BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis

Maha Jarallah Althobaiti
Department of Computer Science
College of Computers and Information Technology
Taif University
Taif 21944, Saudi Arabia

***Abstract*—The user-generated content on the internet including that on social media may contain offensive language and hate speech which negatively affect the mental health of the whole internet society and may lead to hate crimes. Intelligent models for automatic detection of offensive language and hate speech have attracted significant attention recently. In this paper, we propose an automatic method for detecting offensive language and fine-grained hate speech from Arabic tweets. We compare between BERT and two conventional machine learning techniques (SVM, logistic regression). We also investigate the use of sentiment analysis and emojis descriptions as appending features along with the textual content of the tweets. The experiments shows that BERT-based model gives the best results, surpassing the best benchmark systems in the literature, on all three tasks: (a) offensive language detection with 84.3% F1-score, (b) hate speech detection with 81.8% F1-score, and (c) fine-grained hate-speech recognition (e.g., race, religion, social class, etc.) with 45.1% F1-score. The use of sentiment analysis slightly improves the performance of the models when detecting offensive language and hate speech but has no positive effect on the performance of the models when recognising the type of the hate speech. The use of textual emoji description as features can improve or deteriorate the performance of the models depending on the size of the examples per class and whether the emojis are considered among distinctive features between classes or not.**

Keywords*—Deep learning, hate speech detection; offensive language detection; sentiment analysis; transformer-based model; BERT; emoji

I. Introduction

The pervasiveness of hatred and offensive content on the internet has become disturbing, raising an alarm over negative consequences for the target individuals' mental health and the internet society's well-being [1], [2]. Online hateful and offensive language detection aims to make the internet not only accessible but also safe, as hateful speech online threatens society by encouraging hate crimes [3]. It also enables the scientific analyses of such abusive languages, covers their causes, and establishes possible solutions. Thus, in recent years, Artificial Intelligence (AI) and Natural Language Processing (NLP) communities have investigated various techniques as potential solutions for automatically detecting offensive language online with high performance [4], [5], [6], [7], [8], [9], [10].

Recently, a series of workshops and shared tasks have been conducted to explore the problem from various perspectives. Significant attention has been given to defining the problem and investigating the automatic detection techniques of offensive language with all its types and ways, including abuse, aggression, cyberbullying, and hateful content. For example, in 2018, there was the first workshop on Trolling, Aggression and Cyberbullying (TRAC-2018) [11], [12]. In addition, there have been a series of five workshops on online abusive language and harms since 2017 [13], [14], [15], [16], [17]. The sixth edition of this workshop (6th WOAH) will be held on July 14th with the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) [18]. Two editions of shared tasks on offensive language identification were organised at the international workshop on Semantic Evaluation (SemEval) in 2019 [19] and 2020 [20]. Regarding Arabic, there was a shared task on offensive language detection for Arabic at the 4th workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4) [21]. Another shared task on fine-grained hate speech detection on Arabic Twitter will be held on 20 June at the 5th workshop on Open-Source Arabic Corpora and Processing Tools (OSACT5) co-located with LREC 2022 [22].

Several categories have been adopted to define aggressive languages in online content. Among them, [23] classifies online content into hate speech, offensive, neither offensive nor hate-speech, while [24] classifies online content into abusive, hateful, normal, or spam. The study of [25] classifies online comments as racist, sexist, or neither. In addition, [4] proposed a typology of all works that have been grouped under the label of hate speech, cyberbullying, and online abuse. They synthesised the work on online abusive language in a two-fold typology that considers whether (a) the abuse is directed at a specific target and (b) the degree to which it is explicit.

In this study, we propose methods for the following three tasks: (a) offensive language detection (identifying whether a tweet is offensive or not), (b) hate speech detection (i.e., identifying whether a tweet has hate speech or not), and (c) fine-grained hate speech detection (i.e., identifying and recognising the type of hate speech: disability, gender, ideology, race, religion, or social class). We utilise the dataset released by [26] which contains 12,698 Arabic tweets annotated for the three aforementioned tasks. We also investigate the use of two conventional machine learning techniques: Support Vector Machine (SVM) and Logistic Regression (logit). We also explore Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art transformer-based machine learning technique for deep-contextualised word representation. In addition, sentiment analysis and emoji description are explored as potential features that can be utilised in training any model to improve its performance, as our intuition indicates that hate speech and offensive language mostly express negativity which can be exploited. The contributions of this study are summarised as follows.

- We investigate and compare conventional machine-learning techniques and a transfer-based model (BERT) for offensive language and fine-grained hate speech detection.
- We examine the use of sentiment analysis and emoji descriptions as additional textual features for both transformer-based models and conventional machine learning methods.
- We examine our proposed methods on relatively small unbalanced data and with different preprocessing settings.
- We develop a novel and simple method for offensive language and hate-speech detection that outperforms the best benchmark systems reported in the literature with the released dataset used in our study.

II. Related Work

A considerable number of studies for detecting online hate speech and offensive language have been suggested and investigated in the literature in the past ten years but intensively since 2017 [27], [28], [29], [30], [7], [31]. Workshops and shared tasks organised for the task of detecting and recognising online offensive language, hate-speech, and abusive content played a vital role in attracting the attention of the research community to propose potential techniques for the task [14], [13], [11], [15], [20], [18]. Many studies have examined generalised solutions for offensive language detection from online content in multiple languages, while other studies concentrated on examining the suitable features and techniques for one language, such as Greek [32], Chinese [33], Slovene [34], and Croatian [35]. Significant attention has been paid to detect offensive language from online English content [36], [5].

Few studies have been conducted to address the problem of online anti-social behaviour on Arabic; most have targeted offensive language detection, while the remaining studies have investigated the problem of hate speech detection [21], [37], [38], [39]. One of the early works, conducted by [38], targeted vulgar and pornographic obscene speech on Arabic social media using a list-based approach. They used tweets to build a list of seed words for obscene phrases. Then, they employed the list to construct three sublists of obscene words and phrases using multiple measurements, such as the Log Odds Ratio (LOR) for unigrams and bigrams. Conventional Machine Learning (ML) techniques have also been investigated for offensive language and hate speech detection [40], [41]. The most commonly used traditional ML techniques for offensive language and hate-speech detection are SVM [37], [39], [42], [43], [44], Naive Bayes [43], [44], and Logistic Regression (logit) [45], [46].

The study in [47] examined the use of FastText Deep Learning (DL) model on a dataset containing 36 million tweets to detect offensive speech. They reported that the FastText DL model outperformed an SMV classifier trained on character n-gram features. Mohaouchane, Mourhir, and Nikolov [48] explored the use of AraVec word embeddings and four DL models: Bidirectional Long Short-Term Memory (Bi-LSTM), Bi-LSTM with an attention mechanism, Convolutional Neural Network (CNN), and a combined model of CNN and LSTM. The experiments illustrated the outperforming results of the CNN over all other models. Many other architectures of deep neural networks have been investigated, such as Gated Recurrent Unit (GRU) [49], RNN [41], [45], and contextual embeddings (e.g., multilingual BERT [46], [50] and AraBERT [51]).

The results of the techniques in most of the aforementioned studies cannot be compared because every study used their own dataset and the available datasets for Arabic offensive language and hate-speech detection are limited [39], [43], [44]. However, the shared task on offensive language and hate speech detection in the fourth workshop on Open-Source Arabic Corpora and Corpora Processing Tools (OSACT4) provides a manually annotated Twitter dataset, consisting of 10,000 examples, for offensiveness (labels are: OFF or NOT_OFF) and for specifying the offensive content type of an offensive example as hate speech or not (labels are: HS or NOT_HS). This provides an opportunity to compare techniques for both tasks: offensive language detection and hate speech detection. The winning team for Arabic offensive language detection has employed an ensemble system of traditional machine learning technique (SVM) and two DL models: CNN+BiLSTM and multilingual BERT with an F1-score equal to 90.51%, while the best performing system for hate speech detection used SVM and achieved 95.2%, outperforming the second-place system by 12.9%. The winning team has attributed the performance of the winning model to the intensive preprocessing steps which included emoticons and emoji to textual description conversion, dialectal to MSA conversion, word categorisation (e.g., all animal names included in tweets were reduced to only one word.), letter normalisation, stop-word removal, and hashtag segmentation.

A study by Mubarak et al. [26] released an Arabic dataset for detecting offensive language and hate speech, consisting of 12,698 tweets. To our knowledge, this is the largest and most recent corpus so far. It was manually annotated for offensiveness, and fine-grained hate speech. In order to encourage comparisons between future studies, the providers of the dataset experimented with different transformer architectures and SVM to benchmark the dataset for detecting offense and hate speech to encourage comparisons between future studies. They fine-tuned mono- and multilingual transformer models using their training data. For the monolingual task, they utilised AraBERT and QARiB; for the multilingual models, they fine-tuned mBERT and XLM-RoBERTa. It was obvious in their reported results that monolingual models significantly outperformed the multilingual models. For offensive classification, the QARiB model achieved an F1-score equal to 82.31%, outperforming all other models, including AraBERT which came second with an F1-score equal to 80.02%. In contrast, AraBERT outperformed QARIB, achieving an F1-score of 80.14% and winning first place for hate speech detection.

III. Dataset Preparation

A. Dataset Description

We used the largest and most recently released dataset for offensive language and hate speech detection in Arabic [26]. The dataset consists of 12,698 tweets and defined according to the following: offensive language is a language containing any kind of impolite language such as insults, slurs, threats, and encouraging violence. Hate speech is any kind of offensive language that targets a person or group of people based on six common characteristics: disability, gender, ideology, race, religion, or social class. The task of offensive language detection was annotated using two labels OFF (offensive example) and NOT_OFF (not offensive example). Hate speech detection was annotated using two labels HS (hate speech example) and NOT_HS (not hate speech example). The fine-grained hate speech was annotated using 7 labels: HS1 (race/ethnicity/nationality), HS2 (f religion/belief), HS3 (ideology), HS4 (disability/disease), HS5 (social class), HS6 (gender), and NOT_HS (not hate-speech).

In our study, we passed the dataset through a set of preprocessing levels that started with cleaning the data to remove noise, followed by converting emoji into textual description, and then finding the sentiment of the tweet (i.e., positive, negative, and neutral) and appending the sentiment to the text of the tweet as additional textual features. Indeed, various levels of improved preprocessing have been examined when utilised with different techniques to identify their role in improving the performance of every built model.

B. Cleaning

The key component of any successful NLP application is to remove noise and reduce data sparsity as much as possible. It is well known that Arabic used in user-generated online content, including social media, is written in Arabic dialects which have many lexical, syntactic, and morphological differences, increasing the data sparsity of any corpus collected from online sources [52], [53]. In addition, online content is usually noisy, with a considerable number of tags, excessive spaces, repeated characters, Arabizi in which some people, when writing online, transliterate Arabic using Latin letters and numerals. In preprocessing step, we cleaned the text in order to reduce noise and data sparsity by the following:

- removing HTML tags and other symbols such as <LF>
- removing hashtags # and mentions @
- replacing underscore symbol _ of hastags into space
- removing URLs and retweets RT
- removing all types of diacritical marks, punctuation marks, mathematical signs and symbols
- removing repeated letters
- removing symbols different from emojis
- normalising different forms of alif into a bare alif (alif without hamzah), normalising taa' marbutah to haa' and normalising the dotless yaa' (alif maqsurah) to yaa'.

To perform normalisation and repeated letter removal, we used the AraNLP library [54].

C. Textual Emoji Description

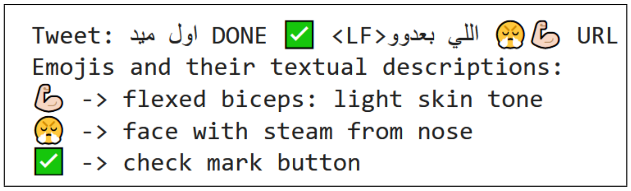
An emoji is a pictograph embedded in text in electronic communication and web pages that conveys emotional cues, attitudes, and feelings that cannot be concluded from typed conversations. They exist in various forms such as facial expressions, common objects, animals, places, and types of weather. Thus, emojis can play an important role in the detection of offensive languages. In [55], the authors observed that the most frequent personal attack on Arabic Twitter is to call a person an animal names such as (*kalb*, "dog")[1] and (*HmAr*, "donkey"). The same observation was reported in a previous study of [57]. In addition, some face emojis (anger and disgust) and objects (shoes) are widely used in offensive communication [57].

Converting an emoji to its textual description was one of the intensive preprocessing steps used to prepare the text before using it to train an SVM model in the study of [40]. Their SVM model achieved F1-score equal to 95% for detection of hate speech in Arabic text, ranking first in the shared task on Arabic offensive language detection in the OSACT4 workshop co-located with LREC 2020. We investigated the use of emoji descriptions as additional textual features that can be appended to tweets with the original text. Different settings can be examined, such as the technique utilised as well as the size and balance of the annotated examples for each class. We plan to investigate the importance of emojis themselves or their textual descriptions when used with deep contextualised word representation techniques, such as BERT, and compare it with other traditional ML techniques, such as SVM and logit.

We used "demoji" package in python [58] which accurately find emojis from a blob of text using data from the Unicode Consortium's emoji code repository. After finding emojis in each tweet, we replaced them in the text with their code (i.e, textual description) equivalents. Fig. 1 shows the results of using the demoji package on a tweet from the training data. The results of the textual descriptions of emojis are in English. Thus, we utilised the Google Translate API to convert the textual descriptions from English to Arabic.

D. Sentiment Analysis

Sentiment analysis can generally be defined as the use of NLP techniques to detect, recognise, and quantify affective states and subjective information. However, the basic idea of



```
Tweet: اول ميد DONE ✅ <LF>اللي بعدوو 😤💪 URL
Emojis and their textual descriptions:
💪 -> flexed biceps: light skin tone
😤 -> face with steam from nose
✅ -> check mark button
```

Fig. 1. Extracting Emojis form a Tweet and Finding their Descriptions.

[1] Throughout the paper, Arabic words are represented as follows: (HSB transliteration, 'English gloss'). More details about the Habash–Soudi–Buckwalter (HSB) scheme can be found in [56]

sentiment analysis is to classify the polarity of any given piece of text (i.e, at the document, sentence, or aspect level) [59], [60]. The importance of sentiment identification for any given piece of text appears in human decision-making. It is notable that offensive language and negativity have a high correlation, as the general atmoshere of offensive language is negative. In contrast, speech free of hate speech or offensive language expresses neutral or positive sentiments. The study of [23] has used a sentiment lexicon to assign sentiment scores to each tweet when detecting offensive languages. In our study, we decided to examine the use of sentiments as additional textual features when using it with deep contextualised word embeddings, such as BERT, or with traditional ML technique (SVM and logit).

We used the CAMeLBERT-DA sentiment analysis model built by fine-tuning the CAMeLBERT Dialectal Arabic (DA) model [61]. For fine-tuning, they used ASTD [62], ArSAS [63], and SemEval [64] datasets. These datasets were collected from Twitter, making them suitable for our dataset as they represent dialectal Arabic, which is mostly used in social media. This model classifies a given text as positive, negative, or neutral. We translated these sentiments into Arabic. That is, we used the Arabic words *AyjAby*, *slby*, and *muHAyd* for "positive", "negative", and "neutral" respectively. We applied the model to the dataset of tweets and then appended the Arabic translation of the sentiment of every tweet to its word components. In addition, we examined the sentiment analysis of the dataset in two different settings: (a) a tweet with its original emojis and (b) a tweet with textual descriptions of its emojis. Thus, we randomly selected 1,000 pre-processed tweets with textual emoji descriptions and the same sample of 1,000 tweets with their original emojis after applying sentiment analyser to both groups of data. Then, we compared the predicted sentiments of the analyser on the same sentence in the two groups, regardless of their correctness. We found that 97.40% of the 1,000 tweets had the same predicted sentiments in both settings (i.e., whether we left original emojis in tweets or replaced them with their textual descriptions). Table I shows some examples of tweets in which the sentiment analyser predicts different sentiments when we change the emoji representation in the tweets (original emoji vs. textual emoji description). English translations in the table are just indicative. It is obvious that some differences in predicting sentiments occur when emojis remain as they are in the tweets or replace them with their equivalent textual descriptions. However, the differences are not large and can be attributed to the manner in which the model was built and the datasets used during fine-tuning.

At the end of this stage, we prepared the dataset using various levels of preprocessing, including cleaning, appending sentiments as additional textual features, and replacing emojis with their corresponding textual descriptions. Table II illustrates the various levels of preprocessing by presenting a sentence from the corpus and the corresponding output of every preprocessing level. "CLN" indicates the tweet after cleaning. "EmoTxt" indicates the tweet after replacing emojis by their textual descriptions. "SA" indicates the tweet after analysing its sentiment and appending it to the text of the tweet. English translations are just indicative.

TABLE I. Examples of different Predicted Sentiments by the Model when using different Emojis Representations

Sentence	Predicted Sentiment
(with Emojis) بس بقى👊 Just enough 👊	negative
(with textual Emojis description) بس بقى قدوم القبضه Just enough oncoming fist	neutral
(with Emojis) عنز 🐐🐐 Goat 🐐🐐	negative
(with textual Emojis description) عنز ماعز ماعز Goat goat goat	neutral
(with Emojis) فينك من امبارح ي🐕 Where have you been since yesterday 🐕	neutral
(with textual Emojis description) فينك من امبارح ي كلب Where have you been since yesterday dog	negative

TABLE II. A Tweet from the Corpus before and after Various Levels of Preprocessing Steps

Original Tweet	🥀. 🚶👌<LF> لن تحصل على غدٍ افضل مادمت تفكر بالامس→🌸 ↪<LF><LF> ⍰⍰ <LF><LF> <LF> • ________________ <LF> 👍 👎<LF><LF> 💗 ▸ URL → 🌸 ↪<LF><LF> ⍰⍰ <LF><LF>You won't get a better tomorrow if you think about yesterday. 🚶👌<LF> <LF> • ________________ <LF> 👍 👎<LF><LF> 💗 ▸ URL
CLN	💗 👎 👍 👌 🚶 🥀 لن تحصل علي غد افضل مادمت تفكر بالامس🌸 🌸 You won't get a better tomorrow if you think about yesterday 🥀 🚶 👌 👍 👎 💗
CLN+ SA	محايد 💗 👎 👍 👌 🚶 🥀 لن تحصل علي غد افضل مادمت تفكر بالامس🌸 🌸 You won't get a better tomorrow if you think about yesterday 🥀 🚶 👌 👍 👎 💗 Neutral
CLN+ EmoTxt	زهرة الكرز لن تحصل علي غد افضل مادمت تفكر بالامس زهرة ذابلة شخص يمشي جهة موافق ممتاز استهجن ينمو القلب cherry blossom You won't get a better tomorrow if you think about yesterday wilted flower person walking OK hand thumbs up thumbs down growing heart
CLN+ EmoTxt+ SA	زهرة الكرز لن تحصل علي غد افضل مادمت تفكر بالامس زهرة ذابلة شخص يمشي جهة موافق ممتاز استهجن ينمو القلب ايجابي cherry blossom You won't get a better tomorrow if you think about yesterday wilted flower person walking OK hand thumbs up thumbs down growing heart positive

IV. Classification Models

We utilised conventional ML techniques (SVM and logit) and deep learning technique (BERT) to perform three tasks: a) detecting if a tweet is offensive or not offensive; b) identifying the type of offensiveness whether it is hate speech or not; and c) identifying the type of hate speech based on race/ethnicity/nationality, religion/belief, ideology, disability/disease, social class, or gender. This section explains the three approaches adopted for the three tasks and the utilised features.

A. BERT Model Classifier

BERT [65] is a multilayer bidirectional Transformer encoder based on the original implementation of transformer architecture introduced by Vaswani et al. [66]. The BERT model resulted in significant improvements in a considerable number of downstream tasks. Furthermore, a wide range of research works on Arabic hate speech and offensive language detection, including those participating in the 2020 shared task on Arabic offensive language detection, have proven its potential to handle the task [21]. In addition, the BERT-based model was the best benchmark system trained on the same dataset we employed in this study, allowing us to compare our proposed method with the best benchmark system.

In our study, we built a BERT-based model by fine-tuning AraBERT [67] on the training data. We selected "AraBERTv0.2-Twitter-base" variant of AraBERT that supports emojis and dialectal Arabic words. We also applied a segmentation function using Farasa to segment the text for the model. We built five BERT models for each task (i.e., offensive language detection, hate-speech detection, and fine-grained hate-speech detection), resulting in a total of 15 models. The five models were built using five versions of the dataset according to various levels of preprocessing, as previously illustrated in Table II.

B. SVM Classifier

We used word n-grams with *n* in the range [1, 3] weighted using Term Frequency-Inverse Document Frequency (TF-IDF). We also used character n-grams with *n* in the range [2, 5] only from the text inside word boundaries using token counts. The word-based TF-IDF vector and character-based count vector were used as features to train the SVM. As in the BERT model, we built 15 SVM classifiers to examine the various situations: three tasks in addition to the five levels of preprocessing the dataset to prepare it before building the models.

C. Logistic Regression Classifier

The same features used in the SVM were examined using a logistic regression classifier. Therefore, we used word n-grams with *n* in the range [1, 3] weighted using Term Frequency-Inverse Document Frequency (TF-IDF). We also used character n-grams with *n* in the range [2, 5] only from the text inside word boundaries using token counts. The word-based TF-IDF vector and character-based count vector were utilised as features to train the logistic regression classifier. The sklearn package in Python was used to train the classifier. The maximum number of iterations for logistic regression in the package was set to 100 by default. In our experiments, we increased this value to 800 iterations to obtain the trained model. Similar to the BERT model and SVM classifier, we built 15 logit classifiers.

V. Experimental Setup

We used the same splits prepared by the data providers where the dataset was partitioned into three parts: 8,888 (70%) for training, 1,269 (10%) for development, and 2,541 (20%) for testing. Table III and Table IV show the distribution of offensive and hate speech data.

TABLE III. Distribution of Offensive and Hate Speech Data [26]

	Train	Dev	Test	Total
OFF	3,172	404	887	4,463
NOT_OFF	5,716	865	1,654	8,235
HS	959	109	271	1,339
NOT_HS	7,929	1,160	2,270	11,359
Total	8,888	1,269	2,541	12,698

TABLE IV. Distribution of Fine-Grained Hate Speech Data. "N.A." Stands for not Available

	Train	Dev	Test
HS1	260	28	N.A.
HS2	27	4	N.A.
HS3	144	14	N.A.
HS4	1	0	N.A.
HS5	72	10	N.A.
HS6	456	52	N.A.
NOT_HS	7928	1161	2,270
Total	8,888	1,269	2,541

At the time of writing the paper, the gold-standard labels for the training and development sets were publicly available, whereas only the tweets of the test set were available without the gold-standard labels. The providers of the data, however, accepted our request and helped us evaluate our best-performing model on the labelled test set and provided us with the results of our best performing model for all three tasks. Therefore, we utilised the results of our built models evaluated on development set in order to compare between them and to evaluate the various preprocessing settings we suggested in this paper. Next, the results of our best-performing model when evaluated on the test set were then compared with the benchmark systems [26] that were trained and tested using the same dataset we used in this study. The employed evaluation metrics in our study are macro-averaged precision, recall and F1- score, in addition to accuracy.

VI. Results and Discussion

The results of the SVM classifiers evaluated on the development set for the three tasks are presented in Table V. These classifiers are trained on the versions of the dataset resulting from the five different levels of preprocessing: Orgi (original tweets), CLN (after cleaning tweets from noise), CLN+SA (after cleaning and appending the sentiment of the tweet to its text), CLN+EmoTxt (after cleaning and replacing emojis with their textual descriptions), and CLN+Emotxt+SA (after cleaning, replacing emojis with their textual descriptions, and appending the sentiment of the tweet to its text). We used the macro-averaged F1-score to rank the various classifiers for each task. For offensive language detection, we observe that the SVM classifier leads to the best results after cleaning the dataset, replacing emojis with their textual descriptions and appending the sentiment of each example to its text. For hate speech detection, the best performing SVM classifier is obtained using sentiment analysis, but without the need for emojis conversion. For fine-grained hate speech detection, it

TABLE V. ACCURACY AS WELL AS MACRO-AVERAGED (P)RECISION, (R)ECALL AND F1 SCORE OF SVM CLASSIFIERS ON DEVELOPMENT SET

Offensive language detection				
	Acc	P	R	F1
Orgi	79.84	78.81	72.48	74.25
CLN	80.00	78.78	72.92	74.64
CLN+SA	80.94	79.58	**74.61**	76.22
CLN+EmoTxt	80.39	79.45	73.28	75.07
CLN+EmoTxt+SA	81.26	**80.54**	74.44	**76.28**
Hate speech detection				
	Acc	P	R	F1
Orgi	92.44	85.48	58.04	61.64
CLN	92.68	85.78	59.83	64.12
CLN+SA	92.83	86.57	**60.75**	**65.37**
CLN+EmoTxt	92.68	85.78	59.83	64.12
CLN+EmoTxt+SA	92.83	**87.61**	60.34	64.89
Fine-grained hate speech detection				
	Acc	P	R	F1
Orgi	92.13	37.50	18.57	20.23
CLN	92.36	**37.92**	**19.63**	**21.79**
CLN+SA	92.28	37.72	19.35	21.48
CLN+EmoTxt	92.13	23.41	18.33	19.56
CLN+EmoTxt+SA	92.13	23.41	18.33	19.56

TABLE VI. ACCURACY AS WELL AS MACRO-AVERAGED (P)RECISION, (R)ECALL AND F1 SCORE OF LOGISTIC REGRESSION CLASSIFIERS ON DEVELOPMENT SET

Offensive language detection				
	Acc	P	R	F1
Orgi	78.66	76.77	71.48	73.01
CLN	79.61	77.90	72.83	74.39
CLN+SA	80.55	78.53	74.84	76.16
CLN+EmoTxt	79.61	77.66	73.16	74.62
CLN+EmoTxt+SA	81.10	**79.69**	**74.92**	**76.50**
Hate speech detection				
	Acc	P	R	F1
Orgi	93.07	87.59	62.13	67.18
CLN	93.07	86.72	**62.54**	67.61
CLN+SA	93.15	87.90	62.19	**67.77**
CLN+EmoTxt	93.07	88.59	61.71	66.74
CLN+EmoTxt+SA	92.97	**88.72**	61.78	66.90
Fine-grained hate speech detection				
	Acc	P	R	F1
Orgi	91.99	51.97	**22.34**	23.91
CLN	92.76	**53.42**	22.03	**25.25**
CLN+SA	92.76	52.71	21.77	24.39
CLN+EmoTxt	92.77	52.86	21.75	24.87
CLN+EmoTxt+SA	92.91	52.86	21.75	24.87

is sufficient to clean the dataset from noises before training the SVM classifier in order to obtain the best results of the algorithm for the task.

The results of the logistic regression (logit) classifiers for the three tasks are presented in Table VI. The logit classifier for hate speech detection achieves the best performance when using sentiment analysis as well as cleaning noisy data, yielding an F1-score of 67.77. Both SVM and logit classifiers require the same preprocessing level (CLN+SA) to achieve the best performance. For offensive language detection, the use of emojis conversion and sentiment analysis when preparing the dataset plays an important role in obtaining the best performing logit classifier, achieving an F1-score equal to 76.50 (slightly better than the best SVM classifier for the same task which achieves F1-score = 76.28). For fine-grained hate speech detection, the only required preprocessing of the data to build a logit classifier with the best performance is to clean the text from noise. In fact, the use of sentiments as additional features or converting emojis into their textual code leads to a decline in the performance of the built classifier. This observation matches what we noticed with the SVM classifiers for fine-grained hate speech detection.

The BERT models outperformed all other SVM and logistic regression classifiers regardless of the preprocessing level used to prepare the dataset, as shown in Table VII. The best performing BERT models achieved an F1-score equal to 85.93, 81.89, and 48.72 for offensive language detection, hate speech detection, and fine-grained hate speech detection, respectively. Regarding the optimal preprocessing steps that can be applied to the dataset before training to improve the model's performance regardless of the utilised ML technique, we observe that CLN+EmoTxt+SA (i.e., cleaning the data, converting emojis to textual words, and appending sentiments as additional textual features) always improves the performance of the model for the offensive language detection task. This can be attributed to the fact that offensive language detection is considered easier than detecting hate speech or identifying the exact type of hate speech. In offensive language detection, every tweet that contains an impolite language, including hate speech, is considered offensive language according to the annotation guidelines followed by the providers of the dataset [26]. Therefore, adding additional features, such as textual descriptions of emojis or sentiments, to each tweet will confirm the boundaries that should be learned by the algorithms to distinguish between normal and offensive tweets. That is, almost all offensive tweets have the sentiment "negative" added as additional features, while normal tweets usually have "positive" or "neutral" sentiments added as additional features. Emojis on the other hand are also considered distinctive features in the case of offensive language detection as offensive emojis that express anger and disgust are not commonly found in normal tweets (i.e., not offensive). Thus, converting emojis to textual descriptions increases the number of additional distinctive features. That is, instead of having only one emoji, we will have, by converting emoji to description, a phrase with words expressing anger and disgust. In contrast, the CLN+EmoTxt+SA usually has a fluctuating impact on the performance of the model for the task of hate speech detection, as it sometimes slightly improves the performance of the model as seen in Table V and Table VII, and sometimes deteriorates the performance of the model, as shown in Table VI. However, CLN+EmoTxt+SA did not allow the model to achieve the

TABLE VII. ACCURACY AS WELL AS MACRO-AVERAGED (P)RECISION, (R)ECALL AND F1 SCORE OF BERT MODELS ON **DEVELOPMENT SET**

Offensive language detection				
	Acc	P	R	F1
Orgi	86.77	48.51	85.55	84.98
CLN	87.63	85.77	**85.72**	85.74
CLN+SA	87.72	85.93	85.63	85.79
CLN+EmoTxt	87.95	86.51	85.36	85.87
CLN+EmoTxt+SA	87.80	**86.52**	85.40	**85.93**
Hate speech detection				
	Acc	P	R	F1
Orgi	93.88	83.60	79.41	80.91
CLN	94.57	82.62	81.90	81.76
CLN+SA	94.33	**82.67**	80.34	**81.89**
CLN+EmoTxt	94.09	81.03	81.81	81.41
CLN+EmoTxt+SA	94.09	80.85	**82.63**	81.71
Fine-grained hate speech detection				
	Acc	P	R	F1
Orgi	92.99	47.68	46.78	45.79
CLN	93.54	**49.74**	49.13	**48.72**
CLN+SA	93.62	49.46	**49.15**	48.55
CLN+EmoTxt	93.31	48.91	48.59	47.91
CLN+EmoTxt+SA	93.07	48.02	47.82	41.16

TABLE VIII. PERFORMANCE COMPARISON OF OUR BEST PERFORMING MODEL AND FOUR BENCHMARK SYSTEMS [26] ON **TEST SET**

Offensive language detection				
	Acc	P	R	F1
AraBERT	92.64	81.04	79.31	80.14
QARiB	92.99	82.99	77.72	80.04
mBERT	91.26	77.55	73.34	75.20
XLM-RoBERTa	92.29	79.96	78.79	79.36
Our Model$_{\text{best}}$	**93.30**	**83.00**	**80.70**	**81.80**
Hate speech detection				
	Acc	P	R	F1
AraBERT	82.09	80.50	79.63	80.02
QARiB	84.02	82.53	82.11	82.31
mBERT	76.43	74.09	73.32	73.66
XLM-RoBERTa	75.00	72.50	72.47	72.48
Our Model$_{\text{best}}$	**85.90**	**84.60**	**84.10**	**84.30**

best performance. This can be attributed to the fact that hate speech in the utilised dataset is considered a type of offensive language. That is, a tweet may contain impolite language and is considered offensive but not "hate-speech". Also, the general unbalance between various classes in the dataset, as seen in Table III, is more obvious in the case of hate speech as there are few annotated hate speech examples compared to not-hate-speech examples. Therefore, converting emojis to textual descriptions actually increases data sparsity and cannot be seen as a vital preprocessing step in the case of hate speech detection, as seen in Tables V, VI, and VII. In the case of fine-grained hate speech detection, there are six types of hate speech, and there is a huge unbalance between the number of annotated examples for these types. For example, we have 260 tweets labelled as "HS1" (race/ethnicity/nationality), 27 tweets were annotated as "HS2" (religion/belief), and 7,928 were annotated as not-hate-speech. Therefore, the preprocessing step of emoji conversion does not have a significant positive effect on the overall performance of the model, as it may increase the data sparsity, especially for unbalanced datasets with few annotated examples and overlapping classes. However, we observe that cleaning the data improves the performance of the model, regardless of the utilised training algorithm. In addition, the use of sentiments as additional features appended to the tweet's text has a good impact on the overall performance of the model. This positive impact is less obvious when the number of annotated examples is insufficient, to distinguish between a large number of overlapping classes, such as in the case of fine-grained hate speech detection.

The best performing model (BERT model) for each task was selected to be evaluated on the test part of the dataset. The results are presented in Table VIII, which shows that our best performing model with its suggested preprocessing levels outperforms all other benchmark models for two tasks: offensive language detection and hate speech detection [26]. The study in [26] did not provide a benchmark system for fine-grained hate speech detection. However, our proposed BERT-based model achieved an F1-score equal to 45.10% on the test set. The precision, recall, and accuracy were 48.20%, 46.10%, and 92.10% respectively.

VII. CONCLUSION

In this study, we proposed an automatic method for detecting offensive language and fine-grained hate speech from Arabic tweets. We compared BERT with two conventional machine learning techniques (SVM and logistic regression). We also investigated the use of sentiments and textual descriptions of emojis as appending features in the dataset, along with the textual content of the tweets. The experiments clarified that the BERT-based model results in the best performance, surpassing the best benchmark systems in the literature, for all three tasks: (a) offensive language detection with an 84.3% F1-score, (b) hate speech detection with an 81.8% F1- score, and (c) fine-grained hate-speech recognition (e.g., race, religion, social class, etc.) with a 45.1% F1-score. Analysing the sentiment of each tweet and using it as a feature slightly improves the performance of the models when detecting offensive language and hate speech, but has little positive effect on the performance of models for recognising the type of hate speech. The use of textual emoji descriptions as features can improve or deteriorate the performance of the models depending on the size of the annotated examples per class and whether the emojis are considered distinctive features between classes. That is, when the number of annotated examples is limited while the classes overlap in the feature space, emojis may not be considered as distinctive features, and converting them to their textual description as additional features may increase the data sparsity and, therefore, deteriorate the performance of the model. However, our proposed models and various levels of preprocessing lead to better results than the benchmark systems reported in the previous study.

REFERENCES

[1] G. S. O'Keeffe, K. Clarke-Pearson *et al.*, "The impact of social media on children, adolescents, and families," *Pediatrics*, vol. 127, no. 4, pp. 800–804, 2011.

[2] E. R. Munro, "The protection of children online: a brief scoping review to identify vulnerable groups," *Childhood Wellbeing Research Centre*, 2011.

[3] P. Burnap, O. Rana, M. Williams, W. Housley, A. Edwards, J. Morgan, L. Sloan, and J. Conejero, "Cosmos: Towards an integrated and scalable service for analysing social media on demand," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 30, no. 2, pp. 80–100, 2015.

[4] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, "Understanding abuse: A typology of abusive language detection subtasks," in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 78–84.

[5] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*. Association for Computational Linguistics, 2019, pp. 1–10.

[6] P. Mishra, H. Yannakoudakis, and E. Shutova, "Tackling online abuse: A survey of automated abuse detection methods," *arXiv preprint arXiv:1908.06024*, 2019.

[7] M. Wiegand, M. Geulig, and J. Ruppenhofer, "Implicitly abusive comparisons–a new dataset and linguistic analysis," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 358–368.

[8] B. Vidgen, T. Thrush, Z. Waseem, and D. Kiela, "Learning from the worst: Dynamically generated datasets to improve online hate detection," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 1667–1682.

[9] R. Hada, S. Sudhir, P. Mishra, H. Yannakoudakis, S. Mohammad, and E. Shutova, "Ruddit: Norms of offensiveness for english reddit comments," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 2700–2717.

[10] P. Röttger, B. Vidgen, D. Nguyen, Z. Waseem, H. Margetts, and J. Pierrehumbert, "Hatecheck: Functional tests for hate speech detection models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 41–58.

[11] R. Kumar, A. K. Ojha, M. Zampieri, and S. Malmasi, Eds., *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. [Online]. Available: https://aclanthology.org/W18-4400

[12] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1–11. [Online]. Available: https://aclanthology.org/W18-4401

[13] Z. Waseem, W. H. K. Chung, D. Hovy, and J. Tetreault, Eds., *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017. [Online]. Available: https://aclanthology.org/W17-3000

[14] D. Fišer, R. Huang, V. Prabhakaran, R. Voigt, Z. Waseem, and J. Wernimont, Eds., *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018. [Online]. Available: https://aclanthology.org/W18-5100

[15] S. T. Roberts, J. Tetreault, V. Prabhakaran, and Z. Waseem, Eds., *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019. [Online]. Available: https://aclanthology.org/W19-3500

[16] S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Online: Association for Computational Linguistics, Nov. 2020. [Online]. Available: https://aclanthology.org/2020.alw-1.0

[17] A. Mostafazadeh Davani, D. Kiela, M. Lambert, B. Vidgen, V. Prabhakaran, and Z. Waseem, Eds., *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. Online: Association for Computational Linguistics, Aug. 2021. [Online]. Available: https://aclanthology.org/2021.woah-1.0

[18] Z. Waseem, B. Vidgen, L. Mathias, and A. Davani, "The 6th Workshop on Online Abuse and Harms (2022)," 2022, [Online]. Available: Available: https:https://www.workshopononlineabuse.com/ (accessed 20-April-2022).

[19] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 75–86.

[20] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin, "SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020)," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 1425–1447. [Online]. Available: https://aclanthology.org/2020.semeval-1.188

[21] H. Mubarak, K. Darwish, W. Magdy, T. Elsayed, and H. Al-Khalifa, "Overview of OSACT4 Arabic offensive language detection shared task," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 48–52. [Online]. Available: https://aclanthology.org/2020.osact-1.7

[22] H. Mubarak, "Arabic Hate Speech 2022 Shared Task!" 2022, [Online]. Available: Available: https:https://sites.google.com/view/arabichate2022/home (accessed 20-April-2022).

[23] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, 2017, pp. 512–515.

[24] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–30, 2018.

[25] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[26] H. Mubarak, S. Hassan, and S. A. Chowdhury, "Emojis as anchors to detect arabic offensive language and hate speech," *arXiv preprint arXiv:2201.06723*, 2022.

[27] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & internet*, vol. 7, no. 2, pp. 223–242, 2015.

[28] M. Dadvar, D. Trieschnigg, and F. d. Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Canadian conference on artificial intelligence*. Springer, 2014, pp. 275–281.

[29] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. d. Jong, "Improving cyberbullying detection with user context," in *European Conference on Information Retrieval*. Springer, 2013, pp. 693–696.

[30] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the first workshop on abusive language online*, 2017, pp. 85–90.

[31] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, "Detecting cyberbullying: query terms and techniques," in *Proceedings of the 5th annual acm web science conference*, 2013, pp. 195–204.

[32] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos, "Deep learning for user comment moderation," *arXiv preprint arXiv:1705.09993*, 2017.

[33] H.-P. Su, Z.-J. Huang, H.-T. Chang, and C.-J. Lin, "Rephrasing profanity in chinese text," in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 18–24.

[34] D. Fišer, T. Erjavec, and N. Ljubešić, “Legal framework, dataset and annotation schema for socially unacceptable online discourse practices in slovene,” in *Proceedings of the first workshop on abusive language online*, 2017, pp. 46–51.

[35] N. Ljubešić, T. Erjavec, and D. Fišer, “Datasets of slovene and croatian moderated news comments,” in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018, pp. 124–131.

[36] I. Clarke and J. Grieve, “Dimensions of abusive language on twitter,” in *Proceedings of the first workshop on abusive language online*, 2017, pp. 1–10.

[37] E. A. Abozinadah, A. V. Mbaziira, and J. Jones, “Detection of abusive accounts with arabic tweets,” *International Journal of Knowledge Engineering-IACSIT*, vol. 1, no. 2, pp. 113–119, 2015.

[38] H. Mubarak, K. Darwish, and W. Magdy, “Abusive language detection on arabic social media,” in *Proceedings of the first workshop on abusive language online*, 2017, pp. 52–56.

[39] A. Alakrot, L. Murray, and N. S. Nikolov, “Dataset construction for the detection of anti-social behaviour in online communication in arabic,” *Procedia Computer Science*, vol. 142, pp. 174–181, 2018.

[40] F. Husain, “Osact4 shared task on offensive language detection: Intensive preprocessing-based approach,” in *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, p. 53.

[41] A. I. Alharbi and M. Lee, “Combining character and word embeddings for the detection of offensive language in arabic,” in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 91–96.

[42] A. Alakrot, L. Murray, and N. S. Nikolov, “Towards accurate detection of offensive language in online communication in arabic,” *Procedia computer science*, vol. 142, pp. 315–320, 2018.

[43] H. Haddad, H. Mulki, and A. Oueslati, “T-hsab: A tunisian hate speech and abusive dataset,” in *International Conference on Arabic Language Processing*. Springer, 2019, pp. 251–263.

[44] H. Mulki, H. Haddad, C. B. Ali, and H. Alshabani, “L-hsab: A levantine twitter dataset for hate speech and abusive language,” in *Proceedings of the third workshop on abusive language online*, 2019, pp. 111–118.

[45] A. Abuzayed and T. Elsayed, “Quick and simple approach for detecting hate speech in arabic tweets,” in *Proceedings of the 4th workshop on open-source Arabic Corpora and processing tools, with a shared task on offensive language detection*, 2020, pp. 109–114.

[46] A. Keleg, S. R. El-Beltagy, and M. Khalil, “Asu_opto at osact4-offensive language detection for arabic text,” in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 66–70.

[47] H. Mubarak and K. Darwish, “Arabic offensive language classification on twitter,” in *International Conference on Social Informatics*. Springer, 2019, pp. 269–276.

[48] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, “Detecting offensive language on arabic social media using deep learning,” in *2019 sixth international conference on social networks analysis, management and security (SNAMS)*. IEEE, 2019, pp. 466–471.

[49] N. Albadi, M. Kurdi, and S. Mishra, “Investigating the effect of combining gru neural networks with handcrafted features for religious hatred detection on arabic twitter space,” *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–19, 2019.

[50] A. Elmadany, C. Zhang, M. Abdul-Mageed, and A. Hashemi, “Leveraging affective bidirectional transformers for offensive language detection,” in *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, p. 102.

[51] M. Djandji, F. Baly, W. Antoun, and H. Hajj, “Multi-task learning using arabert for offensive language detection,” in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2020, pp. 97–101.

[52] M. J. Althobaiti, “Creation of annotated country-level dialectal Arabic resources: An unsupervised approach,” *Natural Language Engineering*, pp. 1–42, 2021, DOI: 10.1017/s135132492100019x.

[53] ——, “Automatic arabic dialect identification systems for written texts: a survey,” *arXiv preprint arXiv:2009.12622*, 2020.

[54] M. Althobaiti, U. Kruschwitz, and M. Poesio, “AraNLP: a Java-based Library for the Processing of Arabic Text,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, Association for Computational Linguistics. Reykjavik, Iceland: European Language Resources Association (ELRA), 2014, pp. 4134–4138.

[55] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, “Arabic offensive language on twitter: Analysis and experiments,” in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2021, pp. 126–135.

[56] N. Habash, A. Soudi, and T. Buckwalter, “On arabic transliteration,” in *Arabic computational morphology*. Springer, 2007, pp. 15–22.

[57] S. A. Chowdhury, H. Mubarak, A. Abdelali, S.-g. Jung, B. J. Jansen, and J. Salminen, “A multi-platform arabic news comment dataset for offensive language detection,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 6203–6212.

[58] B. Solomon, “demoji 1.1.0,” 2021, [Online]. Available: https://pypi.org/project/demoji/ (accessed 20-April-2022).

[59] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[60] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, and M. N. Al-Kabi, “A comprehensive survey of arabic sentiment analysis,” *Information processing & management*, vol. 56, no. 2, pp. 320–342, 2019.

[61] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, “The interplay of variant, size, and task type in Arabic pre-trained language models,” in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. Kyiv, Ukraine (Online): Association for Computational Linguistics, Apr. 2021.

[62] M. Nabil, M. Aly, and A. Atiya, “Astd: Arabic sentiment tweets dataset,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2515–2519.

[63] A. Elmadany, H. Mubarak, and W. Magdy, “Arsas: An arabic speech-act and sentiment corpus of tweets,” *OSACT*, vol. 3, p. 20, 2018.

[64] S. Rosenthal, N. Farra, and P. Nakov, “Semeval-2017 task 4: Sentiment analysis in twitter,” in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 502–518.

[65] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[67] W. Antoun, F. Baly, and 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*, 2020, pp. 9–15.