best performing algorithms.

Our experiments with different sentiment analysis models and approaches show the effectiveness of deep learning models for this task compared to classical machine learning algorithms, such as SVMs. The advantage of such models is that they are better at utilising and capturing semantic features in the text. Since these models rely on word-embeddings, they also incorporate in some way the meanings of the words which are represented through the word vectors in the embedding space.

Moreover, the results show that the best models are the ones that can capture the context such as LSTMs and AraBERT. This matches the intuitive thinking about language and sentences' structure, i.e. different ordering of words leads to different sentences and meanings. In addition, we found that the use of pre-trained language models, when trained on enough data, are superior to other models. This is demonstrated by the gap in performance between AraBERT and the multilingual BERT.

Our analysis of the most popular benchmark dataset on Arabic SA (SemEval) raises questions about the current annotation schemes to create datasets of subjective content. The reannotation of the available datasets shows an extreme change in the labels assigned to tweets. This indicates that annotators' subjectivity and biases affect their choice of the labels. The provided analysis and examples show that annotation can be affected by personal views or perspectives. The current annotation schemes rely heavily on crowd-sourcing, where each instance within the data is guaranteed to be annotated by different annotators. However, it is not guaranteed that the same annotators would annotate the rest of the data, which would probably contain similar related instances. Consequently, since different related instance would be annotated by different annotators, the assigned labels would reflect the biases and subjectivity of different people, which might not align with each other. Thus, the labels for these instances would be inconsistent. Such inconsistencies would degrade the performance of any analyser, due to unclear boundaries between the labels. This could be the reason why the performance on SemEval's dataset peaks around an average recall of 0.60.

In addition, we found that sarcasm is very prominent within sentiment datasets. We found that 16% of the tweets we annotated are sarcastic, which is quite high. Knowing that sarcasm includes expressing opinions and emotions using indirect implicit expressions and phrases, this means that it would impose a challenge for SA systems.

8.1. Moving forward with Arabic Sentiment Analysis

The findings from the comprehensive study for Arabic sentiment analysis motivate for important future research directions in Arabic NLP. We can list them as follows:

- *Arabic Datasets Annotation*: It is important to be careful with annotating Arabic datasets, especially for subjective tasks such as sentiment analysis. Clearer guidelines and restrictive data quality measurements need to be in place to avoid getting inconsistent labels. For example, the ArSAS dataset provides an annotation confidence level with each label [13], which allows filtering out data points that might be noisy.

- *Arabic Sarcasm Detection*: The task of sarcasm detection has been trending in the recent years for English [83, 84, 85, 82]. However, there is no serious work on this task for Arabic yet. Only a couple of studies that tested some models on modest datasets [86, 87, 88]. Additional work is required in this area for Arabic, especially in the context of sarcasm effect on sentiment analysis.

- *Creating Additional Arabic Language Resources*: The best performance achieved in our experiments was using the Arabic version of BERT (AraBERT) [73], which was trained on around 3B Arabic words from news articles. While it shows very promising results, AraBERT is still much smaller than those trained on other languages, and does not cover dialects. This signifies the importance of creating a new Arabic BERT that is trained on a large amount of data that covers different Arabic dialects, which shall have the potential to further improve results in different Arabic NLP tasks in general and Arabic SA in particular.

9. Conclusion

In this paper, we presented an extensive comparative study on the different approaches for Arabic sentiment analysis. The experiments varied from using conventional classification algorithms to the more complex deep learning models. In the experiments, a large variety of hand-engineered features were used, in addition to word-embeddings for the deep learning models. The final results show the superiority of deep learning models, where they achieved the best results. Moreover, during this work, we created the largest set of Arabic word-embeddings that was created using a large corpus of 250M tweets.

Our analysis shows that the performance on some datasets was not quite high. This can be attributed to the nature of the dataset itself, as a new annotation process shows that the sentiment labels are confusing and very subjective, where many labels have been changed from the original annotation. In addition, the new annotation shows that a large portion of the available sentiment datasets is sarcastic, where the meaning is given in an implied way.

Based on the analysis provided previously, we recommended and urge the researchers on Arabic to experiment with and utilise deep learning. The work on language models is quite promising, but it needs more investigation and customisation to handle Arabic dialects. Additionally, our analysis shows that sentiment is highly affected by sarcasm. Thus, identifying sarcasm and detecting it is an essential task that needs to be explored and studied.

Acknowledgements

This work was partially supported by the Defence and Security Programme at the Alan Turing Institute, funded by the UK Government. We also thank Kareem Darwish for his valuable feedback on the general message of the paper and his suggested changes, which enriched the study significantly.

References

[1] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.
[2] S. Kolkur, G. Dantal, R. Mahe, Study of Different Levels for Sentiment Analysis, International Journal of Current Engineering and Technology 55 (22) (2015) 2277–4106.
[3] A. Yadollahi, A. G. Shahraki, O. R. Zaiane, Current state of text sentiment analysis from opinion to emotion mining, ACM Computing Surveys 50 (2) (2017) 1–33.
[4] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) 1–25.

[5] D. M. E.-D. M. Hussein, A survey on sentiment analysis challenges, Journal of King Saud University - Engineering Sciences 30 (4) (2018) 330–338.
[6] L. Yue, W. Chen, X. Li, W. Zuo, M. Yin, A survey of sentiment analysis in social media, Knowledge and Information Systems 60 (2) (2019) 617–663.
[7] S. A. Salloum, A. Q. AlHamad, M. Al-Emran, K. Shaalan, A survey of arabic text mining, in: K. Shaalan, A. E. Hassanien, F. Tolba (Eds.), Intelligent Natural Language Processing: Trends and Applications, Springer International Publishing, Cham, 2018, pp. 417–431.
[8] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, M. N. Al-Kabi, A comprehensive survey of arabic sentiment analysis, Information Processing & Management 56 (2) (2019) 320–342.
[9] G. Badaro, R. Baly, H. Hajj, W. El-Hajj, K. B. Shaban, N. Habash, A. Al-Sallab, A. Hamdi, A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations, ACM Transactions on Asian and Low-Resource Language Information Processing 18 (3) (2019) 1–52.
[10] I. Guellil, F. Azouaou, M. Mendoza, Arabic sentiment analysis: studies, resources, and tools, Social Network Analysis and Mining 9 (1) (2019) 1–17.
[11] S. Rosenthal, N. Farra, P. Nakov, SemEval-2017 task 4: Sentiment analysis in twitter, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 502–518.
[12] M. Nabil, M. Aly, A. Atiya, ASTD: Arabic sentiment tweets dataset, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 2515–2519.
[13] A. Elmadany, H. Mubarak, W. Magdy, An arabic speech-act and sentiment corpus of tweets, in: Proceedings of the 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, 2018, pp. 20–25.
[14] K. Darwish, W. Magdy, Arabic information retrieval, Foundations and Trends in Information Retrieval 7 (4) (2014) 239–342.
[15] N. Y. Habash, Introduction to Arabic Natural Language Processing, Synthesis Lectures on Human Language Technologies 3 (1) (2010) 1–187.
[16] M. Abdul-Mageed, M. Diab, M. Korayem, Subjectivity and sentiment analysis of modern standard Arabic, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 587–591.
[17] A. Hamdi, K. Shaban, A. Zainal, A review on challenging issues in arabic sentiment analysis, Journal of Computer Science 12 (9) (2016) 471–481.
[18] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? sentiment classification using machine learning techniques, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2002, pp. 79–86.
[19] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford (2009) 1–6.
[20] A. Pak, P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), European Language Resources Association (ELRA), Valletta, Malta, 2010, pp. 1320–1326.
[21] E. Kouloumpis, T. Wilson, J. Moore, Twitter sentiment analysis: The good the bad and the omg!, in: Fifth International AAAI conference on weblogs and social media, 2011, pp. 538–541.
[22] S. Batra, D. Rao, Entity based sentiment analysis on twitter, CS224N Project Report, Stanford (2010) 1–12.
[23] L. D. Caro, M. Grella, Sentiment analysis via dependency parsing, Computer Standards & Interfaces 35 (5) (2013) 442–453.
[24] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, T. Wilson, SemEval-2013 task 2: Sentiment analysis in twitter, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 312–320.
[25] S. Mohammad, S. Kiritchenko, X. Zhu, NRC-canada: Building the state-of-the-art in sentiment analysis of tweets, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval-2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 321–327.
[26] S. Rosenthal, A. Ritter, P. Nakov, V. Stoyanov, SemEval-2014 task 9: Sentiment analysis in twitter, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014), Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 73–80.
[27] Y. Miura, S. Sakaki, K. Hattori, T. Ohkuma, 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), Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 628–632.
[28] D. Tang, F. Wei, B. Qin, T. Liu, M. Zhou, Coooolll: A deep learning system for twitter sentiment classification, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 208–212.
[29] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, V. Stoyanov, SemEval-2015 task 10: Sentiment analysis in twitter, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 451–463.
[30] M. Hagen, M. Potthast, M. Büchner, B. Stein, Webis: An ensemble for twitter sentiment detection, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 582–589.
[31] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, M. Jaggi, 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 (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 1124–1128.

[32] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, V. Stoyanov, SemEval-2016 task 4: Sentiment analysis in twitter, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 1–18.

[33] M. Rouvier, B. Favre, 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), Association for Computational Linguistics, San Diego, California, 2016, pp. 202–208.

[34] C. Baziotis, N. Pelekis, C. Doulkeridis, 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), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 747–754.

[35] M. Cliche, BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 573–580.

[36] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237.

[37] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339.

[38] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[39] A. D'Andrea, F. Ferri, P. Grifoni, T. Guzzo, Article: Approaches, tools and applications for sentiment analysis implementation, International Journal of Computer Applications 125 (3) (2015) 26–33, published by Foundation of Computer Science (FCS), NY, USA.

[40] M. Abdul-Mageed, M. Diab, M. Korayem, Subjectivity and sentiment analysis of modern standard Arabic, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 587–591.

[41] M. Abdul-Mageed, M. Diab, 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 2012), European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 3907–3914.

[42] M. Abdul-Mageed, M. Diab, S. Kbler, Samar: Subjectivity and sentiment analysis for arabic social media, Computer Speech & Language 28 (1) (2014) 20–37.

[43] M. Abdul-Mageed, Modeling arabic subjectivity and sentiment in lexical space, Information Processing & Management 56 (2) (2019) 291–307.

[44] M. Abdul-Mageed, Not all segments are created equal: Syntactically motivated sentiment analysis in lexical space, in: Proceedings of the Third Arabic Natural Language Processing Workshop, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 147–156.

[45] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, B. Gupta, Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels reviews, Journal of Computational Science 27 (2018) 386–393.

[46] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, G. Eryiğit, SemEval-2016 task 5: Aspect based sentiment analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 19–30.

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

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

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

[50] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, P. Duan, Word embeddings and convolutional neural network for Arabic sentiment classification, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 2418–2427.

[51] M. Aly, A. Atiya, LABR: A large scale Arabic book reviews dataset, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 494–498.

[52] A. A. Altowayan, L. Tao, Word embeddings for arabic sentiment analysis, in: 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 3820–3825.

[53] A. M. Alayba, V. Palade, M. England, R. Iqbal, Arabic language sentiment analysis on health services, in: 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017, pp. 114–118.

[54] A. Al Sallab, H. Hajj, G. Badaro, R. Baly, W. El Hajj, K. Bashir Shaban, Deep learning models for sentiment analysis in Arabic, in: Proceedings of the Second Workshop on Arabic Natural Language Processing, Association for Computational Linguistics, Beijing, China, 2015, pp. 9–17.

[55] S. R. El-Beltagy, M. El Kalamawy, A. B. Soliman, NileTMRG at SemEval-2017 task 4: Arabic sentiment analysis, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 790–795.

[56] M. Jabreel, A. Moreno, 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), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 694–699.

[57] A. M. Alayba, V. Palade, M. England, R. Iqbal, A combined cnn and lstm model for arabic sentiment analysis, in: A. Holzinger, P. Kieseberg, A. M. Tjoa, E. Weippl (Eds.), Machine Learning and Knowledge Extraction, Springer International Publishing, Cham, 2018, pp. 179–191.

[58] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, Y. Jararweh, Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews, International Journal of Machine Learning and Cybernetics 10 (8) (2019) 2163–2175.

[59] N. Al-Twairesh, H. Al-Negheimish, Surface and deep features ensemble for sentiment analysis of arabic tweets, IEEE Access 7 (2019) 84122–84131.

[60] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, S. Belfkih, Asa: A framework for arabic sentiment analysis, Journal of Information Science 46 (4) (2020) 544–559.

[61] K. Darwish, W. Magdy, A. Mourad, Language processing for arabic microblog retrieval, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 2427–2430.

[62] S. R. El-Beltagy, T. Khalil, A. Halaby, M. Hammad, Combining lexical features and a supervised learning approach for arabic sentiment analysis, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Springer International Publishing, Cham, 2018, pp. 307–319.

[63] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st International Conference on Learning Representations, 2013, pp. 1–12.

[64] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543.

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

[66] M. Heikal, M. Torki, N. El-Makky, Sentiment analysis of arabic tweets using deep learning, Procedia Computer Science 142 (2018) 114–122.

[67] I. Abu Farha, W. Magdy, Mazajak: An online Arabic sentiment analyser, in: Proceedings of the Fourth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, Florence, Italy, 2019, pp. 192–198.

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

[69] S. R. El-Beltagy, Nileulex: A phrase and word level sentiment lexicon for egyptian and modern standard arabic, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Paris, France, 2016, pp. 2900–2905.

[70] O. ElJundi, W. Antoun, N. El Droubi, H. Hajj, W. El-Hajj, K. Shaban, hULMonA: The universal language model in Arabic, in: Proceedings of the Fourth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, Florence, Italy, 2019, pp. 68–77.

[71] S. Merity, N. S. Keskar, R. Socher, Regularizing and optimizing LSTM language models, in: 6th International Conference on Learning Representations, 2018, pp. 1–13.

[72] A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, R. Roth, 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 2014), European Language Resources Association (ELRA), Reykjavik, Iceland, 2014, pp. 1094–1101.

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

[74] A. Abdelali, K. Darwish, N. Durrani, H. Mubarak, 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, Association for Computational Linguistics, San Diego, California, 2016, pp. 11–16.

[75] T. Kudo, Subword regularization: Improving neural network translation models with multiple subword candidates, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 66–75.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[77] F. Chollet, et al., Keras, `https://keras.io` (2015).

[78] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga,

S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL `https://www.tensorflow.org/`

[79] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, 2015, pp. 1–15.

[80] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface's transformers: State-of-the-art natural language processing, ArXiv abs/1910.03771.

[81] J.-Á. González, F. Pla, L.-F. Hurtado, ELiRF-UPV at SemEval-2017 task 4: Sentiment analysis using deep learning, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 723–727.

[82] S. Oprea, W. Magdy, Exploring author context for detecting intended vs perceived sarcasm, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2854–2859.

[83] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 704–714.

[84] D. Bamman, N. Smith, Contextualized sarcasm detection on twitter, in: Ninth International AAAI Conference on Web and Social Media, 2015, pp. 574–577.

[85] G. Abercrombie, D. Hovy, Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations, in: Proceedings of the ACL 2016 Student Research Workshop, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 107–113.

[86] J. Karoui, F. B. Zitoune, V. Moriceau, Soukhria: Towards an irony detection system for arabic in social media, Procedia Computer Science 117 (2017) 161–168.

[87] I. Abbes, W. Zaghouani, O. El-Hardlo, F. Ashour, DAICT: A dialectal Arabic irony corpus extracted from Twitter, in: Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020), European Language Resources Association, Marseille, France, 2020, pp. 6265–6271.

[88] I. Abu Farha, W. Magdy, 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, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 32–39.