IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 31, 2019, accepted June 18, 2019, date of publication June 21, 2019, date of current version July 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924314

Surface and Deep Features Ensemble for Sentiment Analysis of Arabic Tweets

NORA AL-TWAIRESH AND HADEEL AL-NEGHEIMISH
College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Nora Al-Twairesh (twairesh@ksu.edu.sa)

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through the Research Project No R17-03-69.

ABSTRACT Sentiment analysis (SA) of Arabic tweets is a complex task due to the rich morphology of the Arabic language and the informal nature of language on Twitter. Previous research on the SA of tweets mainly focused on manually extracting features from the text. Recently, neural word embeddings have been utilized as less labor-intensive representations than manual feature engineering. Most of these word-embeddings model the syntactic information of words while ignoring the sentiment context. In this paper, we propose to learn sentiment-specific word embeddings from Arabic tweets and use them in the Arabic Twitter sentiment classification. Moreover, we propose a feature ensemble model of surface and deep features. The surface features are manually extracted features, and the deep features are generic word embeddings and sentiment-specific word embeddings. The extensive experiments are performed to test the effectiveness of the surface and deep features ensemble, pooling functions, embeddings size, and cross-dataset models. The recent language representation model BERT is also evaluated on the task of SA of Arabic tweets. The models are evaluated on three different datasets of Arabic tweets, and they outperform the previous results on all these datasets with a significant increase in the F-score. The experimental results demonstrate that: 1) the highest performing model is the ensemble of surface and deep features and 2) the approach achieves the state-of-the-art results on several benchmarking datasets.

INDEX TERMS Arabic sentiment analysis, arabic sentiment embeddings, arabic tweets, surface and deep features ensemble.

I. INTRODUCTION

The abundance of user-generated content in the form of social media websites has produced massive amounts of unstructured text on the web. This text contains sentiment and opinions that are valuable for both individuals and organizations. Sentiment analysis (SA) is ''the field of study that analyzes people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text'' [1]. Hence, given a unit of text, the task of sentiment analysis is to classify the text as positive, negative, or neutral.

Sentiment analysis of Arabic tweets is a complex task due to the rich morphology of the Arabic language and the informal nature of language on Twitter. Approaches to sentiment analysis include supervised learning techniques that exploit machine learning algorithms with feature engineering and unsupervised learning techniques that exploit sentiment lexicons and rule-based methods. The dominant methods that use machine learning algorithms rely on the manual extraction of features to be used in the classification. However, the manual extraction of features is time-consuming and labor-intensive. These manually extracted features are known to be surface features [2].

Distributed word representations learned through neural network models (word embeddings) have emerged as promising models for natural language processing tasks; moreover, they have resulted in numerous state-of-the-art results in the field. Different approaches have been proposed to learn such representations, e.g., the C&W model [3], the word2vec model [4], the Glove model [5], and the most recent fastText model [6]. The significance of these word embeddings is that they can automatically learn features required for classification without manual intervention, as in the hand-crafted feature engineering approaches. These generic word

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

embeddings, also known as pre-trained word vectors, require large amounts of data to perform effectively [4]. We refer to these types of embeddings as generic embeddings throughout this paper.

Generic word embeddings model the semantic and syntactic representation of a word, while the sentiment of a word cannot be captured. Actually, using generic word embeddings can result in words of opposite sentiment to have similar vector representations; this is because these words appear in similar syntactic contexts, e.g., ''This book is good.'' and ''This book is bad.'' [7]. Therefore, in this paper, we propose to learn sentiment-specific word embeddings from massive distant supervised Arabic tweets collected by positive and negative keywords. Then, we use these sentiment embeddings as features in the sentiment classification of Arabic tweets. We compare the performance of the sentiment embeddings with generic word embeddings constructed using the word2vec methods [4]. The Arabic sentiment embeddings are constructed using the models presented in [7]. The sentiment embeddings are also compared to manually extracted features or surface features. Then, following the work in [2] we experiment with ensembles of surface features, generic embeddings, and sentiment embeddings: ensembles of pairs and an ensemble of all the three types of features.

In a very recent work on language representation models, [8] introduced BERT which stands for Bidirectional Encoder Representations from Transformers. BERT is pre-trained by conditioning on both left and right context in all layers unlike previous language representation models. To apply BERT to any NLP task, one only needs to fine-tune one additional output layer to the downstream task; this makes it different from previous word embeddings used in this paper which are applied to the task of SA as features. As this type of language representation model is new, we experimented with it to evaluate its performance on the task of Arabic SA. To the best of our knowledge the exploitation of BERT for the Arabic language has not been used before in any NLP task. We chose BERT because it advances the state-of-the-art for eleven NLP tasks including sentiment classification [8].

The peculiar nature of the Arabic language (with different forms of Arabic: the formal Modern Standard Arabic (MSA) and the informal dialects) affect the performance of text classification tasks [9]; therefore, we decided to evaluate the proposed models on three Arabic tweet datasets.

The research objective of this paper is to explore the impact of generic embeddings and sentiment-specific embeddings on SA of Arabic tweets and compare between them and traditional hand-crafted features and the ensemble of these three types of features. Arabic is a low-resourced language, in this paper we use previously proposed methods in the literature for the English language [2], [7] and apply them on Arabic datasets.

The contributions of this paper are as follows:

1. Learning sentiment-specific word embeddings from Arabic tweets and making these embeddings publicly available for the research community.
2. Comparing the performances of sentiment embeddings, generic embeddings, and manually extracted features in the task of sentiment analysis of Arabic tweets.
3. An ensemble of surface and deep features for Arabic Sentiment Analysis is proposed and evaluated.
4. A recent language representation model (BERT) was evaluated and compared to traditional embeddings.
5. The proposed models are evaluated on three datasets of Arabic tweets.
6. Cross data evaluation is performed on the three datasets and on the aggregation of the datasets.

This paper is organized as follows. Section II reviews the related work. In Section III, details of the models used to construct the Arabic sentiment-specific embeddings are described. In Section IV, the details of the experiments and results are presented. Section V concludes the paper.

II. RELATED WORK

A. TWITTER SENTIMENT CLASSIFICATION

Research on SA of English text started in 2002 with the publication of two studies: whereas [10] presented a supervised learning corpus-based machine classifier, [11] presented an unsupervised classifier based on linguistic analysis. Previously, the focus was mostly on product and movie reviews; it expanded to other domains with the emergence of social media websites. Several studies followed, such as the Opinion Finder tool [12] and SO-CAL [13]. Recent systematic reviews on sentiment analysis are available in [14], [15].

SemEval is an annual series of semantic evaluation tasks conducted to foster competition in several tasks related to semantic analysis systems. Since SemEval 2013 [16], a task has been dedicated to sentiment analysis of Twitter. This task endorses research in SA of short informal text and provides a benchmark for comparing different approaches. Training and test sets are released and several research teams compete to attain the highest classification accuracy feasible. The approaches to twitter sentiment classification were mainly focused on machine learning with feature engineering, such as the winning systems in SemEval 2013 and 2014 Task 4 (Twitter Sentiment Analysis) [17], [18]. However, from 2015 to 2017, the top ranking systems utilized word embeddings to overcome the manual intensive feature engineering approaches. Reference [19] ranked first in the SemEval 2015 SA of Twitter task on the phrase level and second on the message level [20]. Their system consisted of a deep convolutional neural network (CNN) that was pre-trained on a large dataset of 50 million tweets for training word embeddings and a 10-million-tweet corpus for distant supervision. The reported results were 84.49 for phrase-level subtask and 64.59 for message level subtask. However, [21] ranked first in the message-level in SemEval 2016; it was constructed upon the work of [19] by extending their one-layer architecture to a two-layer CNN, and then combining the predictions using a random forest meta-classifier.

They scored 67.05 on the 2015 test set and 63.30 on the 2016 test set.

In SemEval 2017 [22], two systems scored first in the task for English tweets: both [23] and [24] used pre-trained word embeddings, with an ensemble of CNN and Long Short Term Memory (LSTM) in [23] and LSTM with attention in [24]. Each of them scored 68.1. SemEval 2017 also had a subtask for Arabic Tweets, and the highest ranking system [25] earned an F-score of 61 by using a Naive Bayes classifier with a combination of lexical and sentiment features that were manually engineered.

Beyond SemEval tasks, numerous efforts on sentiment classification of tweets using deep learning approaches have been proposed, such as [2], [7], [26]–[33]. Tang *et al.* [7], [27] introduced three neural network models for learning sentiment specific word embeddings by extending the previous word embedding model (C&W) of [3] to incorporate the sentiment information. The C&W model [3] consists of four layers: lookup → linear → hTanh → linear. This representation does not permit the model to capture sentiment information. Therefore, [7] introduced a new layer on top of the last layer while modifying the dimension of the last layer (linear) to K, which is the number of classes (two in this case for positive and negative). The new layer is a softmax layer because it is suitable for predicting the two classes: positive and negative. Hence, this model was called the prediction model. The second model was the ranking model. In this model, the softmax layer in the prediction model is replaced with a ranking loss function, whereas the other four layers of the prediction model are identical in the ranking model. Thus, it produces two real valued sentiment scores as the output. The third model proposed by [7] combines the C&W model with the prediction and ranking models; it is aimed at capturing both the sentiment information and the syntactic context of words in a tweet. The model predicts a two-dimensional vector for each word. The dimensions represent the language model score and sentiment score. This is called the hybrid model. We use these three models to extract sentiment-specific word embeddings from Arabic tweets, as described in Section III.

In [33], the authors proposed a target dependent sentiment analysis system on Twitter. They used automatic features extracted from sentiment embeddings as in [7], generic word embeddings, and sentiment lexicons. They also experimented with different neural pooling functions. Their method achieved an improvement over the state-of-the-art, in three-way targeted sentiment classification.

Building upon the work of [7], the authors of [34] argued that tweet level sentiment-specific embeddings alone are not enough to train a neural network and proposed to add word-level sentiment. In contrast to [7], they build the sentiment specific word embeddings from both a massive dataset of sentiment bearing tweets (collected through hashtags of positive and negative words and through emoticons) and a sentiment lexicon. Hence, they developed a multi-level sentiment-enriched word embedding. They evaluated the proposed model on the SemEval2013 dataset and surpassed the results of [7] on the same dataset by 0.77%.

Moreover, in contrast to [7], [35] proposed a different technique to learn sentiment embeddings, without the need for labeled corpora. The proposed model uses a sentiment lexicon that comprises sentiment intensity scores to refine pre-trained word embeddings by improving each word vector to be closer to semantically and sentimentally similar words where the objective function was based on maximizing the Euclidean distance. The model was applied to conventional word embeddings (word2vec and Glove) and to the sentiment specific embeddings of [7]. The model was evaluated on SemEval 2017 [22] and the results showed that the refinement of the Glove embeddings scored the highest among all embeddings.

In [2], Araque et. al. present a pioneering advancement in the field of deep learning for SA by proposing sentiment models that combine several sentiment classifiers to produce an ensemble of classifiers and an ensemble of surface and deep features. The deep features used are the sentiment embeddings from [7] and generic word embeddings. The models were evaluated on six public datasets from two domains: Twitter and movie reviews. The highest performing models for the Tweets datasets were those that combined the surface features with the generic word embeddings. We follow their approach in the ensemble of features.

In [36], word embeddings were constructed using Glove [5] on a large Twitter corpus, then combined with n-gram features and sentiment intensity scores to be fed into a deep CNN. The model was evaluated on five Twitter data sets from the literature and gave good results. However, the authors did not mention how the Twitter corpus was constructed or collected and if the tweets contained sentiment or not. Also, no experiments were provided on the combination of the Glove word embeddings with the manually engineered features and the comparisons with previous work was not comprehensive of all work that used the same datasets.

Reference [37] proposed to combine sentiment information from the training data and a sentiment lexicon then the sentiment information is encoded into word embeddings using a feed-forward neural network which is combined with a CNN. Thus, the word embeddings fine-tuning with sentiment information is done at the same time the CNN is training. Several experiments were conducted for training variations of the feed-forward network and the CNN. The models were evaluated on several Twitter datasets and gave comparable results to the state of the art on these benchmarks.

With regard to the sentiment analysis of Arabic tweets, the approaches have also evolved from feature engineering to the utilization of neural word embeddings. In [38], a CNN was used in conjunction with the Arabic word embeddings constructed by [39] without any hand-crafted features, for sentiment analysis of Arabic tweets. The model was evaluated on two publicly available Arabic datasets:

SemEval2017 [22] and Arabic Sentiment Tweets Dataset (ASTD) [40]. The highest F1-score achieved were 63 for the SemEval 2017 dataset and 72.14 for ASTD.

In [9], Baly et al. developed a multi-dialect Arabic twitter sentiment dataset that contains tweets from 12 Arabic countries. They compared different sentiment models on the Egyptian and Emarati datasets. The sentiment models were developed using both feature engineering (SVM) and deep learning (LSTM). The results demonstrate the dominance of deep learning. Moreover, they revealed the importance of distinguishing between the Arabic dialects when constructing sentiment analysis classifiers for Arabic tweets.

Pioneering work in the utilization of deep learning models for Arabic sentiment analysis was presented in [41]. Their model incorporated both semantic and sentiment embeddings to train a Recursive Auto Encoder (RAE). However, the sentiment embeddings are learned through an Arabic sentiment lexicon that is written in MSA [42]. They evaluated the model on Arabic texts of different genres including tweets; the F1-score on tweets was 68.9.

In [43], the authors investigated applying SMOTE (Synthetic Minority Over-sampling Technique) with word embeddings on a dataset of Arabic tweets in the Syrian dialect using an ensemble of different machine learning classifiers (k-NN, SVM, Logistic regressions, Stochastic Gradient Descent (SGD), Gaussian Naïve Bayes. and Decision Trees) the best F1-score reached was 63.95.

In [44], generic Arabic word embeddings were constructed using a large corpus (190 M words) collected from news domain and customer reviews and the word2vec model. Then they used these word embeddings as features to train a sentiment classifier by experimenting with six different ML classifiers. They evaluated their classifiers on a combined dataset of Arabic Tweets from ASTD [40] ArTwitter [45] QRCI [46]. The best performance was with the SVM classifier where the F-score was 79.62. A combined LSTM and CNN was presented in [47] and evaluated on the ASTD dataset, the accuracy was 77.62.

For more work on Arabic sentiment analysis, the authors in [48] survey the recent work on Arabic Sentiment Analysis with regards to the considered data/scope, the employed approach and the utilized resources. Moreover, the authors in [49] survey papers that present deep learning techniques for Arabic NLP including sentiment analysis. While the most recent survey in [50], presents a comprehensive survey on Arabic SA including deep learning advances in Arabic SA

B. GENERIC ARABIC WORD EMBEDDINGS

Several Arabic word embeddings have been proposed in the literature. In [39], the first attempt to construct Arabic word embeddings was initiated. The embeddings were evaluated on Arabic word analogy datasets and performed very well. In [51], Arabic word embeddings were constructed using word2vec [4] from a large corpus of Arabic text crawled from the web. These pre-trained word embeddings were then used to train a CNN for sentiment classification. The model was evaluated on Arabic datasets from different genres including the ASTD dataset [40]. The accuracy reported on ASTD was 79.07.

In [52], the authors present AraVec, which is a large-scale pre-trained Arabic word embedding. It provides six different word embedding models learned from three different Arabic genres, namely, World Wide Web pages, Wikipedia Arabic articles, and Tweets, using the word2vec model. For each genre, a Continuous Bag Of Words (CBOW) model and a Skipgram model were provided. The size of the dataset of tweets used was 77 billion tweets. To the best of our knowledge, this is the largest word embedding constructed from Arabic tweets. Therefore, we utilize these embeddings in our work.

III. ARABIC SENTIMENT-SPECIFIC WORD EMBEDDINGS

Tang *et al.* [7] introduced three neural network models to learn sentiment specific word embeddings, by extending the previous word embedding model (C&W) of [3] to incorporate the sentiment information. In line with [7], we use distant supervision to collect the training data. The authors in [7] collected 10 million tweets: 5 million with positive emoticons and 5 million with negative emoticons. However, we use positive and negative keywords, rather than emoticons, to collect the Arabic tweets; this is because of the statement in [53] that Arabic tweets containing emoticons generally do not convey sentiment and are of a chatting nature. They arrived at this conclusion after inspecting a large dataset of 2.2 million Arabic tweets.

We use a list of Arabic positive and negative words to collect 10 million Arabic tweets from the Twitter API during June and July 2017. Prior to classifying the sentiment of tweets, it is important to preprocess the text such that specific letters are normalized and non-Arabic letters are removed. In particular, to reduce noise and sparsity in Arabic text orthographic normalization of certain Arabic letters is performed. Orthographic normalization is the process of unifying the shape of some Arabic letters that have different shapes[43]. The Arabic letters (ى, ي, ة, أ) are normalized to convert multiple shapes of the letter to one shape as follows:

- "إ", "آ", "أ", and "ا" are replaced by "ا";
- "ة" is replaced by "ه";
- "ى" is replaced by "ي";
- and finally, "ؤ" and "ئ" are replaced by "ء".

Punctuation is replaced with a single space, which helps tokenizing words separated by punctuation only (this includes special characters, emojis, or Latin letters). Finally, additional whitespaces are removed, and all non-Arabic letters are omitted. This results in a tweet containing only Arabic words. These tweets are used to construct the Arabic sentiment-specific word embeddings by training the four models (C&W, Prediction, Ranking, and Hybrid) of [7]. Following [7], we set the window size as seven, length of the hidden layer as 20, and learning rate of AdaGrad as 0.1. With regard to the embedding length, we experiment with two embedding sizes, as detailed below in Section IV B.

IV. TWITTER SENTIMENT CLASSIFICATION

A. DATASETS

Given the peculiar nature of the Arabic language, where the different forms of Arabic (MSA and dialects) affect the performance of all text classification tasks [9], we evaluate the proposed models on three datasets of Arabic tweets. The first dataset is the SemEval 2017 Arabic tweet dataset [22]. It contains tweets written in different Arabic dialects and is considered a benchmark for Arabic tweet sentiment classification. It enables us to compare with the highest performing system in the SemEval 2017 Sentiment Analysis in Twitter task for Arabic. The second dataset is the AraSenTi-Tweet dataset [54]; it contains tweets written in the Saudi dialect. The third dataset is the ASTD dataset [40]; it contains tweets written in the Egyptian dialect. For each dataset, we include the tweets that were classified as positive or negative because we are performing two-way sentiment classification. The statistics of these datasets are presented in TABLE 1.

TABLE 1. Statistics of datasets.

Dataset	Positive	Negative	Total
SemEval2017-Train	965	1,270	**2,235**
SemEval2017-Test	1,514	2,222	**3,736**
AraSenTi-Train	4,235	5,515	**9,750**
AraSenTi-Test	722	640	**1,362**
ASTD-Train	647	1,432	**2,079**
ASTD-Test	150	250	**400**

B. EXPERIMENTAL SETTINGS

We perform two-way sentiment classification (positive–negative). To evaluate the models, we apply them in a supervised learning framework for sentiment classification. We construct the sentiment classifier by using SVM with LibLinear [55], as implemented in Python Scikit learn [56]. We set the dual parameter to false and use default values for the other parameters; namely, l2 norm penalty, square hinge loss function, and C = 1.0. The words that compose the tweet are each converted into a word vector. Then, the computed word vectors are combined into one vector that represents the whole tweet by using a set of pooling functions.

The surface features or manually extracted features used are those found in [57]. The generic word embeddings used is the (AraVec) constructed in [52]; it is mentioned in Section 0II B. The sentiment embeddings are extracted using the models of [7], as explained in Section III. Experiments for the generic word embeddings and the sentiment-specific word embeddings were performed on Amazon Web Services (AWS)[1] with different configurations of GPUs and memory for each experiment. As for the BERT model we used Google TPU cloud.[2] The code for all experiments can be found here.[3]

The following are the different features and embeddings used in the models:

[1]https://aws.amazon.com/
[2]https://cloud.google.com/tpu/
[3]https://github.com/halnegheimish/DeepAraSenti

Basline: no features or embeddings are used.

Surface Features (SF): the work in [57] utilized manually extracted features including syntactic and semantic features (Arabic sentiment lexicons). We use the same features, namely, the tweet-length, emoticons, count of positive and negative words, intensifiers, tweet score using the intensity values of the Arabic sentiment lexicon in [53], and Arabic translations of the English lexicons Liu [58] and MPQA [12].

C&W: the Arabic word embeddings built from the collected dataset presented in Section III using the method of [3]. Although these embeddings are constructed from tweets containing sentiment, they do not contain sentiment information.

ASEP: the Arabic Sentiment Embeddings constructed using the Prediction model of [7].

ASER: the Arabic Sentiment Embeddings constructed using the Ranking model of [7].

ASEH: the Arabic Sentiment Embeddings constructed using the Hybrid model of [7].

AraVec: we utilize the Arabic word embeddings learned from a large dataset of Arabic tweets (77 billion) using word2vec in [52]. We use the embeddings constructed using CBOW with dimension 300.

BERT: BERT has a pre-trained model for the Arabic language that was learned from Wikepedia [8].

1) EVALUATION METRICS

All the results are reported using the macro averaged F1-score of the two classes: positive and negative. The F-score (F1), Precision (P), and Recall (R) of the positive and negative classes are as follows:

$$F1 = 2 \times (P \times R)/(P + R)$$
$$P = tp/(tp + fp)$$
$$R = tp/(tp + fn)$$

where tp is the number of positive tweets classified correctly as positive (true positive), fp is the number of negative tweets falsely classified as positive (false positive), fn is the number of positive tweets falsely classified as negative (false negatives), and tn is the number of negative tweets correctly classified as negative (true negatives).

C. EXPERIMENTS

We conduct a set of experiments to evaluate the effectiveness of the pooling functions, sentiment embeddings model and size, features and embeddings ensemble, and data aggregation and cross data evaluation, on the performance of the models. Then, using the most effective development settings, a final test is performed to evaluate the model.

1) EFFECTIVENESS OF POOLING FUNCTIONS

As in [7] and [2], we use the max, average, and min pooling functions to compose the tweet representation. Although [7] used only the concatenation of these pooling functions, we experiment with concatenation and with each individual pooling function. For each of the above embeddings (C&W, ASEP, ASER, ASEH, and AraVec), we experiment with the embeddings using concatenation of the max, avg, and min

TABLE 2. F1-score for different pooling functions.

Method	SemEval				AraSenTi				ASTD			
	min	max	avg	concat	min	max	avg	concat	min	max	avg	concat
C&W	70.46	71.19	77.93	75.81	63.10	52.79	65.45	66.46	69.26	71.32	77.05	78.72
ASEP	72.88	74.30	78.34	76.12	52.05	53.53	67.60	63.53	72.82	74.17	75.48	74.16
ASER	73.76	72.71	77.02	75.01	55.95	50.00	62.84	56.68	68.43	71.55	73.29	76.90
ASEH	74.36	76.75	79.42	77.96	58.39	59.39	69.03	65.48	73.70	77.27	75.95	77.55
AraVec	71.78	74.14	**80.28**	74.03	65.95	58.19	**70.95**	64.86	76.13	71.73	**79.08**	70.68

pooling functions and each individual function. The highest performance was obtained with the avg function in most of the models. Hence, it is used in the following experiments. This result is also consistent with the previous related work in [2]. TABLE 2 presents the macro F1-score for all the pooling functions for each embedding model for each dataset.

TABLE 3. F1-score for different embedding sizes (50 and 300).

Method	SemEval		AraSenTi		ASTD	
	50	300	50	300	50	300
C&W	77.93	72.47	65.45	61.40	75.23	69.61
ASEP	78.34	79.35	67.60	71.25	75.48	74.85
ASER	77.02	77.35	62.84	59.18	73.29	79.07
ASEH	**79.42**	79.00	**69.03**	66.45	75.95	78.36

2) EFFECTIVENESS OF SENTIMENT EMBEDDINGS MODEL AND SIZE

The sentiment embeddings dimension in [7] was set to 50, whereas the dimension of the AraVec embeddings we use in this paper is 300. Accordingly, we empirically test sentiment embeddings with dimensions 50 and 300. Given that a larger dimension infers more information gain, the larger dimension could be expected to perform better. However, this is not the case, as illustrated in TABLE 3. We obtain the highest performance when the sentiment embeddings' size is 50. Moreover, comparing the different sentiment embeddings models (C&W, ASEP, ASER, and ASEH), we observe that the highest performance is the hybrid model ASEH. Hence, in the following experiments and final model, the sentiment embeddings used is the hybrid model ASEH with the embeddings size set to 50.

3) EFFECTIVENESS OF FEATURE EMBEDDINGS ENSEMBLE

In this section, we present the essential set of experiments in this paper. We evaluate the impact of each individual type of features and embeddings, alone and then through ensemble of pairs, then ensemble of all the three. We evaluate the surface features (SF) using the features in [57], generic embeddings (GE) using AraVec [52], and sentiment embeddings using the hybrid model ASEH, individually in each case. Then, we use ensemble of surface and deep features: SF + GE and SF + ASEH; the features that were entirely extracted using deep learning techniques: GE+ASEH; and all the three types of features: SF + GE + ASEH. FIGURE depicts the ensemble of the features. The results are presented

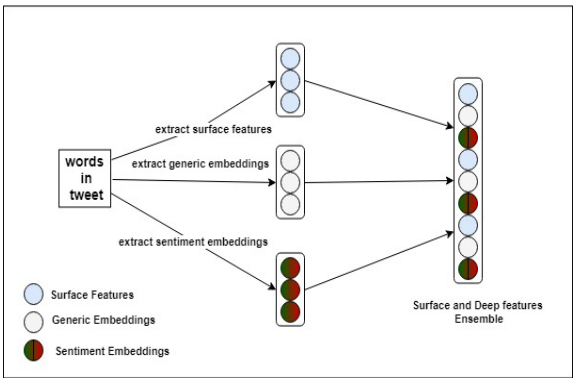

FIGURE 1. Surface and deep features ensemble where the sentiment embeddings here are the ASEH using the hybrid model, the ensemble may be (SF + GE; SF + ASEH; SF+GE+ASEH).

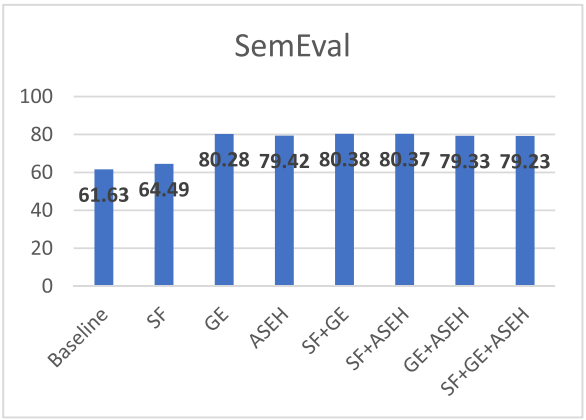

FIGURE 2. F1-score for all models on SemEval dataset.

in TABLE IV. FIGURE 2 illustrates the F1-score for all the models on the SemEval dataset, FIGURE 3 illustrates the F1-score for all the models on the AraSenTi dataset, FIGURE 4 illustrates the F1-score for all the models on the ASTD dataset.

4) EFFECTIVENESS OF BERT MODEL

As mentioned before, the field of NLP has recently witnessed rapid advances in language representation models. In attempt to evaluate one of these recent models we decided to perform an experiment on the effectiveness of the latest language representation model in the field which is the BERT model [8]. There are two existing strategies for applying pre-trained language representations to downstream tasks: feature-based and fine-tuning [8]. For embeddings mentioned in previous

TABLE 4. F1-Score of all methods. SF: Surface features, GE: Generic word embeddings, ASEH: Sentiment embeddings.

Method	Positive			Negative			Average		
	P	R	F1	P	R	F1	P	R	F1
SemEval									
Baseline	61.42	41.02	49.19	67.23	82.45	74.07	61.42	61.73	61.63
SF	59.18	54.10	56.52	70.45	74.57	72.45	59.18	64.33	64.49
GE	77.71	74.83	76.24	83.27	85.37	84.31	77.71	80.10	80.28
ASEH	76.59	73.91	75.23	82.64	84.61	83.61	76.59	79.26	79.42
SF+GE	77.68	75.17	76.40	83.44	85.28	84.35	77.68	80.22	**80.38**
SF+ASEH	77.48	75.43	76.44	83.55	85.06	84.30	77.48	80.24	80.37
GE+ASEH	76.24	74.17	75.19	82.72	84.25	83.48	79.48	79.21	79.34
SF+GE+ASEH	76.30	73.78	75.02	82.53	84.38	83.44	79.41	79.08	79.23
BERT	-	-	-	-	-	-	71.3	62.03	66.27
AraSenTi									
Baseline	66.89	55.40	60.61	57.85	69.06	62.96	66.89	62.23	61.78
SF	73.61	67.82	70.60	66.57	72.50	69.41	73.61	70.16	70.00
GE	78.04	63.02	69.73	65.73	80.00	72.16	78.04	71.51	70.95
ASEH	70.74	71.33	71.03	67.35	66.72	67.03	70.74	69.02	69.03
SF+GE	75.85	77.42	76.63	73.92	72.19	73.04	75.85	74.81	**74.84**
SF+ASEH	66.86	81.58	73.49	72.35	54.38	62.09	66.86	67.98	67.79
GE+ASEH	71.64	67.27	69.38	65.55	70.05	67.72	68.59	68.66	68.55
SF+GE+ASEH	73.20	74.62	73.90	70.81	69.27	70.03	72.01	71.94	71.97
BERT	-	-	-	-	-	-	64.34	78.39	70.88
ASTD									
Baseline	82.08	58.00	67.97	78.57	92.40	84.93	82.08	75.20	76.45
SF	81.58	41.33	54.87	72.84	94.40	82.23	81.58	67.87	68.55
GE	83.33	63.33	71.97	80.77	92.40	86.19	83.33	77.87	79.08
ASEH	87.91	53.33	66.39	77.35	95.60	85.51	87.91	74.47	75.95
SF+GE	81.36	64.00	71.64	80.85	91.20	85.71	81.36	77.60	78.68
SF+ASEH	90.00	54.00	67.50	77.74	96.40	86.07	90.00	75.20	76.79
GE+ASEH	80.80	67.33	73.45	82.18	90.40	86.10	81.49	78.87	**79.77**
SF+GE+ASEH	78.69	64.00	70.59	80.58	89.60	84.85	79.63	76.80	77.72
BERT	-	-	-	-	-	-	80.12	52.14	69.59

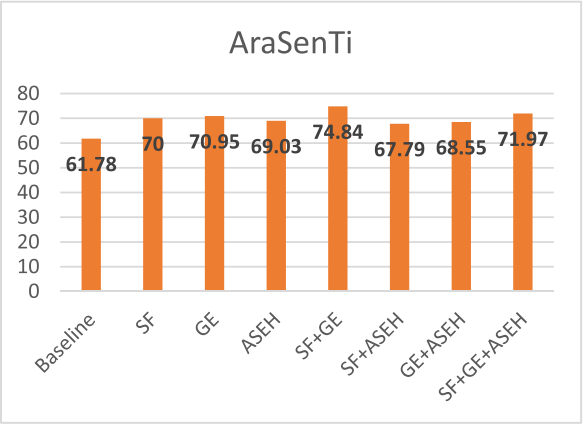

FIGURE 3. F1-score for all models on AraSenTi dataset.

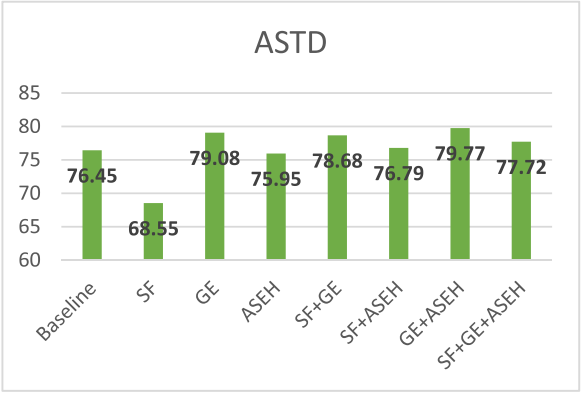

FIGURE 4. F1-score for all models on ASTD dataset.

sections, they are applied as features to the task of sentiment classification. As for BERT, this pre-trained model was fine-tuned by adding a simple classification layer that performs the task of sentiment classification to the pre-trained model. As mentioned in [8], most model hyperparameters are the same as in pre-training, with the exception of the batch size, learning rate, and number of training epochs which are task-specific. Therefore, we performed an exhaustive search on

TABLE 5. Data aggregation of datasets and cross-dataset evaluation.

Training	All	AraSenti		SemEval		ASTD	
Testing	All	SemEval	ASTD	AraSenti	ASTD	AraSenti	SemEval
SF	64.11	64.67	67.71	50.66	59.33	53.82	46.81
GE	78.69	**68.67**	**76.18**	73.79	78.83	70.05	**73.90**
SF+GE	**80.47**	57.16	69.76	72.83	77.65	**71.51**	72.97
ASEH	75.58	72.20	75.67	67.71	**80.83**	69.38	72.65
SF+ASEH	77.77	62.09	67.92	67.59	74.40	71.13	70.29
GE+ASEH	78.76	67.33	72.46	**74.87**	77.02	70.26	72.80
SF+GE+ASEH	80.08	60.86	71.56	73.27	73.19	70.40	71.35

these parameters according to the suggested ranges in [8]: batch size = 16, 32, learning rate (Adam) = 5e − 5, 3e-5, 2e-5, and epochs = 3,4. We found the optimal values to be: batch size = 32, learning rate = 2e-5, and epochs = 3. For each dataset, we performed 10 runs and reported the average of the F1-score for each run. The results are in the last row of TABLE IV.

5) EFFECTIVENESS OF DATASET AGGREGATION AND CROSS-DATASET EVALUATION

In this experiment, we evaluate the impact of training on a certain dataset while testing on a different dataset. We use the highest performing model, i.e., the average pooling function; moreover, the dimension of sentiment embeddings is 50. First, we aggregate the training sets of all the three datasets and test sets of all three datasets and perform the experiment. The results are in the first column of TABLE V. Then, we train on a dataset and test on the remaining two, as illustrated in TABLE V. We observe that aggregating the datasets into one dataset with the ensemble of surface features and generic features significantly increases the performance; it surpasses the highest performance among all the experiments on each individual dataset. Moreover, when training on a dataset and testing on another, we observe that the best performing model is in coherence with the best performing model when training and testing on the same dataset. For the SemEval dataset, when we train on AraSenTi or ASTD, the highest performance is obtained when the generic embeddings are used. For the AraSenTi dataset, the highest performance is obtained when using the deep features and when using the ensemble of surface features and generic embeddings. For the ASTD dataset, the highest performance is when the sentiment-specific embeddings are used while the training was on SemEval dataset.

D. RESULTS AND DISCUSSION

TABLE IV presents the results of the highest performance models according to the experiments mentioned above. We observe that the sentiment-specific embeddings outperform the baseline and surface features for the SemEval and ASTD datasets. However, the generic embeddings that are constructed using a large dataset of Arabic tweets outperform the sentiment-specific embeddings. This could be because the dataset of 10 million tweets that was collected is not large enough to capture all the sentiment representations in Arabic tweets. It is well known that the Arabic language's morphology causes data sparsity; this is because an Arabic lemma can have hundreds of surface forms. However, the sentiment embeddings constructed through the hybrid model ASEH are highly similar to the generic embeddings model on the SemEval dataset. Nonetheless, the sentiment embeddings and generic word embeddings outperform previous work on all the three datasets. The most significant improvement is on the SemEval dataset. Whereas the highest score reported in the literature on this dataset is 63 (in [38]), while our model scores 80.38 using the ensemble of surface features and generic embeddings, i.e., it outperforms the previous work by +14%. As for the AraSenti dataset, we can see from TABLE IV and FIGURE 3 that the best F1-score was also through the ensemble of surface features and generic embeddings as in SemEval dataset. It outperforms previous work on the same dataset [57] by +5%. The ASTD dataset performed best with the ensemble of deep features alone as demonstrated in TABLE IV and FIGURE 4. This could be because the surface features we used in this paper were tailored to the Saudi Dialect [57] while the ASTD dataset contains tweets written in the Egyptian Dialect. This also confirms the superiority of deep learning methods over manual feature engineering for languages that have different forms such as Arabic. As for the BERT model, we notice that its performance was worse than the generic embeddings and the sentiment specific embeddings but slightly better than the surface features. This proves that these new language representation models are better than hand-crafted features. We attribute the superiority of the generic and sentiment-specific embeddings on BERT to the data they were trained on as the generic and sentiment-specific embeddings were trained on tweets while BERT was trained on Wikipedia pages that are written in MSA. As we mentioned before that the peculiar nature of the Arabic language affects the

performance of text classification tasks, and consequently the genre of the data that these models are trained on.

Moreover, the final experiment on data aggregation reveals that, although the three datasets contain tweets written in different dialects, their aggregation into one dataset enhances performance. Hence, the highest F-score in all the models proposed in this paper is on the aggregation of the datasets using the ensemble of surface features and generic embeddings. We argue that although previous work on SA of Arabic tweets claimed that models should be dialect-specific, our experiments proved that the aggregation of datasets from different dialects enhances the performance.

TABLE 6. Comparison between performance of our models and those of related work.

Method	SemEval	AraSenTi	ASTD
Dahou et al. [51]	-	-	79.07
Al-Twairesh et al. [57]	-	69.9	65.8
Gridach et al. [38]	63	-	72.14
Our models	**80.38**	**74.84**	**79.77**

In TABLE VI, a comparison between the performance of our models and related work on the same datasets is presented. We observe the superiority of our proposed models compared to the related work. Our results are in line with [2], where their highest performing model for sentiment analysis of tweets was the classifier with ensemble of surface features and generic embeddings.

V. CONCLUSION

In this paper, we proposed an ensemble of surface and deep features for sentiment classification of Arabic tweets. Different models were explored to incorporate sentiment into word embeddings, which resulted in the construction of sentiment-specific embeddings. An explorative study was performed to evaluate the combination of generic word embeddings, sentiment-specific word embeddings, and manually crafted (surface) features. The models were evaluated on three Arabic Tweets datasets. The conclusion is that generic word embeddings constructed from a massive dataset of Arabic tweets without incorporating sentiment information using the popular word2vec method outperformed the sentiment-specific embeddings. For future work, we suggest collecting a larger dataset of Arabic tweets to construct the sentiment embeddings.

Moreover, the best performing model was the ensemble of surface features and generic word embeddings. This is consistent with previous work on English tweets. Aggregation of the datasets into a single dataset also resulted in enhanced performance by using the same model, i.e., the ensemble of surface features and generic embeddings.

A recent trend in language representation models is to pre-train the model on a language model objective before fine-tuning that same model for a supervised downstream task; this minimizes the number of hyperparameters that need to be learned from scratch. The BERT model is one these recent models, therefore we opted to evaluate it on our task of sentiment classification of Arabic tweets. However, this model did not beat the generic word embeddings or sentiment-specific embeddings or their ensemble. It was slightly better than the surface features. This could be because it was trained on Wikipedia while both types of embeddings were learned from tweets. Therefore, for future work we propose to train BERT on tweets.

REFERENCES

[1] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[2] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, vol. 2013, pp. 1–9.

[5] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2016.

[7] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1555–1565.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: https://arxiv.org/abs/1810.04805

[9] R. Baly, G. El-Khoury, R. Moukalled, R. Aoun, H. Hajj, K. B. Shaban, and W. El-Hajj, "Comparative evaluation of sentiment analysis methods across arabic dialects," *Procedia Comput. Sci.*, vol. 117, pp. 266–273, May 2017.

[10] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process. Volume*, 2002, pp. 79–86.

[11] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2002, pp. 417–424.

[12] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. Hum. Lang. Technol. Conf. Conf. Empirical Methods Natural Lang. Process.*, 2005, pp. 347–354.

[13] M. Taboada, C. Anthony, J. Brooke, J. Grieve, and K. Voll, "SO-CAL: Semantic orientation calculator," Simon Fraser Univ., Vancouver, BC, Canada, 2008.

[14] R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Inf. Process. Manag.*, vol. 53, no. 1, pp. 122–150, Jan. 2017.

[15] M. V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers," *Comput. Sci. Rev.*, vol. 27, pp. 16–32, Feb. 2018.

[16] T. Wilson, Z. Kozareva, P. Nakov, S. Rosenthal, V. Stoyanov, and A. Ritter, "SemEval-2013 task 2: Sentiment analysis in Twitter," in *Proc. 8th Int. Workshop Semantic Eval., (SemEval)*, vol. 13, 2013, pp. 1–18.

[17] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," in *Proc. 7th Int. Workshop Semantic Eval. (SemEval)*, 2013, p. 321.

[18] X. Zhu, S. Kiritchenko, and S. Mohammad, "NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 443–447.

[19] A. Severyn and A. Moschitti, "UNITN: Training deep convolutional neural network for Twitter sentiment classification," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 464–469.

[20] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in Twitter," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 451–463.

[21] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, and M. Jaggi, "SwissCheese at SemEval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 1124–1128.

[22] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 502–518.

[23] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter sentiment analysis with CNNs and LSTMs," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 573–580.

[24] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 747–754.

[25] S. R. El-Beltagy, M. El kalamawy, and A. B. Soliman, "NileTMRG at SemEval-2017 Task 4: Arabic sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 790–795.

[26] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th Int. Conf. Comput. Linguistics (COLING)*, 2014, pp. 69–78.

[27] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, Feb. 2016.

[28] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Context-sensitive Twitter sentiment classification using neural network," in *Proc. AAAI*, Feb. 2016, pp. 215–221.

[29] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Syst. Appl.*, vol. 69, pp. 214–224, Mar. 2017.

[30] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[31] G. Lee, J. Jeong, S. Seo, C. Kim, and P. Kang, "Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network," *Knowl.-Based Syst.*, vol. 152, pp. 70–82, Jul. 2018.

[32] Z. Yuan, S. Wu, F. Wu, J. Liu, and Y. Huang, "Domain attention model for multi-domain sentiment classification," *Knowl.-Based Syst.*, vol. 155, pp. 1–10, Sep. 2018.

[33] D.-T. Vo and Y. Zhang, "Target-dependent twitter sentiment classification with rich automatic features," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 1–7.

[34] S. Xiong, H. Lv, W. Zhao, and D. Ji, "Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings," *Neurocomputing*, vol. 275, pp. 2459–2466, Jan. 2018.

[35] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 671–681, Mar. 2018.

[36] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for Twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[37] Z. Ye, F. Li, and T. Baldwin, "Encoding sentiment information into word vectors for sentiment analysis," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 997–1007.

[38] M. Gridach, H. Haddad, and H. Mulki, "Empirical evaluation of word representations on arabic sentiment analysis," in *Proc. Int. Conf. Arabic Lang. Process.* Cham, Switzerland: Springer, vol. 782, 2018, pp. 147–158.

[39] M. A. Zahran, A. Magooda, A. Y. Mahgoub, H. Raafat, M. Rashwan, and A. Atyia, "Word representations in vector space and their applications for arabic," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, 2015, pp. 430–443.

[40] M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2015, pp. 2515–2519.

[41] A. Al-Sallab, R. Baly, H. Hajj, K. B. Shaban, W. El-Hajj, and G. Badaro, "Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 16, no. 4, p. 25, Sep. 2017.

[42] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, "A large scale arabic sentiment lexicon for arabic opinion mining," in *Proc. ANLP*, Oct. 2014, pp. 165–173.

[43] S. Al-Azani and E.-S. M. El-Alfy, "Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text," *Procedia Comput. Sci.*, vol. 109, pp. 359–366, Jan. 2017.

[44] A. A. Altowayan and L. Tao, "Word embeddings for arabic sentiment analysis," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 3820–3825.

[45] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in *Proc. IEEE Jordan Conf. Appl. Elect. Eng. Comput. Technol. (AEECT)*, Dec. 2013, pp. 1–6.

[46] A. Mourad and K. Darwish, "Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs," in *Proc. 4th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal. (WASSA)*, Jun. 2013, pp. 55–64.

[47] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for arabic sentiment analysis," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*, 2018, pp. 179–191.

[48] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, and M. N. Al-Kabi, "A comprehensive survey of arabic sentiment analysis," *Inf. Process. Manag.*, vol. 56, no. 2, pp. 320–342, Mar. 2019.

[49] G. Badaro, R. Baly, H. Hajj, W. El-Hajj, K. B. Shaban, N. Habash, A. Al-Sallab, and 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 Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 18, no. 3, p. 27, May 2019.

[50] M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, and B. Gupta, "Deep learning for arabic NLP: A survey," *J. Comput. Sci.*, vol. 26, pp. 522–531, May 2018.

[51] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, "Word embeddings and convolutional neural network for arabic sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguistics*, Dec. 2016, pp. 2418–2427.

[52] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of arabic word embedding models for use in arabic NLP," *Procedia Comput. Sci.*, vol. 117, pp. 256–265, Jan. 2017.

[53] N. Al-Twairesh, H. Al-Khalifa, and A. Al-Salman, "AraSenTi: Large-scale Twitter-specific arabic sentiment lexicons," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 697–705.

[54] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman, and Y. Al-Ohali, "AraSenTi-tweet: A corpus for arabic sentiment analysis of Saudi tweets," *Procedia Comput. Sci.*, vol. 117, pp. 63–72, Jan. 2017.

[55] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[57] N. Al-Twairesh, H. Al-Khalifa, A. Alsalman, and Y. Al-Ohali, "Sentiment analysis of arabic tweets: Feature engineering and a hybrid approach," May 2018, *arXiv:1805.08533*. [Online]. Available: https://arxiv.org/abs/1805.08533

[58] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2004, pp. 168–177.

NORA AL-TWAIRESH received the Ph.D. degree in computer science from King Saud University. She is currently an Assistant Professor with the Information Technology Department, College of Computer and Information Sciences, King Saud University (KSU). She is also a member of the iWAN Research Group, KSU. She has published several research papers. Her research interests include natural language processing, data science, and social media mining. She has served as a Program Committee Member in many national and international conferences and as a Reviewer for several journals.

HADEEL AL-NEGHEIMISH received the M.Sc. degree in computing, with a specialization in artificial intelligence, from Imperial College London, in 2015, where she is currently pursuing the Ph.D. degree in machine learning. She is also a Lecturer of computer science with King Saud University, Riyadh. Her research interests include relational learning and natural language processing.