An Ensemble of Arabic Transformer-based Models for Arabic Sentiment Analysis

Ikram El Karfi, Sanaa El Fkihi
ENSIAS, Mohammed V University, Rabat, Morocco

***Abstract*—In recent years, sentiment analysis has gained momentum as a research area. This task aims at identifying the opinion that is expressed in a subjective statement. An opinion is a subjective expression describing personal thoughts and feelings. These thoughts and feelings can be assigned with a certain sentiment. The most studied sentiments are positive, negative, and neutral. Since the introduction of attention mechanism in machine learning, sentiment analysis techniques have evolved from recurrent neural networks to transformer models. Transformer-based models are encoder-decoder systems with attention. Attention mechanism has permitted models to consider only relevant parts of a given sequence. Making use of this feature in encoder-decoder architecture has impacted the performance of transformer models in several natural language processing tasks, including sentiment analysis. A significant number of Arabic transformer-based models have been pre-trained recently to perform Arabic sentiment analysis tasks. Most of these models are implemented based on Bidirectional Encoder Representations from Transformers (BERT) such as AraBERT, CAMeLBERT, Arabic ALBERT and GigaBERT. Recent studies have confirmed the effectiveness of this type of models in Arabic sentiment analysis. Thus, in this work, two transformer-based models, namely AraBERT and CAMeLBERT have been experimented. Furthermore, an ensemble model has been implemented to achieve more reasonable performance.**

Keywords*—*Transformers; BERT; ensemble learning; Arabic sentiment analysis

I. Introduction

Given the tremendous amounts of Arabic digital content that has been produced in the last couple of years, an increasing number of research works have been devoted to the automatic processing of this language. In this regard, different techniques have been used to classify a specific text. Many studies have addressed this task by making use of basic machine learning models such as Naïve Bayes (NB) and Support Vector Machine (SVM). The authors in [1] addressed Arabic text classification using SVM and NB combined with the N-gram feature. The best accuracy of SVM was achieved without the N-gram, as for NB the best accuracy was achieved when the N-gram feature was considered. Whereas the authors in [2] introduced their Arabic Jordanian twitter corpus, then evaluated N-grams and stemming techniques together with TF-IDF or TF weighting schemes. Experiments have been carried out by making use of SVM and NB. Results showed that training SVM model on top of stems and bigrams using TF-IDF could give better performance compared to NB model. In a similar work [3], the authors performed sentiment analysis on Arabizi text which is Arabic text written in Latin alphabets. For experimentation purposes, the authors used NB and SVM classifiers. Besides, they evaluated the filtering step, which consists of removing stop words and mapping emojis to their corresponding Arabic words. Results indicated that SVM model outperformed NB model. However, filtering step did not greatly improve the accuracy.

Recently deep learning models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) have proven to be efficient for analyzing Arabic content. Many researchers have relied on deep learning models to tackle Arabic sentiment analysis task. In [4] performed a binary sentiment classification in Arabic. Initially, they applied preprocessing to clean input texts. Next, a word embedding layer has been used to represent texts as numerical vectors to be fed to the LSTM layer. Finally, a SoftMax layer followed to predict the polarity of the text. The experiments showed quite good results with an accuracy ranging from 80% to 82%. In another work [5] the authors made an empirical comparison between deep learning models (LSTM, CNN) and other machine learning models for both binary and multiclass classification using different datasets. The paper showed that deep learning models are effective for larger datasets. In contrast, basic machine learning algorithms perform well on smaller datasets.

More recently, with the increasing popularity of transformer models, sentiment analysis task has been significantly improved in terms of performance. Transformer models have replaced deep learning models and achieved state-of-the-art results on many automatic language processing tasks such as sentiment analysis [6], named entity recognition [7], question answering [8], and many other tasks.

The ineffectiveness of the existing methods performing sentiment analysis in Arabic is the main motivation for proposing a transformer-based ensemble method. In the last few years transformer language models alone led to significant improvements in sentiment analysis. Hence, making use of the advantages of this type of models to investigate more reliable approaches is indeed necessary to tackle sentiment analysis in Arabic being a morphologically rich language. In this paper, we propose an ensemble model that combines the strengths of two transformer language models.

Many different disciplines have made significant use of ensemble approaches to address text classification. However, there is relatively few studies on the use of ensemble methods in Arabic sentiment analysis. The primary goal in this paper is to propose an ensemble model that combines two base transformer models namely AraBERT and CAMeLBERT into a single model. To the best of our knowledge this is the first

study that investigates the ensemble of transformer-based models in Arabic sentiment analysis. Experimental results demonstrate that the proposed ensemble method outperforms stand-alone classifiers and majority voting ensemble model.

The rest of this paper is structured as follows. Related works will be introduced in Section II. The overall proposed methodology will be discussed in Section III. Experiments and results are given in Section IV. Then conclusions are drawn in Section V.

II. Related Work

Given the effectiveness of transformer-based models, there have been various transformer models used in Arabic sentiment analysis. The widely utilized models are Multilingual BERT, AraBERT, and MARBERT [9]. The author in [10] addressed sentiment analysis in Modern Standard Arabic (MSA) and other Arabic dialects such as Levantine, Egyptian, and Gulf. The author opted for three-way classification according to three scales (positive, neutral, and negative) and using different algorithms, namely: Naive Bayes classifiers (NB), Support Vector Machine (SVM), Random Forest Classifier, and BERT model (Bidirectional Encoder Representations from Transformers). The best results are obtained with BERT model reaching an accuracy score of more than 83%. The author in [11] addressed sarcasm and sentiment detection using two variants of transformer-based models, namely AraELECTRA and AraBERT. Evaluation results showed that AraBERT performs the best in terms of accuracy for both sarcasm and sentiment detection. In a similar work, [12] addressed the same tasks: sarcasm detection and sentiment analysis. The authors have examined six BERT-based models including: MARBERT [13], QARiB [14], AraBERTv02 [15], GigaBERTv3 [16], Arabic BERT [17], and mBERT [18]. MARBERT achieved promising results for both tasks.

Several studies in the literature investigated ensemble methods in Arabic sentiments analysis. The authors in [19] investigated different deep learning models to improve Arabic sentiment analysis accuracy. The authors proposed an ensemble model combining a Convolutional Neural Network (CNN) model and a Long Short-Term Memory (LSTM) model. To evaluate their model, they used the ASTD dataset [20] which consists of 10000 tweets. In this work, they focused only on opinion classification, hence the objective class tweets were removed. To construct their ensemble model, they experimented different CNN models and LSTM models with different hyper-parameters. The best CNN model is obtained by configuring the parameter fully connected layer size to 100. As for LSTM, the best model is obtained by using a dropout rate of 0.2, based on the best CNN model and the best LSTM model they built an ensemble model where the final predicted class is obtained using soft voting. Results show that the ensemble model achieved better results in terms of accuracy and F1-score compared to LSTM model and CNN model. In another study [21] , the authors introduced their approach to address three SemEval related sentiment analysis subtasks in Arabic. First Subtask (A) addresses Message Polarity Classification, then Subtask (B) addresses Topic-Based Message Polarity Classification, finally Subtask (D) which addresses Tweet quantification. The authors proposed two systems, the first is developed using their previous proposed sentiment analyzer [22] based on a scored lexicon. The second system is an ensemble of three different classifiers namely Convolutional Neural Network using Word2vec, Multilayer Perceptron and Logistic Regression. Using voting between the three classifiers the authors determined the final outcome. Evaluation results showed that their systems outperformed all the other systems by achieving an accuracy of 0.58 and 0.77 on Subtask A and Subtask B respectively, as for Subtask D their system outperformed the other systems as well by achieving 0.127 in terms of KLD score.

It seems clear that none of the existing ensemble methods has addressed Arabic sentiment analysis by making use of transformer language models. Accordingly, this study will be focusing on investigating the use of transformer language models in Arabic sentiment classification as well as proposing an ensemble technique based on transformer models to enhance classification accuracy.

III. Proposed Methodology

This section presents the methodology proposed in this paper. We will be discussing the background of transformer-based models and the models adopted in this work. Then, describe the proposed ensemble model architecture.

A. Background

A variety of neural network architectures have been proposed and used for text classification tasks, including sentiment classification. Among these numerous architectures, the best adapted architecture to sequential data is recurrent neural networks (RNNs). They have demonstrated to be effective on data where elements order is important. For example, in a sentence, the order in which words occur has a significant impact on the meaning of that sentence. Since its introduction, RNNs have been the state-of-the-art for capturing and processing dependencies in sequences. Nevertheless, it also has its drawbacks, it has been proved that RNNs cannot process large sequences of text such as long paragraphs. Moreover, in practice, data is processed sequentially, which makes it difficult to perform parallel computing using RNNs. Recently, a new architecture called Transformers has been introduced. Similar to RNNs, transformers use attention mechanism and inherit the encoder-decoder architecture of the sequence-to-sequence models to deal with sequential data. However, its architecture does not involve recurrent networks in order to speed up the training process. Transformers were firstly introduced by [23], and they were initially designed to perform translation. As illustrated below in Fig. 1, a transformer consists of two blocks, on the left, the encoder stack, and on the right, the decoder stack. The encoder stack is made up of a multi-head self-attention layer and a fully connected feed forward network. In addition to these two layers, the decoder stack has one more layer called the masked multi-head attention layer. As transformer architecture does not rely on recurrence, word position is not provided. To preserve this information positional encoding technique has been introduced. In addition to the input embedding vector, a positional vector with the same dimension as the input embedding vector is added to capture the context of a word in a sentence.

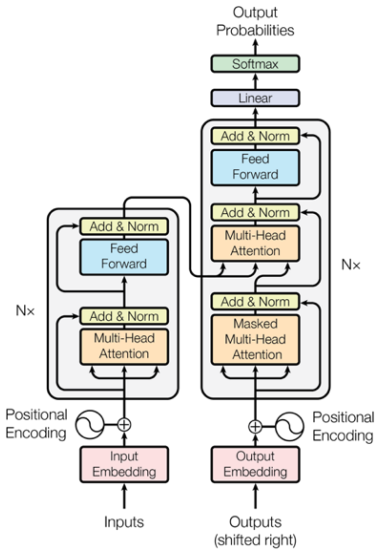

Fig. 1. Transformer Architecture [23].

Transformer-based models include three types of models: encoder-only, decoder-only, and encoder-decoder, following a brief introduction to each type of transformers, its architecture and its applications.

1) Encoder-only models: In this type of transformers only the encoder part is needed. A vector representing the input sequence is fed to the first encoder block that consists of a bi-directional self-attention layer and a feed-forward layer, the output of this block is passed to the following encoder block, which itself is composed of two layers. Each encoder block tries to enrich the embedding vector with contextual information. The final encoder block outputs the last contextual encoding. This type of transformer is suitable for tasks such as text classification or named entity recognition. The most popular examples of this type of models are: BERT [18], ELECTRA [24], and RoBERTA [25].

2) Decoder-only models: In decoder models only decoder stack is used. It consists of N identical decoder blocks; a single decoder block is composed of three layers. The first layer is a masked multi-head attention layer, in which future information is masked and only previous positions in the input sequence have attention. Similarly, to the encoder block, the next layers are multi-head self-attention layers and a fully connected feed-forward network. Decoder-only based models are also called autoregressive models and are more suited for tasks such as text generation. The most widely used models trained with decoder-only architectures are GPT (Generative Pre-trained Transformer) [26].

3) Encoder-decoder models: Also called sequence-to-sequence models, this type of models is implemented using both blocks: encoder block and decoder block. In the encoder block, the whole sequence is considered, whereas in the decoder block for a given word, only the words that precede are considered. Encoder-decoder models are best suited for tasks that involve the input of a sequence of items (words, letters, etc.) and then outputs another sequence. This architecture can be applied in the case of machine translation or question answering, where a sequence of words is treated sequentially and the result is also a sequence of words. Recently, many encoder-decoder based models have been introduced.

B. Transformer Language Models for Arabic

Transformers were initially introduced as a novel architecture for translation. Ever since, they have been mostly used for natural language processing. In sentiment analysis task, pretrained transformer language architectures have significantly improved the performance of models. Each model has its own size and trained on different type of datasets. Table I summarizes the most applied models in Arabic text classification.

In this paper, we have selected some of the most effective Arabic transformation models in sentiment analysis in Arabic. Each of these models is based on different architectures and trained using different Arabic variants. Hereafter, we discuss each of the selected models and their architecture.

1) AraBERT [27] pretrained BERT model using a pretrained dataset of 70 million sentences, collected from Wikipedia dumps, Arabic news websites and two large corpora: Abulkhair Arabic Corpus [31] and OSIAN [32]. AraBERT comes in two versions AraBERTv0.1 and AraBERTv1. In this study AraBERTv0.2 is used for experiments.

TABLE I. SUMMARY OF ARABIC PRETRAINED MODELS

Model name	Ref	Size	Source	Data type	Parameters
Multilingual BERT	[18]	-	Wikipedia	MSA	110M
AraBERT	[27]	24GB	Wikipedia+ Abulkhair Corpus+ OSIAN+ news websites	MSA	136M(base) 371M(large)
ArabicBERT	[17]	95GB	Wikipedia+ OSCAR+ other sources	MSA/ Dialect	110M(base) 340M(large)
CAMeLBERT	[28]	167B	Gigaword+ Abulkhair Corpus+ OSIAN+ Wikipedia+ OSCAR+ dialectal corpora+ OpenITI corpus	MSA / Dialect/ Classical	17.3B
MARBERT	[13]	128GB	Twitter API	MSA/ Dialect	160M
Arabic ALBERT	[29]	-	OSCAR+ Wikipedia	MSA	12M(base) 18M(large) 60M(xlarge)
GigaBERT	[16]	-	Gigaword+ Wikipedia+ OSCAR	MSA	125M
XLM-RoBERTa	[30]	2.5TB	CommonCrawl	MSA	270M(base) 550M(large)

2) CAMeLBERT [28] implemented their Arabic pre-trained language model on top of three variants of Arabic: Modern Standard Arabic (MSA), dialectal Arabic, and classical Arabic. The authors evaluated the proposed model by making use of 12 datasets to address five tasks: Named Entity Recognition, POS tagging, sentiment analysis, dialect identification, and poetry classification.

C. Ensemble Models

Ensemble learning is a technique that combines multiple machine learning models to improve the performance of the learning model and achieve a higher accuracy score than would be achieved by any single model in the ensemble. In this study, two ensemble techniques have been evaluated. The first technique is the majority voting. It is the most commonly used technique for ensemble learning. The second technique is based on calculating the sum of raw outputs of each model in the ensemble.

1) Majority voting: In majority voting, the final output of the ensemble model is determined by counting for each class the number of votes of multiple models. The class with the majority of votes is predicted.

2) The proposed method SUM: As illustrated in Fig. 2, that represents the proposed ensemble model. Firstly, a raw text is fed to the model as input, then transformed into vector representation so that it can be processed with encoder-decoder approach. Then the decoder-block of each model outputs probabilities for each class. Finally, the output is obtained by calculating the weighted sum of the probabilities from the same class, then for each class, the argmax operation is applied to find the class with higher probability value.

For a given text, let PAr_{Neg} and PAr_{Pos} denote the probabilities predicted with AraBERT model for the class Negative and the class Positive respectively. Whereas, $PCaM_{Neg}$ and $PCaM_{Pos}$ denote the probabilities predicted with CAMeLBERT model for the class Negative and the class Positive respectively. For each class, the final probability is obtained by calculating the weighted sum of both probabilities, namely the probability given with AraBERT model and CAMeLBERT model. Weights values are not selected randomly, the main reason of selecting weight values 0, 7 and 0, 3 for CAMeLBERT and AraBERT respectively, is that CAMeLBERT model tend to perform well on the majority of the datasets (see Table II). Thus, we considered 70% of the probability generated by CAMeLBERT model and 30% of the probability generated by AraBERT model. The final probabilities are calculated using the following equations:

$$PF_{Neg} = (0.3 \times PAr_{Neg}) + (0.7 \times PCaM_{Neg})$$

$$PF_{Pos} = (0.3 \times PAr_{Pos}) + (0.7 \times PCaM_{Pos})$$

Next, the final output corresponds to the index with higher probability value.

$$\text{Final Output} = \text{argmax}([PF_{Neg}, PF_{Pos}])$$

Therefore, if the output index is 0, the model will assign to the input text the class Negative, and if the output index is 1 the model will assign Positive.

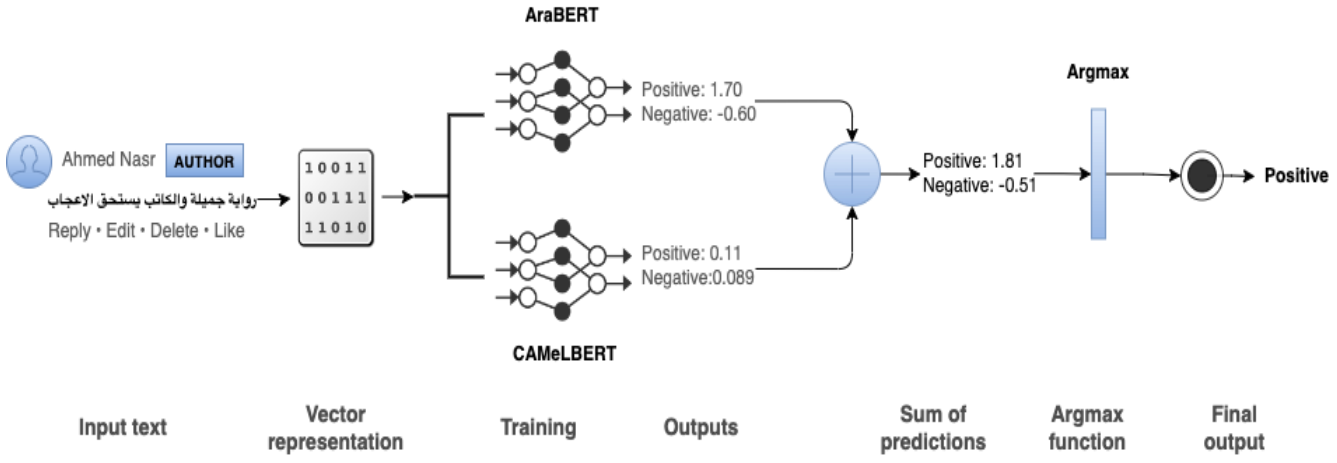

Fig. 2. The Architecture of the Proposed Ensemble Model (SUM).

TABLE II. ACCURACY RESULTS OF DIFFERENT MODELS ON THREE PUBLICALY AVAILABLE ARABIC DATASETS

			Negative class			Positive class			Accuracy	Macro-F1
			Precision	Recall	F1-score	Precision	Recall	F1-score		
Twitter Dataset		Abdulla et al. (SVM) [33]	-	-	-	-	-	-	87.2	-
	Unbalanced	Dahou et al. (CNN) [34]	-	-	-	-	-	-	85.01	-
		AraBERT	94.03	96.43	95.21	96.32	93.85	95.06	95.14	95.14
		CAMeLBERT	94.97	96.43	95.70	96.35	94.87	95.61	95.65	95.65
		Majority Voting	93.24	98.47	95.78	98.37	92.82	95.51	95.65	95.65
		SUM (ours)	95.02	96.22	96.22	97.37	94.87	96.10	**96.16**	**96.16**

	Balanced	Dahou et al. (CNN) [34]	-	-	-	-	-	-	86.3	-
		AraBERT	95.52	95.52	96.00	95.14	96.17	95.65	95.83	95.83
		CAMeLBERT	96.30	90.55	93.33	90.26	96.17	93.12	93.23	93.23
		Majority Voting	94.58	95.52	95.05	95.03	93.99	94.51	94.78	94.78
		SUM (ours)	97.87	91.54	94.60	91.33	97.81	94.46	94.53	94.53
Gold Dataset		Refaee and Rieser (SVM) [35]	-	-	-	-	-	87.74	-	
	Unbalanced	Dahou et al. (CNN) [34]	-	-	-	-	-	75.8	-	
		AraBERT	94.72	87.99	91.23	73.51	87.18	79.77	87.77	85.50
		CAMeLBERT	94.63	90.69	92.62	78.03	86.54	82.07	89.54	87.34
		Majority Voting	93.20	94.12	93.66	84.21	82.05	83.12	**90.78**	**88.39**
		SUM (ours)	94.36	90.20	92.23	77.01	85.90	81.21	89.01	86.72
	Balanced	Dahou et al. (CNN) [34]	-	-	-	-	-	-	73.8	-
		AraBERT	85.80	83.73	84.76	85.71	87.57	86.63	85.75	86.63
		CAMeLBERT	86.83	87.35	87.09	88.59	88.11	88.35	87.75	87.72
		Majority Voting	82.97	90.96	86.78	91.12	83.24	87.01	86.89	86.89
		SUM (ours)	87.95	87.95	87.69	89.13	88.65	88.89	88.32	88.29
ASTD Dataset	Unbalanced	Dahou et al. (CNN) [34]	-	-	-	-	-	-	79.07	-
		AraBERT	93.38	85.30	89.16	71.67	86.00	78.18	85.51	83.67
		CAMeLBERT	91.47	89.63	90.54	77.07	80.67	78.83	86.92	84.68
		Majority Voting	90.37	91.93	91.14	80.56	77.33	78.91	87.53	85.03
		SUM (ours)	92.42	87.90	90.10	74.85	83.33	78.86	86.52	84.48
	Balanced	Dahou et al. (CNN) [34]	-	-	-	-	-	-	75.9	-
		AraBERT	86.90	83.44	85.14	85.71	88.76	87.21	86.25	86.17
		CAMeLBERT	87.76	85.43	86.58	87.28	89.35	88.30	87.50	87.44
		Majority Voting	83.95	90.07	86.90	90.51	84.62	87.46	87.19	87.18
		SUM (ours)	89.80	87.42	88.59	89.02	91.12	90.06	**89.38**	**89.32**

IV. Experiments

In this section, we discuss the implemented models and their results. Two transformer language models are experimented namely CAMeLBERT and AraBERT, an ensemble of these two models, as well as majority voting ensemble model.

A. Dataset

For experimentation purposes, in this work we consider four datasets of different sizes and sources. The first dataset is Twitter dataset collected by Abdulla et al,. [33] composed of about 2000 tweets, written in MSA and Jordanian dialect. And consisted of 958 negative tweets and 993 positive tweets. The second dataset is Arabic Gold-Standard Twitter dataset collected by Refaee and Rieser [35] composed of 6512 tweets, divided in three classes: Negative, neutral, and positive. The negative class contains 1941 tweets, the neutral class contains 3694, and the positive class contains 876 tweets. In this study we perform a binary classification, thus only Positive and negative classes are utilized. The third dataset is Arabic sentiment tweets dataset (ASTD) proposed by [19] which contains 10006 tweets written in MSA and Arabic dialect. Tweets are labeled as one of four classes: negative, positive, neutral, and objective. As this study focuses on binary classification only positive and negative tweets are considered. After data preprocessing, we are left with 1684 negative tweets and 799 positive tweets. The fourth dataset is a dataset that we have proposed in a previous work [36] which is consisted of 1299 Modern Standard Arabic books reviews with a balance between positive and negative reviews. Reviews are collected from Goodreads website and annotated manually. After data collection, each given text is decomposed into tokens. Then, Arabic stop words are filtered out as they do not hold any information. The next step is normalization, where elongation, hamza, and diacritics are removed. Finally, all emoticons and emojis are deleted based on a preselected list of the most commonly used emoticons and emojis.

B. Results

To investigate the effectiveness of the proposed ensemble model three models have been implemented, including two transformer language-based models: AraBERT and CAMeLBERT and majority voting model. All models are trained on the same training set, which represents 80% of the whole dataset, and tested on the same testing set composed of the 20% remaining data. For each of the four datasets the models are trained and tested on both balanced and unbalanced datasets. Performance results of the proposed ensemble method are compared with stand-alone models and majority voting ensemble model. The models are evaluated using accuracy, F1-score, precision and recall metrics. The mathematical formulas of each of the used metrics is defined as follows:

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

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

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

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Where, TP, FP, TN, and FN refer to "True Positive", "False Positive", "True Negative", and "False Negative" respectively.

Table II shows the performance of each model on balanced and unbalanced datasets compared to existing models. As can be seen, it is clear that ensemble models provide remarkable improvement over baseline models. Majority voting model has achieved the best accuracy score on unbalanced Gold dataset with an accuracy score of 90.78% followed by our ensemble model with 89.01%, whereas on Twitter, and ASTD datasets the best accuracy score is achieved by our proposed ensemble model SUM. On Twitter dataset SUM model achieved an accuracy score of 96.16% with unbalanced dataset followed by majority voting model and CAMeLBERT model with the same accuracy score of 95.65%. As for ASTD dataset our proposed model outperformed all other models by achieving the first best accuracy score of 89.38% on balanced dataset, the second-best accuracy score is achieved by majority voting model with 87.53% on unbalanced dataset.

The results of all proposed models on our dataset are shown on Table III. AraBERT model has achieved better results than CAMeLBERT in terms of accuracy and F1-score. On the other hand, the performance of ensemble models varies from one model to another. Compared to the proposed ensemble model, majority voting model has failed to improve the performance. It has achieved an accuracy of 94.98% against an accuracy of 95.75% achieved by AraBERT. Whereas, our proposed ensemble model has reached the best results in terms of accuracy and F1-score. The mediocre performance of majority voting may be explained by the size of the dataset and the number of combined models.

TABLE III. COMPARISON OF DIFFERENT MEASURES OF PERFORMANCE ON OUR DATASET

Model	Accuracy	F1-score	Recall	Precision
AraBERT	95.75	95.72	96.09	95.35
CAMeLBERT	92.66	92.66	93.75	91.60
Majority Voting	94.98	94.78	92.19	97.52
SUM (ours)	**96.53**	**96.50**	**96.88**	96.12

In summary, the best results have been achieved by our proposed ensemble model on balanced datasets. Thus, it is obvious from the conducted comparative experiments that training models on balanced data can improve classification performance, it can help models to learn better and achieve better accuracy results.

V. CONCLUSION

In this work, we have implemented an ensemble model based on two transformer language models, namely AraBERT and CAMeLBERT. The proposed ensemble model was evaluated on top of our balanced dataset composed of modern standard Arabic book reviews. In addition, to investigate more the performance of our proposed model it has been trained on top of three other datasets namely Twitter dataset, Gold dataset and ASTD dataset. Compared to majority voting and the two stand-alone transformer-based models, our proposed ensemble model has achieved the highest score of accuracy and F1 metrics on all datasets. In this paper, we have proposed a domain-independent model, the proposed ensemble model has achieved state-of-the-art on several datasets of different sources and domains. Thus, researchers can adopt our proposed model to address sentiment analysis in Arabic regardless of data type (MSA/Dialect) and domain. To continue working towards improving the model's performance, for future work, we plan to experiment more transformer models, combine multiple models and evaluate all possible combinations to determine the optimized model. Finally, we will be considering increasing the size of our training set as accuracy increases with the size of training data.

REFERENCES

[1] H. Al-Rubaiee, R. Qiu, and D. Li, "Identifying Mubasher software products through sentiment analysis of Arabic tweets," 2016 Int. Conf. Ind. Informatics Comput. Syst., pp. 1–6, 2016.

[2] K. M. Alomari, H. M. Elsherif, and K. Shaalan, "Arabic tweets sentimental analysis using machine learning," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017, vol. 10350 LNCS, pp. 602–610, doi: 10.1007/978-3-319-60042-0_66.

[3] R. Duwairi, M. Faqeeh, M. Wardat, and A. Alrabadi, "Sentiment analysis for Arabizi text," 2016, pp. 127–132, doi: 10.1109/IACS.2016.7476098.

[4] A. Albayati and A. Al-Araji, "Arabic Sentiment Analysis (ASA) Using Deep Learning Approach," Univ. Baghdad Eng. J., vol. 26, pp. 85–93, 2020, doi: 10.31026/j.eng.2020.06.07.

[5] A. Soufan, "Deep learning for sentiment analysis of Arabic text," 2019, doi: 10.1145/3333165.3333185.

[6] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-Based Sentiment Analysis using BERT," 2019.

[7] U. Naseem, M. Khushi, V. Reddy, S. Rajendran, I. Razzak, and J. Kim, "BioALBERT: A Simple and Effective Pre-trained Language Model for Biomedical Named Entity Recognition," in 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–7, doi: 10.1109/IJCNN52387.2021.9533884.

[8] W. Yoon, J. Lee, D. Kim, M. Jeong, and J. Kang, "Pre-trained Language Model for Biomedical Question Answering BT - Machine Learning and Knowledge Discovery in Databases," 2020, pp. 727–740.

[9] A. S. Alammary, "BERT Models for Arabic Text Classification: A Systematic Review," Appl. Sci., vol. 12, no. 11, p. 5720, 2022.

[10] S. Bilal, "A Linguistic System for Predicting Sentiment in Arabic Tweets," in 2021 3rd International Conference on Natural Language Processing (ICNLP), 2021, pp. 134–138, doi: 10.1109/ICNLP52887.2021.00028.

[11] A. Wadhawan, "Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets," arXiv Prepr. arXiv2103.01679, 2021.

[12] A. Abuzayed and H. Al-Khalifa, "Sarcasm and Sentiment Detection In {A}rabic Tweets Using {BERT}-based Models and Data Augmentation," in Proceedings of the Sixth Arabic Natural Language Processing Workshop, Apr. 2021, pp. 312–317, [Online]. Available: https://aclanthology.org/2021.wanlp-1.38.

[13] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic," 2021, doi: 10.48550/ARXIV.2101.01785.

[14] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, "Pre-Training BERT on Arabic Tweets: Practical Considerations," 2021.

[15] 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, May 2020, pp. 9–15, [Online]. Available: https://aclanthology.org/2020.osact-1.2.

[16] W. Lan, Y. Chen, W. Xu, and A. Ritter, "An Empirical Study of Pre-trained Transformers for {A}rabic Information Extraction," in Proceedings of the 2020 Conference on Empirical Methods in Natural

Language Processing (EMNLP), Nov. 2020, pp. 4727–4734, doi: 10.18653/v1/2020.emnlp-main.382.

[17] A. Safaya, M. Abdullatif, and D. Yuret, "{KUISAIL} at {S}em{E}val-2020 Task 12: {BERT}-{CNN} for Offensive Speech Identification in Social Media," in Proceedings of the Fourteenth Workshop on Semantic Evaluation, Dec. 2020, pp. 2054–2059, doi: 10.18653/v1/2020.semeval-1.271.

[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Jun. 2019, vol. 1, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[19] M. Heikal, M. Torki, and N. El-Makky, "Sentiment Analysis of Arabic Tweets using Deep Learning," Procedia Comput. Sci., vol. 142, pp. 114–122, 2018, doi: 10.1016/j.procs.2018.10.466.

[20] M. Nabil, M. Aly, and A. F. Atiya, "ASTD: Arabic sentiment tweets dataset," in Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, Sep. 2015, pp. 2515–2519, doi: 10.18653/v1/d15-1299.

[21] S. R. El-Beltagy, M. El Kalamawy, and A. B. Soliman, "NileTMRG at SemEval-2017 Task 4: Arabic Sentiment Analysis," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Aug. 2017, pp. 790–795, doi: 10.18653/v1/S17-2133.

[22] S. R. El-Beltagy, "NileULex: A phrase and word level sentiment lexicon for Egyptian and modern standard Arabic," in Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, 2016, pp. 2900–2905.

[23] A. Vaswani et al., "Attention is All you Need," in Advances in Neural Information Processing Systems, 2017, vol. 30, [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[24] K. Clark, M.-T. Luong, Q. V Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." arXiv, 2020, doi: 10.48550/ARXIV.2003.10555.

[25] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach." arXiv, 2019, doi: 10.48550/ARXIV.1907.11692.

[26] A. Radford and K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018.

[27] W. Antoun, F. Baly, and H. Hajj, "{A}ra{BERT}: Transformer-based Model for {A}rabic Language Understanding," in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, May 2020, pp. 9–15, [Online]. Available: https://aclanthology.org/2020.osact-1.2.

[28] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The Interplay of Variant, Size, and Task Type in {A}rabic Pre-trained Language Models," in Proceedings of the Sixth Arabic Natural Language Processing Workshop, Apr. 2021, pp. 92–104, [Online]. Available: https://aclanthology.org/2021.wanlp-1.10.

[29] A. Safaya, "Arabic-ALBERT." Zenodo, Aug. 2020, doi: 10.5281/zenodo.4718724.

[30] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Jul. 2020, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.

[31] I. A. El-khair, "1.5 billion words Arabic Corpus," ArXiv, vol. abs/1611.0, 2016, [Online]. Available: http://arxiv.org/abs/1611.04033.

[32] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, "{OSIAN}: Open Source International {A}rabic News Corpus - Preparation and Integration into the {CLARIN}-infrastructure," in Proceedings of the Fourth Arabic Natural Language Processing Workshop, Aug. 2019, pp. 175–182, doi: 10.18653/v1/W19-4619.

[33] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013, 2013, pp. 1–6, doi: 10.1109/AEECT.2013.6716448.

[34] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, "Word embeddings and convolutional neural network for Arabic sentiment classification," in COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers, Dec. 2016, pp. 2418–2427, [Online]. Available: https://www.aclweb.org/anthology/C16-1228.

[35] E. Refaee and V. Rieser, "An Arabic twitter corpus for subjectivity and sentiment analysis," in Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, 2014, pp. 2268–2273.

[36] I. El Karfi, S. El Fkihi, and R. Faizi, "A spectral clustering-based approach for sentiment classification in modern standard Arabic," in MCCSIS 2018 - Multi Conference on Computer Science and Information Systems; Proceedings of the International Conferences on Big Data Analytics, Data Mining and Computational Intelligence 2018, Theory and Practice in Modern Computing 2018 and Connected Sma, 2018, pp. 59–65, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054167622%5C&partnerID=40%5C&md5=101982324dd788664f052d2919012106.