IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 November 2023, accepted 5 December 2023, date of publication 13 December 2023, date of current version 20 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3342755

RESEARCH ARTICLE

Enhancing Arabic Aspect-Based Sentiment Analysis Using End-to-End Model

GHADA M. SHAFIQ, TAHER HAMZA, MOHAMMED F. ALRAHMAWY, AND REEM EL-DEEB
Department of Computer Science, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt
Corresponding author: Ghada M. Shafiq (ghadashafiq@mans.edu.eg)

ABSTRACT The majority of research on the Aspect-Based Sentiment Analysis (ABSA) tends to split this task into two subtasks: one for extracting aspects, Aspect Term Extraction (ATE), and another for identifying sentiments toward particular aspects, Aspect Sentiment Classification (ASC). Although these subtasks are closely related, they are performed independently; while performing the Aspect Sentiment Classification task, it is assumed that the aspect terms are pre-identified, which ignores the practical interaction required to properly perform the ABSA. This study addresses these limitations using a unified End-to-End (E2E) approach, which combines the two subtasks into a single sequence labeling task using a unified tagging schema. The proposed model was evaluated by fine-tuning the Arabic version of the Bidirectional Encoder Representations from Transformers (AraBERT) model with a Conditional Random Fields (CRF) classifier for enhanced target-polarity identification. The experimental results demonstrated the efficiency of the proposed fine-tuned AraBERT-CRF model, which achieved an overall F1 score of 95.11% on the SemEval-2016 Arabic Hotel Reviews dataset. The model's predictions are then subjected to additional processing, and the results indicate the superiority of the proposed model, achieving an F1 score of 97.78% for the ATE task and an accuracy of 98.34% for the ASC task, outperforming previous studies.

INDEX TERMS Sentiment analysis, aspect-based, AraBERT, CRF, transfer learning.

I. INTRODUCTION

Due to the development of the internet and its platforms, users frequently share their ideas on various blogs and social media platforms. Understanding and analyzing users' ideas and opinions about a product or service is essential for businesses and owners. Sentiment Analysis (SA) is concerned with this type of information; it understands human opinions and analyzes them to obtain the required knowledge. SA can be performed at the document, sentence, or aspect levels [1]. The Aspect-Based Sentiment Analysis (ABSA) is the most challenging type. It is usually divided into four subtasks: Aspect Term Extraction (ATE), the first subtask, which aims to extract the explicit aspect terms in each sentence; and Aspect Sentiment Classification (ASC), the second subtask, which seeks to identify the sentiment polarities toward the given aspects. For example, the sentence *"the food is delicious, but the service is terrible"*, contains the aspect term "*food*" with a *positive* sentiment polarity and the aspect term "*service*" with a *negative* sentiment polarity.

Aspect Category Identification and Aspect Category Sentiment Classification are the remaining subtasks; their goal is to identify a category for an aspect word from a pre-defined set of categories and determine its corresponding sentiment, respectively. In this research, we are concerned only with the ATE and ASC subtasks.

Arabic ABSA has gained some attention over the past few years. However, the research involved still needs to be improved due to the relative complexity and ambiguity of the Arabic language's morphology. Additionally, the lack of publicly available annotated datasets and tools for processing Arabic text also represents a challenge [1].

Most studies involved in Arabic ABSA evaluate the ATE and ASC subtasks independently, ignoring the relatedness

The associate editor coordinating the review of this manuscript and approving it for publication was Mahdi Zareei.

TABLE 1. Example to clarify the difference among ABSA approaches applied on the Arabic sentence " الطعام لذيذ ولكن الخدمة سيئة," i.e., the food is delicious, but the service is terrible, which contains the aspect term الطعام "food" with a positive sentiment polarity and the aspect term الخدمة "service" with a negative sentiment polarity.

Task	Input	Example	Output				
ABSA	Sentence + Aspect	الطعام لذيذ ولكن الخدمة سيئة + الطعام الطعام لذيذ ولكن الخدمة سيئة + الخدمة	Positive Negative				
ATE	Sentence	الطعام لذيذ ولكن الخدمة سيئة	O	B	O	O	B
ASC	Sentence	الطعام لذيذ ولكن الخدمة سيئة	O	NEG	O	O	POS
Joint E2E-ABSA[a]	------------	-------------	O	B-NEG	O	O	B-POS
Unified E2E-ABSA	Sentence	الطعام لذيذ ولكن الخدمة سيئة	O	B-NEG	O	O	B-POS

[a] Join output labels of both ATE and ASC tasks.

and dependency of the two subtasks [2], [3], [4], [5], [6], [7]. Some studies are either only extracting aspects from a given sentence (ATE) [8], [9], [10] or predicting sentiment polarities (ASC) assuming that aspect entities are pre-identified input features to the model, which is not the case in real-world scenario [11], [12], [13], [14].

On the contrary, ABSA research in the English language is more evolved. Recent research directions are towards tackling both Aspect Term Extraction and Aspect Sentiment Classification through a single model using an End-to-End (E2E) approach, which can help overcome the limitations of previous studies. The E2E-ABSA can be carried out in one of two approaches [15]. The first approach is known as the Joint approach; it involves performing the two subtasks in parallel with two sets of labels: one for the aspect boundaries (B, I, and O) [16] denoting the Beginning, Inside, and Outside of the aspect term, respectively, for the ATE task, and the other set of labels represents the sentiment polarities (positive, negative, and neutral) for the ASC task. The outcomes of both tasks are combined to produce the final label. However, the lack of a correlation between the aspect boundaries and the corresponding sentiment polarities could cause this approach to suffer from error propagation [15].

The second approach is the unified approach, which combines the two subtasks into a single sequence labeling task. The aspect boundary labels and the sentiment polarity labels are combined to generate one set of unified labels (B-positive, I-positive, etc.). Although the unified approach preserves the dependency between the aspect boundaries and their sentiment polarities, it makes model prediction more challenging and can result in performance degradation [17]. The model must identify the aspect boundary and the sentiment polarity without providing any implicit prior information about the aspect terms. An example of an Arabic sentence that clarifies the differences among ABSA approaches is shown in TABLE 1.

Several techniques, from rule-based to traditional machine and deep learning techniques, have been used to handle the Arabic ABSA. Rule-based techniques are static techniques with no learning models involved; they also rely on external resources, which are scarce in Arabic. Machine Learning (ML) techniques, on the other hand, rely on intensive feature engineering to adjust the data and select the appropriate features. Although Deep Learning (DL) techniques have overcome the intensive feature engineering limitation, they require a large dataset for models to train and produce accurate results [9].

Recently, pre-trained transformer-based language models [18] have attracted much attention due to their significant influence on various Natural Language Processing (NLP) applications, including ABSA. A large amount of unlabeled texts were used to train these models to make them efficient in comprehending the input context. As a result, these models can be fine-tuned to handle a variety of tasks and deliver remarkable results without the need for large datasets [9], [19], [20]. AraBERT [21] is a pre-trained language model specifically designed to handle the complexities and ambiguity of the Arabic language and has achieved state-of-the-art performances in many Arabic NLP tasks.

Utilizing this model to evaluate our proposed model can have a great influence. The bi-directionality of BERT [22] allows it to learn the context of each word with respect to the entire sequence simultaneously, making it easier for the model to identify the aspect boundaries. Furthermore, the self-attention mechanism of BERT [18] allows for the association of opinion words with their relevant aspect terms in order to predict sentiment polarity.

The Conditional Random Fields (CRF) [23] classifier has also proven its efficiency in delivering accurate results in a variety of sequence labeling tasks [9]; it preserves the dependencies between tags/labels, ensuring the correctness of the predicted tag sequence and boosting overall performance.

Motivated by the aforementioned, the following is a summary of the main contributions of this study:

- This study aims to tackle the subtasks of ABSA, specifically ATE and ASC, by integrating them into a single sequence labeling task using a unified E2E approach in order to overcome the previously mentioned limitations of two separate models for each subtask. To the best of our knowledge, this is the first study to apply a unified E2E-ABSA on the SemEval-2016 Arabic Hotel Reviews dataset [2].

- Preparing the dataset of Arabic Hotel Reviews [2] so that it matches the desired classification task.
- Several experiments were applied to evaluate the proposed E2E approach, utilizing a feature-based vs. fine-tuned AraBERT model along with CRF vs. softmax [24] to assess the impact of different implementations on the performance of the proposed model.
- Resolve the complexity and morphological ambiguity of the Arabic language usingthe AraBERT model.
- Preserve the tag/label dependencies using Conditional Random Fields.
- Experimental results demonstrate that the proposed fine-tuned AraBERT-CRF model outperforms single-task methods and yields a better ABSA task representation.

The rest of the paper is structured as follows: Section II provides the related works in the ABSA field. The proposed model is presented in Section III. Section IV discusses the conducted experiments and comparisons of the achieved results with other related works. Section V shows our conclusion and future directions.

II. RELATED WORKS

This section provides an overview of the works applied to the two subtasks of Arabic ABSA, Aspect Term Extraction and Aspect Sentiment Classification, showing their advantages and limitations. However, while some studies may have covered other ABSA subtasks, our focus in this study is entirely on ATE and ASC. In addition, some of the work on English E2E-ABSA is presented due to the lack of work on Arabic E2E-ABSA.

A. ASPECT TERM EXTRACTION (ATE) AND ASPECT SENTIMENT CLASSIFICATION (ASC)

The Aspect Term Extraction task (or Opinion Target Expression (OTE) Extraction) extracts the explicit target opinionated words or phrases in each text. It is usually formulated as a sequence labeling task with a BIO tagging schema [16]. Consequently, the Aspect Sentiment Classification task identifies the sentiment polarities towards the given aspects [1]. This task is usually addressed with several naming conventions of aspect term/based polarity/sentiment identification/classification; however, for simplicity, we will refer to it as ASC.

A lot of research has been conducted regarding the two subtasks. In [25], the authors provided a benchmark annotated Arabic News Posts dataset with a lexicon-based approach to evaluate their work on aspect term extraction and aspect term polarity identification. The same authors then investigated enhancing their baseline work by utilizing a set of ML classifiers, including CRF, Naïve Bayes (NB), Decision Tree (J48: WEKA[1] implementation), and K-Nearest Neighbor (IBK: WEKA implementation) along with a set of morphological and word features including Named Entity Recognition (NER), Part-of-Speech (POS) tagging and N-Grams. Results demonstrated that the J48 classifier outperformed other classifiers regarding the ATE task, whereas the CRF classifier achieved the best performance regarding the aspect term polarity identification task [7]. The authors in [2] created a benchmark dataset of Arabic Hotel Reviews in SemEval-2016 for the ABSA task. They applied the Support Vector Machine (SVM) classifier as a baseline model. An enhanced study is introduced in [26]; the authors experimented with applying NB, Bayes Networks, J48, IBK, and SVM (SMO: WEKA implementation) classifiers along with the same set of features utilized in [7]. However, the SMO classifier outperformed the baseline work and achieved the best results regarding the OTE task and the sentiment polarity identification task, respectively. Although ML-based models perform well, they rely significantly on data preprocessing and intensive feature engineering.

Additionally, Deep Learning models have made significant contributions to the ABSA task. The authors in [14] proposed INSIGHT-1 at SemEval-2016. They applied a Convolutional Neural Network (CNN) model for the ABSA task on the Arabic Hotel Reviews dataset. The authors in [4], the same authors of [26], have examined the use of the Recurrent Neural Network (RNN) model to address the OTE task as well as the aspect sentiment polarity identification task. They combined the word2vec [27] word embedding along with the features presented in the previous experiment [26]. The results demonstrated that the SMO classifier outperformed the RNN model regarding performance metrics; however, the RNN was faster during the execution time. Other variations of RNN are then explored in many studies. The Bidirectional Long Short-Term Memory (BiLSTM) [28] and the Bidirectional Gated Recurrent Unit (BiGRU) were the most utilized techniques in combination with the CRF classifier. In [5], the authors utilized a BiLSTM-CRF model for the ATE task, whereas in [3], a BiGRU-CRF model was utilized. In [8], a BiLSTM-attention-LSTM-CRF model is utilized for the OTE task. As a feature representation, a combination of Continuous-Bag-of-Words (CBOW) [27] and character-level embeddings generated via CNN is utilized in [3] and [8]. The fastText [29] character-level embedding is utilized in [5]. For the aspect-based sentiment polarity classification task, the authors in [3] proposed an interactive attention network model (IAN) combined with a BiGRU. In [5], they proposed the Aspect Based-LSTM-Polarity Classification (AB-LSTM-PC) model with an aspect attention-based vector. In [12], the authors used a combination of CBOW and skip-gram character-level embeddings. They applied a Stacked Bidirectional Independent LSTM (Bi-Indy-LSTM) with a

[1] https://www.cs.waikato.ac.nz/ml/weka/

TABLE 2. Summary of related works for the aspect term extraction (ATE) and aspect sentiment classification (ASC) tasks.

Ref/Year	Discussed Task	Model	Dataset	Features	Performance			
					P(%)	R(%)	F1(%)	ACC(%)
[2]/2016	ATE	SVM *baseline*	Hotel Reviews	N-Unigrams	-	-	30.9	-
	ASC				-	-	-	76.42
[7]/2016	ATE	J48	Gaza News Posts	POS, NER, and N-Grams	81.3	82.5	81.7	-
	ASC	CRF			-	-	-	87.9
[14]/2016	ASC	CNN	Arabic Hotel Reviews	(Aspect + text) word embeddings randomly initialized	-	-	-	82.7
[4]/2018	ATE	RNN	Arabic Hotel Reviews	POS, NER, N-Grams, morphological and word features + word2vec word embedding	-	-	48	-
	ASC				-	-	-	87
[26]/2019	ATE	SVM	Arabic Hotel Reviews	POS, NER, N-Grams, morphological and word features	89.8	90	89.8	-
	ASC				-	-	-	95.4
[5]/2019	ATE	BiLSTM-CRF	Arabic Hotel Reviews	fastText char-level embedding	-	-	69.98	
	ASC	AB-LSTM-PC + Soft Attention			-	-	-	82.6
[8]/2020	ATE	BiLSTM-attention-LSTM-CRF	Arabic Hotel Reviews	CNN char-level + CBOW word-level embeddings	-	-	72.83	-
[12]/2021	ASC	Bi-Indy-LSTM + recurrent attention	Arabic Hotel Reviews	(Aspect + text) word embeddings using skip-gram and CBOW	-	-	-	87.31
[6]/2021	ATE	BiGRU	Arabic Hotel Reviews	MUSE sentence-level embeddings	-	-	93	92.82
	ASC				90.8	90.5	90.86	91.40
[3]/2021	ATE	BiGRU-CNN-CRF	Arabic Hotel Reviews	AraVec [32] word-level + CNN char-level embedding	-	-	69.44	-
	ASC	IAN-BGRU			-	-	-	83.98
[11]/2021	ASC	fine-tune Arabic BERT	HAAD [33]	(Aspect + text) Arabic BERT word embeddings	-	-	-	73
			Gaza News Posts		-	-	-	85.73
			Arabic Hotel Reviews		-	-	-	89.51
[9]/2022	ATE	fine-tune AraBERT-BiGRU-CRF	Gaza News Posts	AraBERTv0.1 word embedding	87.7	88.5	88.1	-
[10]/2022	ATE	fine-tune AraBERT-BiLSTM-CRF	Arabic Hotel Reviews	AraBERTv0.2 word embedding + Flair string embedding	-	-	79.9	-
[13]/2022	ASC	fine-tune AraBERT	HAAD	(Aspect + text) word embeddings using (AraBERT + Arabic BERT) along with Seq2Seq dialect normalization	-	-	-	74.85
			Arabic Hotel Reviews		-	-	-	84.65

position-weighting and an attention mechanism combined with a GRUs layer for the aspect sentiment classification task.

An improvement in the performance concerning the previous experiments was observed after utilizing character-level embeddings and attention mechanisms.

Consequently, pre-trained language models based on transformer architecture [18] have achieved remarkable success in Arabic ABSA. The authors in [10] used a combination of AraBERT and Flair embeddings for aspect extraction. They compared attaching a BiLSTM-CRF and BiGRU-CRF layer on top of the stacked embeddings. The results showed that fine-tuning AraBERT with a BiLSTM-CRF layer achieved better performance. In [11], the authors fine-tuned the pre-trained language model Arabic BERT [30] for the aspect sentiment polarity classification task. They used a sentence-pair classification approach where the aspect term is paired with the input sentence as an auxiliary sentence. In [13], the authors combined AraBERT and Arabic BERT and fine-tuned the generated Sequence-to-Sequence (Seq2Seq) model for the aspect term polarity task. In [6], the authors investigated the use of a Multilingual Universal Sentence Encoder (MUSE) [31] with a pooled BiGRU for the aspect extraction and aspect polarity classification tasks. The model achieved a state-of-the-art result, indicating the superiority of the pre-trained language models. TABLE 2 summarizes the work on both ATE and ASC tasks, respectively.

B. END-TO-END ASPECT-BASED SENTIMENT ANALYSIS (E2E-ABSA)

Despite the efficiency of the previously discussed single-task approaches, they lack the practical interaction required to fully perform the ABSA task. Additionally, the work involved in ASC relies on the aspect term as a pre-identified feature of the model in conjunction with the input sentence, which is not the case in real-world scenarios. To overcome these limitations, various studies on English ABSA have developed models that can perform these subtasks jointly, either through a hierarchical approach [34] or an End-to-End approach [17, 35, 36, 15]. The following studies correspond to the English E2E-ABSA.

In [34], the authors propose a hierarchical multi-task learning framework. The framework consists of ATE and Aspect Sentiment Detection modules with a sentiment lexicon and an attention mechanism. The BiLSTM-CRF layer is utilized for predicting the final target sentiment label. The authors also utilize attaching BERT embeddings, which eventually boost the performance. In [35], the authors utilized a BERT-SAN model where the BERT model is fine-tuned along with a neural classification layer and a Self-Attention Network (SAN) for the unified E2E-ABSA. In [36, 15], the authors applied two stacked BiLSTM layers for a unified E2E-ABSA. The GloVe [37] embeddings and target-position information are used as features. In [17], the authors propose a CasNSA model that consists of several modules: a contextual semantic representation module, a target boundary recognizer, and a sentiment polarity identifier. The model was tested on four different datasets, and the highest F1-score achieved was on the SemEval-2014 dataset.

Joining the Aspect Term Extraction and Aspect Sentiment Classification tasks together into a single task can achieve the required dependency and relatedness between aspect terms and their sentiment polarities. However, dealing with both tasks simultaneously requires a model capable of processing a large search space and can converge to achieve good results with a fast execution time.

III. THE PROPOSED METHODOLOGY

In contrast to previous studies, our proposed model neither relies on intensive feature engineering nor a pre-identified aspect term but rather on the relationships between words, contextualized information, and tag dependencies.

We formulated the subtasks of ABSA, Aspect Term Extraction and Aspect Sentiment Classification tasks as an End-to-End sequence labeling task via a unified tagging schema. The proposed model was evaluated by utilizing the pre-trained language model AraBERT along with different classification techniques.

After preparing the data for the desired task, the AraBERT model is utilized to extract the required features in two approaches: first, as a feature-based model with no weights modified during the training phase. Second, as a fine-tuned model. The extracted feature vectors are then processed with a fully connected neural network layer (Dense) to reduce the dimensionality and interpret the data for the final classification stage. Two classification algorithms were applied at this stage: a multi-layer perceptron with a softmax activation function and a liner-chain CRF. FIGURE 1 shows the overall architecture of the proposed model. More details regarding the architecture's components will be explained in the next subsections.

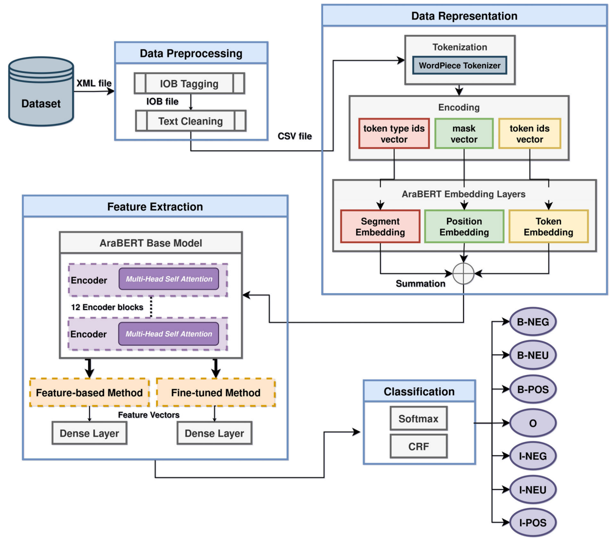


FIGURE 1. The architecture of the proposed model.

TABLE 3. Description of the arabic hotel reviews dataset.

Dataset	Sentence			Aspect Terms		
	Train	Test	Total	Train	Test	Total
Sentence-level	4802	1227	6029	96,246	23,856	120,102

A. DATASET

The Arabic Hotel Reviews dataset [2] from the SemEval-2016 workshop was utilized to evaluate the proposed model. It contains reviews written in Modern Standard Arabic and Dialectal Arabic as well. The dataset was annotated on two levels: text-level annotation and sentence-level annotation. In this research, only the sentence-level annotation is targeted.

As displayed in FIGURE 2, reviews are written in XML format, with each review containing multiple sentences, each of which contains a text with several attributes (target, category, and polarity). According to those attributes, the aspect terms and their corresponding sentiment polarities are extracted. TABLE 3 depicts the distribution of sentences and aspect terms in the dataset.

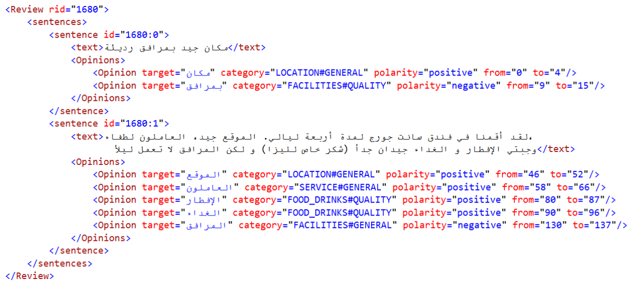

```
<Review rid="1680">
    <sentences>
        <sentence id="1680:0">
            <text>مكان جيد بمرافق رديئة</text>
            <Opinions>
                <Opinion target="مكان" category="LOCATION#GENERAL" polarity="positive" from="0" to="4"/>
                <Opinion target="بمرافق" category="FACILITIES#QUALITY" polarity="negative" from="9" to="15"/>
            </Opinions>
        </sentence>
        <sentence id="1680:1">
            <text>لقد أقمنا في فندق سانت جورج لمدة أربعة ليالي. الموقع جيد، العاملون لطفاء،
                 وجبتي الإفطار و الغداء جيدان جداً (شكر خاص لليزا) و لكن المرافق لا تعمل ليلاً</text>
            <Opinions>
                <Opinion target="الموقع" category="LOCATION#GENERAL" polarity="positive" from="46" to="52"/>
                <Opinion target="العاملون" category="SERVICE#GENERAL" polarity="positive" from="58" to="66"/>
                <Opinion target="الإفطار" category="FOOD_DRINKS#QUALITY" polarity="positive" from="80" to="87"/>
                <Opinion target="الغداء" category="FOOD_DRINKS#QUALITY" polarity="positive" from="90" to="96"/>
                <Opinion target="المرافق" category="FACILITIES#GENERAL" polarity="negative" from="130" to="137"/>
            </Opinions>
        </sentence>
    </sentences>
</Review>
```

FIGURE 2. Snapshot from the sentence-level annotated arabic hotel reviews dataset.

B. DATA PREPROCESSING

1) TEXT CLEANING AND IOB TAGGING

The preprocessing stage went through the following procedures to prepare the dataset to be compatible with our experiment: to begin, the original XML file of the dataset is transformed to IOB file format. As presented in Algorithm 1, based on the 'target' and 'polarity' attributes, each sentence is divided into a list of words, and each word is assigned an appropriate label from the label list [$B-NEG$, $B-NEU$, $B-POS$, $I-NEG$, $I-NEU$, $I-POS$, O].

Except for O, each label consists of two parts: the target's boundary and the sentiment polarity. If the word is not included in the 'target' attribute, the label O is assigned. If the 'target' attribute consists of only one word, the label $B-POS$, $B-NEG$ or $B-NEU$ is assigned based on the 'polarity' attribute. Finally, if the 'target' attribute consists of several words, the label $B-$ is assigned to the first word, followed by $I-$ to the remaining words combined with *positive*, *negative* or *neutral* polarity.

The IOB file is then converted to a CSV file with some text cleaning, which includes removing punctuation, digits, and any non-Arabic letters, normalizing Hamza (أ, إ, آ to ا) and ta-marbuta (ة to ه) and normalizing letters with diacritics ("أَكَلَ"

to “اكل” i.e.,“eat”). Word elongation is also removed to avoid any duplication of letters (“جميييل” i.e., “niiiice” to “جميل” “nice”) Algorithm 2 summarizes the steps involved.

Algorithm 1 XML to IOB

```
Input: dataset file in XML format
Output: dataset file in iob format
Sentences ← []
For <sentence> ∈ xml file do
    Text, [target, from, to, polarity] ← extract the text and its corresponding attributes
    Sentences ← Sentences + {‘text’: Text, ’attributes’: [target, from, to, polarity]}
End
For sentence ∈ Sentences do
    Dict ← { } (Create a dictionary for each sentence)
    For attribute ∈ sentence [“attributes” ], do
        Remove attribute with ‘NULL’ target
        Update the dictionary with the target’s position starting index as a key
        Dict[from] ← [target, from, to, polarity]
    End
    Last_end ← 0 (pointer)
    For key ∈ Sort(Dict) do
        target, from_, to_, polarity ← Dict[key]
        Extract the text that precedes the first target
        Text_with_Os ← text [last_end: from_]
        Update the pointer to point to the remaining text
        Last_end ← to_
        If the current target consists of only one word:
        t ← t + target + “B-” + polarity
        Else
        Do the same for the first word, then
        change “B-” to “I-” for the remaining words.
    End
    Concatenate the text that precedes the first target with t
    S ← text_with_Os + t
    If the current target is the last target that appears in the sentence:
    S ← S + Text[to_: ]
    Replace the white spaces in S with “O” followed by a new line
    Write the sentence S to the .iob output file.
End
Return the.iob file
```

TABLE 4 shows a distribution of classes in the dataset. A sample from the dataset after preprocessing is presented in TABLE 5.

Algorithm 2 IOB to CSV

```
Input: .iob file output from Algorithm 1
Output: dataset file in .csv format
word_list, label_list ← [], []
idx_list ← [] (keeps track of words within the same sentence)
idx ← 0
For line ∈ . iob file, do
    If the line is NOT empty line
    word, label ← Extract the word and its label
    word ← Remove_punctuation (word)
    word ← Remove_diacritics(word)
    word ← Remove_elongation(word)
    word ← Remove_non_arabic_letters_digits(word)
    word ← Normalization(word)
    word_list ← word_list + word
    label_list ← label_list + label
    idx_list ← idx_list + idx
    Else
    idx ← idx + 1 (new sentence)
End
csv_file ← dataframe([idx_list, word_list, label_list])
Join words with the same idx as one sentence in a new column
Join labels with the same idx as one label sequence in a new column
Drop the remaining columns
Return the .csv file
```

TABLE 4. Distribution of classes in arabic hotel reviews dataset after preprocessing.

Tag	O	B-POS	B-NEG	B-NEU	I-POS	I-NEG	I-NEU
Train	79081	5846	3151	662	1130	629	85
Test	19370	1430	786	163	274	194	26
Total	98451	7276	3937	825	1404	823	111

TABLE 5. Samples of the dataset after preprocessing.

Sentence	Label Sequence
كانت **الغرفه** ممتازه وكذلك **الموظفين** i.e., **The room** was excellent and so were **the staff**	**B-POS** O O **B-POS** O
فريق العمل الودود والمتعاون على الاطلاق i.e., Absolutely friendly and cooperative **staff team**	O O O O **I-POS B-POS**
موقع الاوتيل جيد **والاكل** جيد **والحلويات** مميزه i.e., **The hotel’s location** is good, **the food** is good, and **the desserts** are distinctive	O **B-POS** O **B-NEU** O **I-NEU B-NEU**

C. DATA REPRESENTATION

1) TOKENIZATION AND ENCODING

Before feeding the input sentence to the model, it must be tokenized and encoded in a specified form. This stage makes use of BERT’s WordPiece tokenizer. It divides the token into subtokens of known and unknown words; for example, the word “سيئه” i.e., “bad”, could be tokenized into two subwords, ـه## and سيئـ; this suffix in the Arabic language is similar to the "ing"," ed", "s" and others, but with a different meaning. For instance, the word "Learning" could be tokenized into "Learn" and "##ing".

This tokenization method eliminates the out-of-vocabulary (OOV) problem, hence resolving the complexity and ambiguity of the Arabic Language without the need for extensive preprocessing (stemming or lemmatization). However, this strategy may result in a mismatch between the input tokens and the labels in the dataset. As illustrated in FIGURE 3, the label sequence has only six elements, whereas the tokenized sentence has nine tokens. To deal with this problem, each subtoken beginning with '# #' will be ignored, leaving only the known part of the token and its corresponding label to be fed to the model.

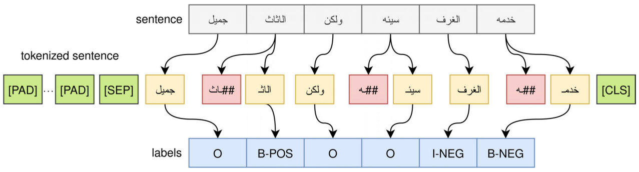


FIGURE 3. Example that clarifies the problem with the wordpiece tokenizer.

Furthermore, BERT's tokenizer attaches special tokens to each sentence. [CLS] and [SEP] tokens are attached at the beginning and end of the sentence, respectively. It also pads the input sentences to the same length by appending the special token [PAD] at the end. Tokens are then encoded into three vectors of integer values: a vector of token ids utilizing the BERT's vocabulary, a vector of mask values, and a vector of token type ids. These vectors are utilized as inputs to the BERT embedding layers to generate the initial representations.

D. FEATURE EXTRACTION

1) PRE-TRAINED ARABERT MODEL

AraBERT is a pre-trained Arabic language model based on the BERT language model. It embeds a sequence of words into a sequence of contextualized vectors with specific dimensions. The BERT model has three Embedding Layers:

- Token Embedding, which encodes the meaning of each word utilizing an input ids vector.
- Segment Embedding, which encodes the sentence position utilizing a mask-encoded vector.
- Position Embedding, which encodes the word's position in the input sentence utilizing a token type ids vector.

Those embeddings are concatenated, providing context-independent word embeddings. To generate the contextualized embeddings, the self-attention mechanism of the Transformer's Encoder component is utilized [18]. In which each input element is connected to every other input element, and the weightings (attention scores) between them are dynamically calculated based on that connection.

As illustrated in FIGURE 4, the initial embeddings are utilized in combination with randomly initialized weight matrices, Query (Q_w), Key (K_w), and Value (V_w), to form Q, K, and V matrices, which are used to calculate attention scores as indicated in (1) [18].

At each time step, a dot product is applied to calculate the similarity between a target word (Query word) and every other word in the sequence (Key words). A division and a softmax function are used to normalize the scores calculated; d_k denotes the dimension of the K matrix, which is the same as the embedding dimension (768 for BERT-base). The normalized scores are used to weight the V matrix, resulting in a weighted feature vector for each input token.

$$Attention\,(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The AraBERT-base model comprises 12 attention heads included within each of its 12 Encoder blocks. The outputs of each attention head are combined to generate the final contextualized embeddings, as shown in FIGURE 5.

Two experiments were conducted: first, utilizing AraBERT as a feature-based model, and second, fine-tuning its parameters within a Deep Learning model.

2) DENSE LAYER

We implement a Multi-Layer Perceptron (MLP) that comprises two Dense hidden layers with a Rectified Linear Unit (ReLU) [38] activation function to reduce the input dimensionality and speed up the training process.

The embeddings from the final Encoder block, referred to as the last hidden state, are used as inputs to the ReLU activation function defined by (2) as follows:

$$D = ReLU(WH + b) = Max((WH + b),\ 0) \tag{2}$$

where H is the last hidden state matrix of dimensions: sequence length $x768$, W is a trainable weight matrix, and b is a bias term. FIGURE 6 illustrates the process of MLP with ReLU Dense layers.

E. CLASSIFICATION

1) SOFTMAX

For the classification stage, we initially investigated utilizing a fully connected layer with a softmax activation function to predict a tag for each input token. Softmax [24] is a function that normalizes the output of a neural network to a probability distribution over the predicted output classes as follows:

$$\hat{y} = softmax\,(WD + b) = \frac{e^{d_i}}{\sum_{j=0}^{C} e^{d_j}} \tag{3}$$

where $\hat{y}$ denotes the matrix of predicted probabilities, d_i denotes the hidden representation of a token with respect to class i, while d_j denotes the representation with respect to all classes C.

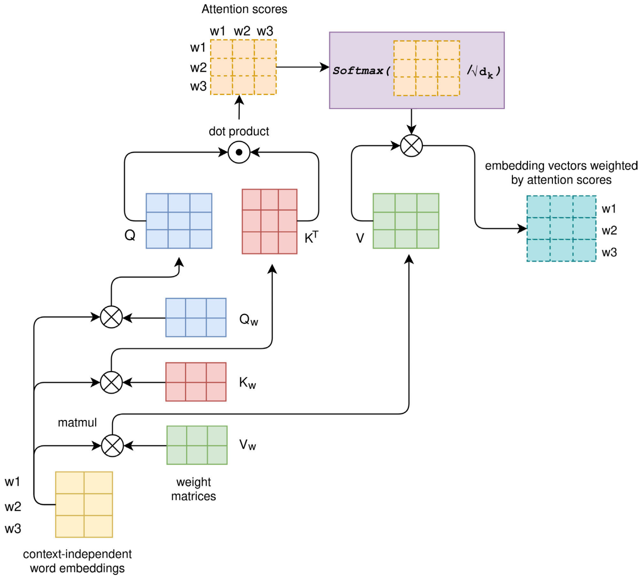


FIGURE 4. Self-attention mechanism adapted from [18]. Words in the sequence with a 768-dimensional vector are represented by w1, w2, and w3. matrices Q_W, K_W, and V_W are of size: number of words in the sequence x 768.

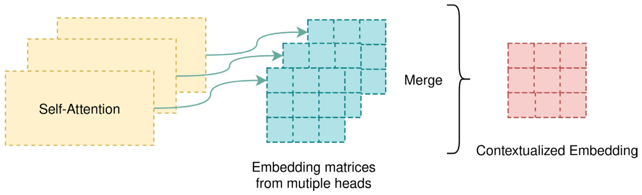


FIGURE 5. Multi-head attention mechanism. each encoder block contains 12 attention heads, each with its own self-attention.

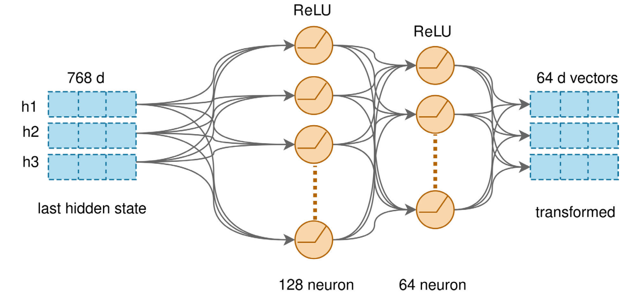


FIGURE 6. ReLU activation function is used in a multi-layer perceptron with two dense hidden layers of 128 and 64 neurons, respectively.

Because the proposed E2E-ABSA is a multi-class classification task, the model is trained to minimize the categorical cross-entropy [39] between predicted and true results as follows:

$$L(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{K=1}^{C} y_i^k \log(\hat{y}_i^k) \tag{4}$$

where y_i^k indicates the i^{th} true label which is a one-hot encoded vector of class k; $\hat{y}_i^k$ indicates the i^{th} predicted probability of class k, where N is the number of samples in the training dataset.

2) CONDITIONAL RANDOM FIELDS

Instead of modeling tagging decisions independently, we can model them jointly using conditional Random Fields (CRF) [23]. The linear-chain CRF is a discriminative model for predicting the probability of a sequence of labels given a sequence of observations while taking the labels' dependencies into account.

We experimented with utilizing the liner-chain CRF as a classification layer for our proposed model.

For a sequence input $X = \{x_1, x_2, \ldots, x_n\}$, we consider the matrix P to be the emission scores outputted by AraBERT hidden states after processing. Its size is n x k where k is the number of distinct tags/labels and $P_{i,j}$ represents the score of the tag j given the observed word i.

For a sequence of labels $Y = \{y_1, y_2, \ldots, y_n\}$, the sequence score is defined by (5) [40] as follows:

$$S(X, Y) = \sum_{i=0}^{n} A_{y_i,y_{i+1}} + \sum\nolimits_{i=1}^{n} P_{i,y_i} \tag{5}$$

where A is the transition scores matrix learned during the training and $A_{i,j}$ represents a transition score from tag i to tag j, responsible for setting constraints on the tags to ensure the tag dependencies.

After calculating the sequence score, the softmax function is applied to calculate the likelihood probability for the correct tag sequence Y over all the possible tag sequences $\hat{y}$. It is defined by (6) [40] as follows:

$$P(Y|X) = \frac{e^{s(X,Y)}}{\sum_{Y'\hat{\in}Y} e^{s(X,Y')}} \tag{6}$$

Our models' parameters are trained to maximize the log-likelihood of the correct tag sequence by minimizing the negative log-likelihood defined by (7) [40].

$$-\,log\,(P\,(Y|X)) = -[S\,(X, Y) -\ log \sum\nolimits_{Y'\hat{\in}Y} e^{s(X,Y')}] \tag{7}$$

For prediction, the Viterbi algorithm is used to find the tag sequence with the highest score Y^*:

$$Y^* = Argmax_{Y'\in\hat{Y}} S\left(X, Y'\right) \tag{8}$$

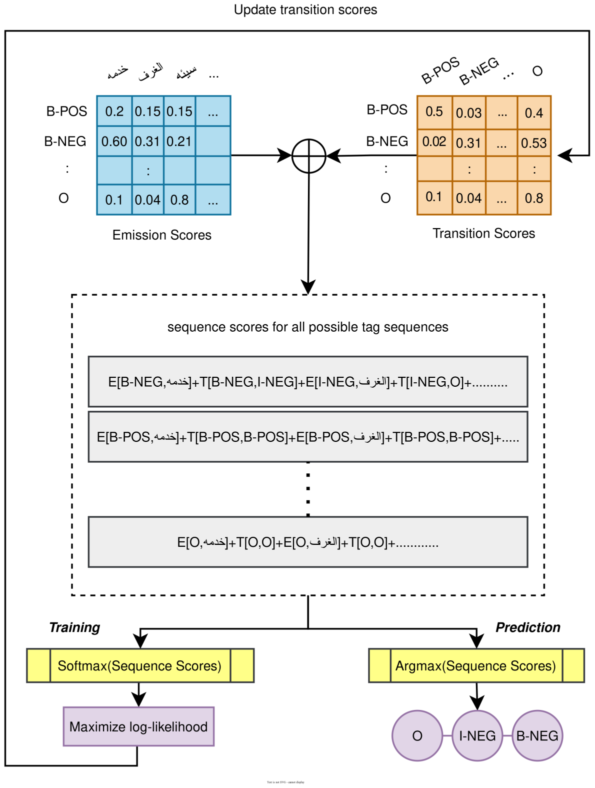
The operational flow within CRF is shown in FIGURE 7.

IV. EXPERIMENTAL RESULTS AND EVALUATION

This section represents the experimental settings, evaluation metrics along with results and discussion of the conducted experiments.

A. EXPERIMENTAL SETTINGS

The base version of the pre-trained language model AraBERT is utilized during experiments.The AraBERT-base model was released in four versions: AraBERTv0.1, AraBERTv1, AraBERTv0.2, and AraBERTv2. We utilized AraBERTv0.2.


Update transition scores
Emission Scores
Transition Scores
sequence scores for all possible tag sequences
E[B-NEG,خدمه]+T[B-NEG,I-NEG]+E[I-NEG,الغرف]+T[I-NEG,O]+..........
E[B-POS,خدمه]+T[B-POS,B-POS]+E[B-POS,الغرف]+T[B-POS,B-POS]+.....
E[O,خدمه]+T[O,O]+E[O,الغرف]+T[O,O]+............
Training
Prediction
Softmax(Sequence Scores)
Argmax(Sequence Scores)
Maximize log-likelihood
O
I-NEG
B-NEG

	خدمه	الغرف	سيئه	...
B-POS	0.2	0.15	0.15	...
B-NEG	0.60	0.31	0.21	
:		:		
O	0.1	0.04	0.8	...

	B-POS	B-NEG	...	O
B-POS	0.5	0.03	...	0.4
B-NEG	0.02	0.31	...	0.53
:		:		:
O	0.1	0.04	...	0.8

FIGURE 7. The operations flow within a liner-chain CRF.

The model is available on the HuggingFace model page under the aubmindlab name[2].

For building our functional network, we used the Keras[3] API, which is built on top of the TensorFlow Python package. The model was trained for 5 epochs with a batch of 32 input samples, each padded to a maximum length of 64 characters. Each input token is encoded into a 768-dimensional vector. The Adam optimizer [41] is utilized with a learning rate 5e-5. The model comprises two Dense layers with 128 and 64 neurons, respectively. Other hyper-parameters are the same as those in the pre-trained AraBERTv0.2 implementation. All experiments were run on Google Colaboratory with a Tesla P100 GPU, 25 GB RAM, and 167 GB Disk Space.

B. EVALUATION METRICS

All experiments were evaluated with four versions of k-fold cross-validation [42]: 3, 5, 10, and 15. The entire dataset (train and test) is shuffled and divided into k smaller sets; for each k, the model is trained using k-1 of the folds as training data, then the model is validated on the test part. This process is repeated k times with a new model and different testing folds in each case. The performance measure is then the average of the values computed in the loop to ensure the model's resistance to overfitting.

The following metrics [43], [44] will be used to evaluate our proposed model, including Precision (P), Recall (R), F1 score, Accuracy (ACC), Area Under Curve (AUC), and Area Under Precision-Recall (AUPR), which are defined by (9–15) as follows:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{10}$$

$$F1\ score = \frac{2\,(Precision * Recall\ )}{Precision + Recall} \tag{11}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

$$AUC = \int_0^1 TPR\ d(FPR) \tag{14}$$

$$AUPR = \sum_n (Recall_n - Recall_{n-1}) Precision_n \tag{15}$$

The precision (9) is the ratio of correctly predicted values for a class to all of its predictions, while the Recall (10), or the True Positive Rate (TPR), is the ratio of correctly predicted values for a class to the number of actual samples of that class in the dataset.

F1 score (11) is the harmonic average of Precision and Recall and is used mainly for evaluating sequence labeling tasks [9], [13], [17], [34].

The Accuracy (12) is obtained by dividing the correctly classified labels by the total number of labels in the dataset. The Receiver Operating Characteristic (ROC) [44] curve is summarized by AUC (14) based on the TPR and False Positive Rate (FPR) at different classification thresholds. The higher the AUC, the better the model's performance in distinguishing between positive and negative classes.

The Precision-Recall (PR) curve is summarized by AUPR [44], defined by (15), as the weighted mean of precisions achieved at each threshold n, where the weights are the increase in Recall from the previous threshold $n - 1$. We calculate the True Positive (TP), True Negative(TN), False Positive (FP), and False Negative (FN) for each tag independently. For example, in terms of the B-POS tag:

- TP is the number of samples predicted as B-POS, and its actual label is also B-POS.
- FP is the number of samples predicted as B-POS, but its actual label is something else.
- FN is the number of B-POS samples but predicted as something else.
- TN is the number of samples predicted as not B-POS, and its actual label is also not B-POS.

The evaluation scores are evaluated token-wise [45], then an average value is calculated as the proposed model's evaluation score (macro-average [43]). Furthermore, BERT's tokenizer generates new labels that are not defined in the dataset, which are created by [CLS], [SEP], and [PAD] tokens discussed earlier. Those labels are ignored since

[2] https://huggingface.co/aubmindlab/bert-base-arabertv02
[3] https://keras.io/api/

TABLE 6. Feature-Based AraBERT-Softmax evaluation results using different K-folds.

#Fold	3					5					10					15				
Label / Metric	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR
B-POS	0.52	0.15	0.23	0.94	0.33	0.52	0.25	0.33	0.96	0.39	0.55	0.25	0.34	0.96	0.41	0.54	0.32	0.40	0.97	0.46
B-NEG	0.35	0.02	0.04	0.93	0.16	0.36	0.07	0.11	0.95	0.20	0.45	0.06	0.11	0.95	0.21	0.45	0.11	0.17	0.96	0.27
B-NEU	0.00	0.00	0.00	0.90	0.01	0.02	0.00	0.00	0.90	0.02	0.00	0.00	0.00	0.93	0.03	0.00	0.00	0.00	0.93	0.03
O	0.88	0.96	0.92	0.99	0.97	0.89	0.96	0.93	0.99	0.97	0.89	0.97	0.93	0.99	0.97	0.90	0.97	0.93	0.99	0.98
I-POS	0.17	0.00	0.00	0.92	0.05	0.09	0.00	0.00	0.93	0.08	0.06	0.00	0.00	0.93	0.06	0.17	0.01	0.02	0.94	0.09
I-NEG	0.00	0.00	0.00	0.89	0.02	0.07	0.00	0.00	0.92	0.04	0.07	0.00	0.00	0.92	0.03	0.07	0.00	0.00	0.93	0.05
I-NEU	0.00	0.00	0.00	0.84	0.00	0.00	0.00	0.00	0.86	0.00	0.00	0.00	0.00	0.87	0.00	0.03	0.00	0.00	0.90	0.00
Macro-average	27.40	16.12	16.95	91.57	22.01	27.87	18.28	19.65	93.00	24.30	28.79	18.26	19.68	93.57	24.42	**29.91**	**20.09**	**21.65**	**94.57**	**26.85**

TABLE 7. Feature-Based AraBERT-CRF evaluation results using different K-folds.

#Fold	3					5					10					15				
Label / Metric	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR
B-POS	0.52	0.23	0.31	0.95	0.34	0.50	0.24	0.32	0.95	0.35	0.53	0.24	0.33	0.96	0.39	0.51	0.27	0.35	0.96	0.39
B-NEG	0.29	0.03	0.05	0.91	0.14	0.42	0.05	0.09	0.94	0.19	0.39	0.07	0.11	0.95	0.22	0.36	0.08	0.13	0.95	0.20
B-NEU	0.00	0.00	0.00	0.89	0.02	0.00	0.00	0.00	0.88	0.01	0.00	0.00	0.00	0.91	0.02	0.02	0.00	0.00	0.90	0.02
O	0.89	0.96	0.92	0.99	0.96	0.89	0.96	0.92	0.99	0.97	0.89	0.96	0.93	0.99	0.97	0.89	0.96	0.92	0.99	0.97
I-POS	0.14	0.00	0.01	0.86	0.03	0.21	0.01	0.01	0.92	0.06	0.16	0.01	0.01	0.91	0.06	0.27	0.01	0.01	0.93	0.08
I-NEG	0.03	0.00	0.00	0.91	0.02	0.02	0.00	0.00	0.88	0.02	0.05	0.01	0.01	0.90	0.03	0.00	0.00	0.00	0.92	0.04
I-NEU	0.00	0.00	0.00	0.84	0.00	0.00	0.00	0.00	0.85	0.00	0.00	0.00	0.00	0.86	0.00	0.00	0.00	0.00	0.86	0.00
Macro-average	26.71	17.39	18.43	90.71	21.57	29.10	17.97	19.26	91.67	22.85	28.83	18.37	19.87	92.57	24.14	29.31	18.75	20.20	93.00	24.30

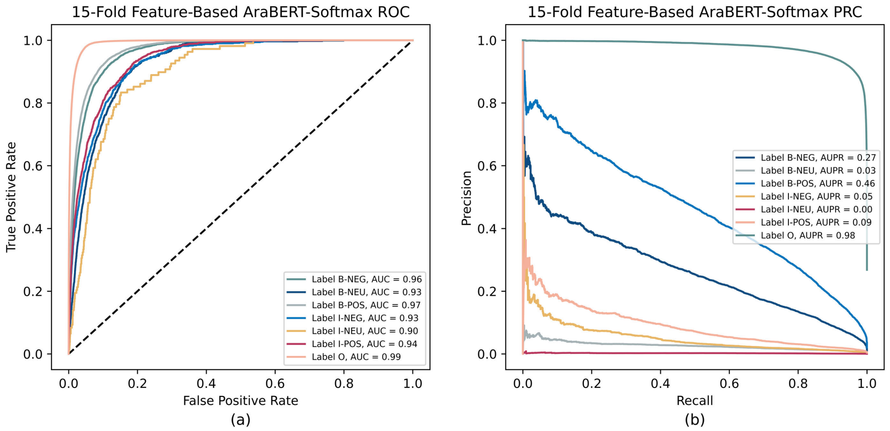


FIGURE 8. (a) ROC curve and (b) PR curve for 15-Fold Feature-based AraBERT-softmax model in a One-vs-Rest approach.

they are irrelevant to the actual inference. Therefore, only the seven entities specified by B-NEG, B-POS, B-NEU, I-POS, I-NEG, I-NEU, and O are reported for the evaluation metrics.

C. RESULTS AND DISCUSSION

This section presents the experiments that were carried out in this study, along with an analysis of the obtained results.

1) EXPERIMENT 1: FEATURE-BASED METHOD

In this experiment, we investigated the impact of utilizing the pre-trained AraBERT model as a feature-based model while keeping its parameters fixed during the training process.

As stated in TABLE 6 and TABLE 7, the performance of the feature-based AraBERT model on our E2E-ABSA task is not particularly outstanding in all folds when using either the CRF or the MLP with softmax as classifiers. The model does not appear to learn the required contextualized features.

This behavior is expected because the AraBERT model was pre-trained on two specific tasks: Next Sentence Prediction and Masked Language Modeling [21]. The representation of the model is obviously insufficient for the downstream task, and task-specific fine-tuning is required to take advantage of AraBERT's capabilities in enhancing performance. However, the best performance was achieved by the 15-fold AraBERT-softmax model with a Precision of 29.91%, Recall of 20.09%, F1 score of 21.65%, AUC of 94.75%, and AUPR of 26.85%.

Additionally, the average-AUC value seems high compared to the other model performance metrics. This is often true for highly imbalanced datasets. As illustrated in FIGURE 8(a), the ROC curve has two lines: one for how often the model correctly identifies positive cases (TPR) and another for how often it mistakenly identifies negative cases as positive (FPR). However, the false positive rate could be pulled down due to the large number of true negatives, resulting in a high-pointed ROC curve.

In our proposed feature-based model, it is apparent that the model is confusing tags $B-POS$, $B-NEG$, $B-NEU$, $I-POS$, $I-NEG$, and $I-NEU$ with tag O, and in some instances, it predicts tag O more often than the correct tag (current tag in a one-vs-rest). This implies that the model

TABLE 8. Fine-Tuned AraBERT-Softmax evaluation results using different K-folds.

#Fold	3					5					10					15				
Label / Metric	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR
B-POS	0.79	0.79	0.79	1.00	0.88	0.89	0.90	0.90	1.00	0.96	0.94	0.94	0.94	1.00	0.99	0.96	0.97	0.96	1.00	0.99
B-NEG	0.78	0.86	0.81	1.00	0.88	0.90	0.91	0.90	1.00	0.96	0.95	0.94	0.94	1.00	0.98	0.96	0.97	0.97	1.00	0.99
B-NEU	0.47	0.46	0.45	0.99	0.47	0.75	0.69	0.71	1.00	0.81	0.86	0.83	0.84	1.00	0.93	0.90	0.88	0.89	1.00	0.96
O	0.98	0.97	0.98	1.00	1.00	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00
I-POS	0.69	0.74	0.71	1.00	0.77	0.83	0.84	0.83	1.00	0.90	0.91	0.91	0.91	1.00	0.96	0.93	0.96	0.94	1.00	0.97
I-NEG	0.71	0.83	0.76	1.00	0.82	0.85	0.91	0.88	1.00	0.94	0.93	0.92	0.92	1.00	0.97	0.93	0.97	0.95	1.00	0.99
I-NEU	0.42	0.35	0.33	1.00	0.37	0.64	0.61	0.61	1.00	0.74	0.86	0.80	0.81	1.00	0.90	0.91	0.88	0.89	1.00	0.96
Macro-average	69.13	71.50	69.23	99.85	74.14	83.53	83.54	83.19	1.00	90.14	91.93	90.50	90.79	1.00	96.14	94.14	94.52	94.18	1.00	**98.00**

TABLE 9. Fine-Tuned AraBERT-CRF evaluation results using different k-folds.

#Fold	3					5					10					15				
Label / Metric	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR	P	R	F1	AUC	AUPR
B-POS	0.84	0.87	0.85	0.99	0.91	0.90	0.92	0.91	1.00	0.96	0.95	0.95	0.95	1.00	0.97	0.96	0.97	0.97	1.00	0.99
B-NEG	0.86	0.88	0.87	1.00	0.91	0.90	0.92	0.91	1.00	0.95	0.96	0.94	0.95	1.00	0.96	0.97	0.97	0.97	1.00	0.98
B-NEU	0.69	0.63	0.66	0.99	0.74	0.79	0.71	0.74	0.99	0.82	0.89	0.87	0.88	1.00	0.94	0.92	0.90	0.91	1.00	0.95
O	0.98	0.98	0.98	1.00	1.00	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
I-POS	0.79	0.79	0.78	0.99	0.80	0.89	0.84	0.86	0.99	0.89	0.92	0.92	0.92	1.00	0.94	0.93	0.96	0.95	1.00	0.97
I-NEG	0.80	0.84	0.82	1.00	0.86	0.90	0.88	0.89	1.00	0.92	0.95	0.92	0.93	1.00	0.95	0.96	0.95	0.96	1.00	0.97
I-NEU	0.57	0.55	0.55	1.00	0.67	0.78	0.71	0.73	1.00	0.81	0.89	0.85	0.87	1.00	0.94	0.94	0.92	0.92	1.00	0.95
Macro-average	79.03	79.24	78.78	99.57	84.14	87.97	85.20	86.13	99.85	90.71	93.60	92.16	92.75	1.00	95.88	**95.41**	**95.23**	**95.16**	1.00	97.37

TABLE 10. Example of model inference with softmax as a classification layer.

Sentence	مثيل	لها	ليس	الطعام	جودة	و	نظافة
Label	O	O	O	I-POS	B-POS	O	B-POS
Prediction	O	O	O	B-POS	B-POS	O	O

TABLE 11. Example of model inference with CRF as a classification layer.

Sentence	مثيل	لها	ليس	الطعام	جودة	و	نظافة
Label	O	O	O	I-POS	B-POS	O	B-POS
Prediction	O	O	O	I-POS	B-POS	O	B-POS

has asymmetric error distribution, and the ROC curve fails to explicitly show this performance difference.

Concurrently, as all tags contribute equally to the classification task, the PR curve is used instead; this metric computes a weighted average precision value for each tag independent of the predictions of other tags. As illustrated in FIGURE 8(b), the model that is considered good with ROC-AUC, performs poorly with PR curve that focuses on the positive labels (current tag) and not the true negatives.

2) EXPERIMENT 2: FINE-TUNED METHOD

In this experiment, the AraBERT model's parameters are fine-tuned during the training process.

As demonstrated in TABLE 8 and TABLE 9, results were significantly improved when the model's parameters were adjusted for our E2E-ABSA task rather than using the model as a feature-based only.

With CRF as a classifier, the best performance was achieved by 15-fold with 95.41% Precision, 95.23% Recall, 95.16% F1 score, 100% AUC, and 97.37% AUPR; similarly, using MLP with softmax as a classifier, the best performance was achieved by 15-fold with 94.14% Precision, 94.52% Recall, 94.18% F1 score, 100% AUC, and 98% AUPR.

Consequently, As illustrated in FIGURE 9(a), the data point that is close to the 1 on the TPR axe is actually the optimal threshold, which means that at this threshold, the classifier is perfectly able to distinguish between positive class (current tag in a one-vs-rest) and the negative class (rest of tags). However, as AUC excels under imbalanced settings, the results could be misleading. For instance, the performance gap between AUC and the pointwise metrics (P, R, and F1 score) regarding the 3-fold AraBERT-Softmax model, presented in TABLE 8, is significant. The AUC is 99.85%, whereas P, R, and F1 score are 69.13%, 71.50%, and 69.23%, respectively, which means that even a perfect ROC-AUC does not mean that the predictions are well-calibrated.

FIGURE 9(b) illustrates the AUPR in a one-vs-rest approach for the 15-fold fine-tuned AraBERT-CRF model, where the average-AUPR is 97.37%. As presented in TABLE 8, CRF outperformed softmax in all mentioned pointwise metrics; however, the average-AUPR for the 15-fold AraBERT-Softmax model is 98%. It should be noted that CRF employs the transition scores matrix to generate the prediction probabilities, whereas the PR curve and ROC curve depend only on the emission scores of each token independently. However, CRF still produces quite stable results.

Furthermore, we observe that maintaining boundary-sentiment consistency within the same aspect term, particularly for those with multiple words (e.g., "نظافة و جودة الطعام" i.e., "food cleanliness and quality") is difficult for the AraBERT-Softmax model. In contrast, the AraBERT-CRF model resolves this problem by employing the transition matrix component to generate predictions based on the features from both the current and previous tags. As illustrated in FIGURE 10, when using softmax, the E2E-ABSA problem becomes a token-wise classification problem, predicting a tag for each token independently of other tags in the sequence. However, this behavior can lead to errors in the overall prediction. For instance, in TABLE 10, the word نظافة "cleanliness" is misclassified as a non-aspect, ignoring its relation to the word الطعام "food" while it represents the beginning of the aspect نظافة الطعام "food cleanliness" and

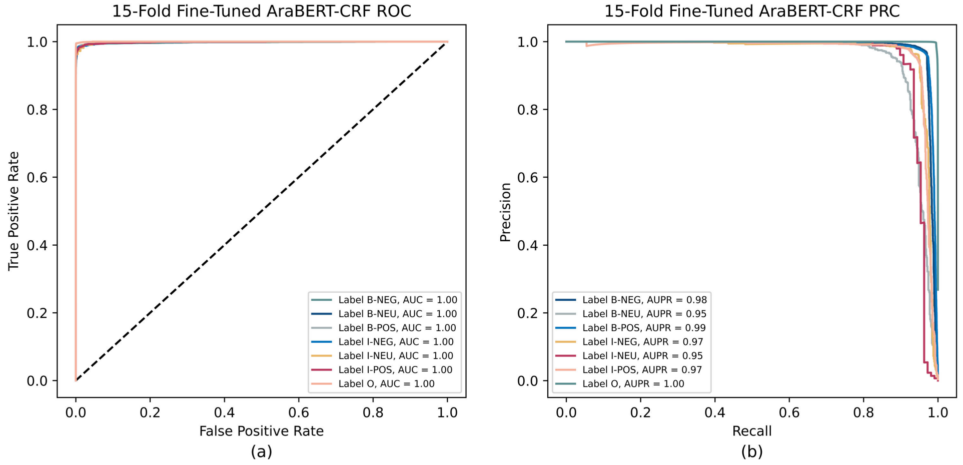


FIGURE 9. **(a) ROC curve and (b) PR curve for 15-Fold Fine-tuned AraBERT-CRF model in a One-vs-Rest approach.**

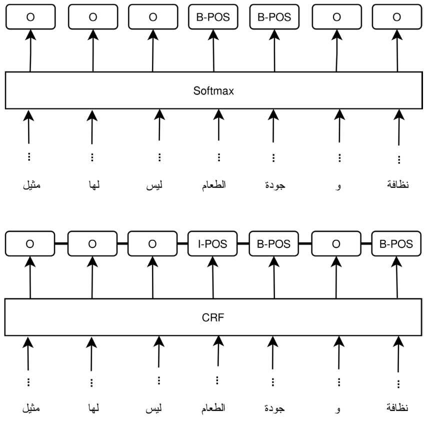


FIGURE 10. **The difference between CRF and softmax behaviour.**

should have been assigned the tag B-POS. Additionally, the tag B-POS is assigned to the aspect الطعام "food" while its actual tag is I-POS, ignoring its relation to the word جودة "quality" and the word نظافة "cleanliness" when it represents the inside of the aspect جودة الطعام "food quality" and the aspect نظافة الطعام "food cleanliness".

On the other hand, when CRF is used as a classifier, the E2E-ABSA problem turns into a sequence labeling problem. As shown in FIGURE 10, the model predicts a tag for each token while accounting for tag transition dependencies. For instance, in TABLE 11, CRF ensures that a predicted tag for a certain word is compatible with the other tags in the tag sequence, preventing errors that could occur when using softmax as the classification layer.

As demonstrated in the results above, CRF outperformed softmax by a small margin; this is likely due to the multi-head self-attention mechanism of BERT; it leads to incorporating significant information of concatenated local and global context words and learning further interactive aspect-sentiment representations, which helps the proposed model in producing improved sequence representations. The difference will be more evident when utilizing context-free embedding models.

However, the best performance was achieved by the 15-fold fine-tuned AraBERT-CRF model with 95.41% Precision, 95.23% Recall, and 95.16% F1 score. The confusion matrix of this model is presented in FIGURE 11; based on the differences between predicted and actual labels, it is demonstrated that the model can discriminate between labels effectively.

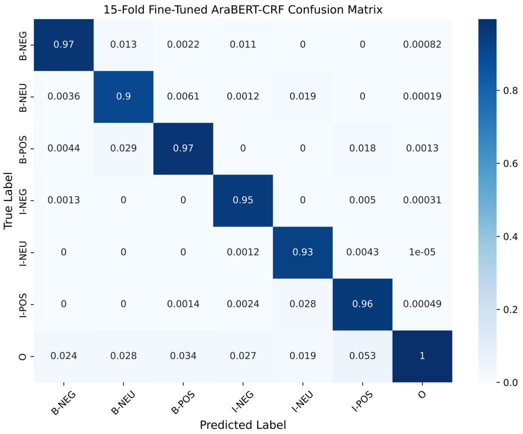


FIGURE 11. **Normalized confusion matrix of the 15-Fold Fine-Tuned AraBERT-CRF model showing all entities in the dataset.**

3) COMPARISONS WITH EXISTING STUDIES

To evaluate the proposed E2E-ABSA approach, we further processed the predicted labels to separate them into two distinct categories: aspect term labels (B, I and O) and sentiment polarity labels (positive, negative, and neutral), to be appropriate for comparisons with the previous single-task approaches. By reformulating the predictions into two separate tasks, we can maximize the evaluation scores, which results in a better classifier for each task and ultimately enhances the ABSA task.

TABLE 12. Experimental results after splitting the prediction of the E2E Fine-tuned AraBERT model for the tasks of aspect term extraction (ATE) and aspect sentiment classification (ASC).

Model	Task	#Fold	P(%)	R(%)	F1(%)	ACC(%)
Fine-tuned AraBERT-Softmax	ATE	3	82.98	90.23	86.29	-
		5	90.78	92.86	91.79	-
		10	95.25	96.29	95.75	-
		15	96.93	97.11	97.02	-
	ASC	3	75.99	80.81	78.23	90.20
		5	86.45	85.98	86.18	94.43
		10	92.01	93.73	92.85	97.12
		15	95.53	95.68	95.61	98.07
Fine-tuned AraBERT-CRF	ATE	3	91.01	89.66	90.32	-
		5	93.78	92.97	93.36	-
		10	96.89	96.10	96.49	-
		15	97.80	97.05	**97.78**	-
	ASC	3	86.69	84.24	85.39	93.64
		5	89.77	91.03	90.38	95.59
		10	94.63	94.70	94.66	97.69
		15	96.25	96.19	**96.22**	**98.34**

We utilized the Precision, Recall, and F1 score metrics to evaluate the two tasks in addition to the Accuracy for evaluating the ASC task. As shown in TABLE 12, the 15-fold fine-tuned AraBERT-CRF model achieved the best performance, with an F1 score of 97.78% for the ATE task and 96.22% for the ASC task, respectively, and an Accuracy of 98.34% for the ASC task.

Additionally, we observed that the ATE task consistently outperforms the ASC task and the E2E-ABSA task. This result indicates that the boundary information learned by the model enhances the evaluation scores of the overall E2E-ABSA task. Therefore, utilizing a model that can set constraints on the boundary information is crucial for improving the overall E2E-ABSA task, and the CRF model can be a straightforward and efficient solution.

Furthermore, we observed that k-fold cross-validation may have an impact on the model's performance. By employing k-fold cross-validation, all parts of the dataset can be used for training and testing, forcing the model to attend to a larger context and increasing the possibility of associating with relevant opinion words without overfitting. According to TABLE 12, the best results are obtained at k=15, which means that small k is likely insufficient to involve the potential opinion words and does not offer an accurate evaluation of the model's performance. FIGURE 12 illustrates the train and test error of the 15-fold AraBERT-CRF model which shows the model's resistance to overfitting. If the model overfits in a particular fold, the training error of that fold will be less than the testing error; hence, when summing/averaging the errors of all folds, a model that overfits would have a low cross-validated performance.

As a result, we compared the proposed 15-fold fine-tuned AraBERT-CRF model with several previous research works on the Arabic Hotel Reviews dataset to evaluate its quality. Consequently, the proposed model outperformed the previous single-task methods. As shown in TABLE 13, compared to BiGRU-CRF [3], BiLSTM-CRF [5], and BiLSTM-Attention-CRF [8], where they achieved 69.88%, 69.44%, and 72.83% F1 score, respectively, it achieved 28.34%, 27.9%, and 24.95% increases in the F1 score for the ATE task, respectively. Compared to BF-BiLSTM-CRF [10], our proposed model achieved 17.88% increases in F1 score for the ATE task; however, compared to MUSE-BiGRU [6], it achieved 4.78% and 6.94% absolute gains on ATE F1 score and ASC Accuracy score, respectively, indicating that a unified E2E model with an appropriate design can be more effective than the single-task approaches on the ABSA task.

While the work presented in TABLE 13 for the ASC task utilized a pre-identified aspect information, the proposed model achieved better results without aspect term annotation; it outperformed the SVM model used in [26] by 2.94% and achieved 98.34% accuracy. Compared to the Bi-Indy-LSTM-recurrent attention model in [12], our proposed model increased the Accuracy by 11.03%. The work in [11] and [13] fine-tuned Arabic-based BERT models with a single layer for

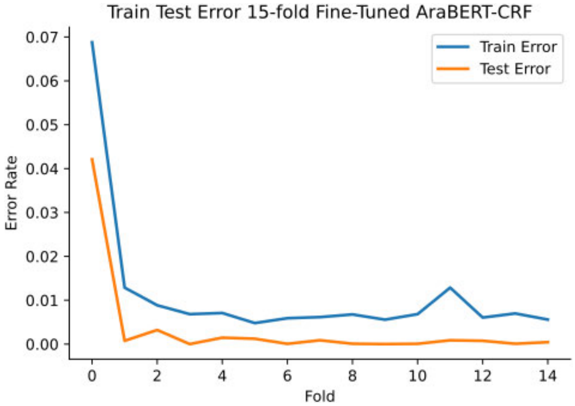


FIGURE 12. Train test error of the 15-Fold Fine-Tuned AraBERT-CRF model.

TABLE 13. Comparison results of different studies on the arabic hotel reviews dataset.

Task	Model	Acc (%)	F1 (%)
Aspect Term Extraction (ATE)	BiLSTM-CRF [5]	-	69.88
	BiGRU-CRF [3]	-	69.44
	BiLSTM-Attention-CRF [8]	-	72.83
	BF-BiLSTM-CRF [10]	-	79.9
	MUSE-BiGRU [6]	-	93
	(proposed) Fine-tuned AraBERT-Softmax	-	97.02
	(proposed) Fine-tuned AraBERT-CRF	**-**	**97.78**
Aspect Sentiment Classification (ASC)	SVM [26]	95.4	-
	Bi-Indy-LSTM + Recurrent Attention [12]	87.31	-
	fine-tuned Arabic BERT [11]	89.51	-
	fine-tuned AraBERTv0.1 [13]	84.65	-
	MUSE-BiGRU [6]	91.40	-
	(proposed) Fine-tuned AraBERT-Softmax	98.07	-
	(proposed) Fine-tuned AraBERT-CRF	**98.34**	-

token classification; they achieved an accuracy of 84.65% and 89.51%, respectively. By comparing these models to our proposed model, it is clear that our model outperformed them by 13.69% and 8.83%, respectively, on the ASC task. FIGURE 13 illustrates the comparisons with the previous single-task approaches.

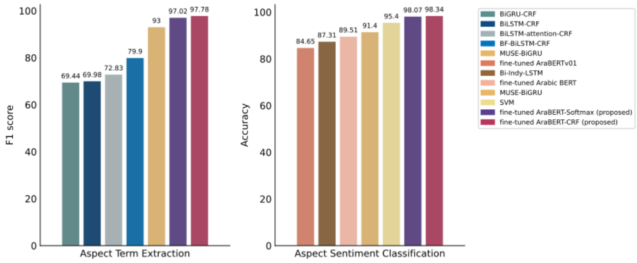


FIGURE 13. Comparisons results with the previous single-task approaches.

Furthermore, it is observed that fine-tuning the AraBERT model using MLP with softmax already outperformed the existing works without using CRF; this is likely due to AraBERT representations encoding the associations between input tokens, which significantly enhances the model performance.However, utilizing a model that sets restrictions about which tag should come before or after another helps direct the model to more accurate tag-sentiment prediction.

V. CONCLUSION AND FUTURE WORK

This study aims to investigate the importance of tackling the subtasks of ABSA, specifically Aspect Term Extraction and Aspect Sentiment Classification, simultaneously through a single model to preserve the relationships between the two subtasks, which was neglected by most related domain researchers. Unlike single-task approaches, our model creates a direct interaction between aspect terms and their sentiment polarities, dissolving the need to take the aspect features into account along with the sentence as pre-identified information to retrieve the sentiment polarity.

To address this problem, we utilized a unified tagging schema to create an End-to-End ABSA task and evaluated the proposed approach on the SemEval-2016 Arabic Hotel Reviews dataset. Several experiments were performed utilizing the AraBERT model, and results showed that the proposed fine-tuned AraBERT-CRF model outperformed the existing state-of-the-art models by achieving an overall F1 score of 95.11%.

Further processing is then made on the predictions, splitting them into ATE-labels and ASC-labels for a valid comparison. Results indicate that even after splitting the predicted labels, the model still surpassed the existing methods, achieving an F1 score of 97.78% for the ATE and an accuracy of 98.34% for the ASC.

Although the unified tagging schema solved the error propagation problem, it suffers from the large search space and requires a model capable of dealing with such a problem.

For future work, we plan to explore other subtasks of ABSA, aspect category detection and aspect category sentiment classification. We may also utilize the triplet extraction technique, which is concerned with extracting the target, opinionated word, and their corresponding sentiment polarity in one model. Additionally, other Deep Learning techniques, different embeddings, and datasets could be evaluated via the unified approach to assessing its impact on the task of ABSA.

REFERENCES

[1] R. Obiedat, D. Al-Darras, E. Alzaghoul, and O. Harfoushi, "Arabic aspect-based sentiment analysis: A systematic literature review," *IEEE Access*, vol. 9, pp. 152628–152645, 2021.

[2] M. Al-Smadi, O. Qwasmeh, B. Talafha, M. Al-Ayyoub, Y. Jararweh, and E. Benkhelifa, "An enhanced framework for aspect-based sentiment analysis of Hotels' reviews: Arabic reviews case study," in *Proc. 11th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2016, pp. 98–103.

[3] M. M. Abdelgwad, T. H. A. Soliman, A. I. Taloba, and M. F. Farghaly, "Arabic aspect based sentiment analysis using bidirectional GRU based models," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6652–6662, Oct. 2022.

[4] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews," *J. Comput. Sci.*, vol. 27, pp. 386–393, Jul. 2018.

[5] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, "Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 8, pp. 2163–2175, Aug. 2019.

[6] M. Al-Smadi, M. M. Hammad, S. A. Al-Zboon, S. Al-Tawalbeh, and E. Cambria, "Gated recurrent unit with multilingual universal sentence encoder for Arabic aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 261, Feb. 2023, Art. no. 107540.

[7] A.-S. Mohammad, M. Al-Ayyoub, H. N. Al-Sarhan, and Y. Jararweh, "An aspect-based sentiment analysis approach to evaluating Arabic news affect on readers," *J. Universal Comput. Sci.*, vol. 22, no. 5, pp. 630–649, 2016.

[8] S. Al-Dabet, S. Tedmori, and M. Al-Smadi, "Extracting opinion targets using attention-based neural model," *Social Netw. Comput. Sci.*, vol. 1, pp. 1–10, Sep. 2020.

[9] R. Bensoltane and T. Zaki, "Towards Arabic aspect-based sentiment analysis: A transfer learning-based approach," *Social Netw. Anal. Mining*, vol. 12, no. 1, pp. 1–16, Dec. 2022.

[10] A. S. Fadel, M. E. Saleh, and O. A. Abulnaja, "Arabic aspect extraction based on stacked contextualized embedding with deep learning," *IEEE Access*, vol. 10, pp. 30526–30535, 2022.

[11] M. M. Abdelgwad, T. Hassan A Soliman, and A. I. Taloba, "Arabic aspect sentiment polarity classification using BERT," 2021, *arXiv:2107.13290*.

[12] S. Al-Dabet, S. Tedmori, and M. Al-Smadi, "Enhancing Arabic aspect-based sentiment analysis using deep learning models," *Comput. Speech Lang.*, vol. 69, Sep. 2021, Art. no. 101224.

[13] M. E. Chennafi, H. Bedlaoui, A. Dahou, and M. A. A. Al-qaness, "Arabic aspect-based sentiment classification using Seq2Seq dialect normalization and transformers," *Knowledge*, vol. 2, no. 3, pp. 388–401, Aug. 2022.

[14] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 at SemEval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis," 2016, *arXiv:1609.02748*.

[15] X. Li, L. Bing, P. Li, and W. Lam, "A unified model for opinion target extraction and target sentiment prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6714–6721.

[16] B. Carpenter, "Coding chunkers as taggers: IO, BIO, BMEWO, and BMEWO+," *LingPipe Blog*, p. 14, Oct. 2009. [Online]. Available: https://web.archive.org/web/20170805150451/https://lingpipe-blog.com/2009/10/14/coding-chunkers-as-taggers-io-bio-bmewo-and-bmewo/

[17] H. Ding, S. Huang, W. Jin, Y. Shan, and H. Yu, "A novel cascade model for end-to-end aspect-based social comment sentiment analysis," *Electronics*, vol. 11, no. 12, p. 1810, Jun. 2022.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[19] M. F. Abdelfattah, M. W. Fakhr, and M. A. Rizka, "ArSentBERT: Fine-tuned bidirectional encoder representations from transformers model for Arabic sentiment classification," *Bull. Electr. Eng. Informat.*, vol. 12, no. 2, pp. 1196–1202, Apr. 2023.

[20] R. Bensoltane and T. Zaki, "Combining BERT with TCN-BiGRU for enhancing Arabic aspect category detection," *J. Intell. Fuzzy Syst.*, vol. 44, no. 3, pp. 4123–4136, Mar. 2023.

[21] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," 2020, *arXiv:2003.00104*.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[23] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:219683473

[24] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, Architectures and Applications*. Berlin, Germany: Springer, 1990, pp. 227–236.

[25] M. Al-Ayyoub, H. Al-Sarhan, M. Al-So'ud, M. Al-Smadi, and Y. Jararweh, "Framework for affective news analysis of Arabic news: 2014 Gaza attacks case study," *J. Univers. Comput. Sci.*, vol. 23, pp. 327–352, Jan. 2016.

[26] M. Al-Smadi, M. Al-Ayyoub, Y. Jararweh, and O. Qawasmeh, "Enhancing aspect-based sentiment analysis of Arabic Hotels' reviews using morphological, syntactic and semantic features," *Inf. Process. Manag.*, vol. 56, no. 2, pp. 308–319, Mar. 2019.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, *arXiv:1607.01759*.

[30] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 2054–2059.

[31] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," 2019, *arXiv:1907.04307*.

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

[33] M. Al-Smadi, O. Qawasmeh, B. Talafha, and M. Quwaider, "Human annotated Arabic dataset of book reviews for aspect based sentiment analysis," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, Aug. 2015, pp. 726–730.

[34] X. Wang, G. Xu, Z. Zhang, L. Jin, and X. Sun, "End-to-end aspect-based sentiment analysis with hierarchical multi-task learning," *Neurocomputing*, vol. 455, pp. 178–188, Sep. 2021.

[35] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting BERT for end-to-end aspect-based sentiment analysis," 2019, *arXiv:1910.00883*.

[36] B. Xu, X. Wang, B. Yang, and Z. Kang, "Target embedding and position attention with LSTM for aspect based sentiment analysis," in *Proc. 5th Int. Conf. Math. Artif. Intell.*, Apr. 2020, pp. 93–97.

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

[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[39] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss functions and metrics in deep learning," 2023, *arXiv:2307.02694*.

[40] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, *arXiv:1603.01360*.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[42] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc., Ser. B*, vol. 36, no. 2, pp. 111–133, Jan. 1974.

[43] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.

[44] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432.

[45] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, and O. De Clercq, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. Workshop Semantic Eval., Assoc. Comput. Linguistics*, 2016, pp. 19–30.

GHADA M. SHAFIQ received the B.Sc. degree from the Department of Computer Science, Faculty of Computer and Information Sciences, Mansoura University, Mansoura, Egypt, in 2018. Since 2018, she has been a Demonstrator with the Department of Computer Science, Faculty of Computer and Information Sciences, Mansoura University. Her research interests include natural language processing, artificial intelligence, and language models.

TAHER HAMZA, photograph and biography not available at the time of publication.

MOHAMMED F. ALRAHMAWY received the B.Eng. degree in electronics engineering and the M.Sc. degree in automatic control engineering from Mansoura University, Egypt, in 1997 and 2001, respectively, and the Ph.D. degree in computer science from the Real-Time Systems Research Group, Department of Computer Science, University of York, U.K., in 2011. In 2005, he joined the Real-Time Systems Research Group, Department of Computer Science, University of York, as a Ph.D. Research Student. In 2011, he joined the Department of Computer Science, Mansoura University, as a Lecturer. Since January 2022, he has been the Acting Head of the Department of Computer Science, Mansoura University. In January 2023, he was a Professor of computer science with the Department of Computer Science, Mansoura University. His current research interests include deep learning, network and graph analytics, real-time systems and languages, NLP, cloud computing, distributed and parallel computing, image processing, computer vision, the IoT, and big data. He was a recipient of the Best M.Sc. Thesis Award from Mansoura University, in 2003. His Ph.D. was fully funded by the Egyptian Ministry of Higher Education.

REEM EL-DEEB was born in El-Mahalla El-Kubra, Egypt, in 1987. She received the B.S., M.Sc., and Ph.D. degrees in computer science from the Faculty of Computers and Information, Mansoura University, Egypt, in 2008, 2012, and 2019, respectively. In 2009, she joined the Department of Computer Science, Mansoura University, as a Teaching Assistant. In 2012, she was an Assistant Lecturer. In 2019, she was an Assistant Professor. Her current research interests include natural language processing, artificial intelligence applications, machine learning for text semantic analysis, and language understanding. She was a recipient of the Scientific Publishing Grant Award in 2019.