Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.037112

Article

Multi-Task Learning Model with Data Augmentation for Arabic Aspect-Based Sentiment Analysis

Arwa Saif Fadel[1,2,*], Osama Ahmed Abulnaja[1] and Mostafa Elsayed Saleh[1]

[1]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

[2]Department of Computer Science, Faculty of Computer Sciences and Engineering, Hodeidah University, Yemen

*Corresponding Author: Arwa Saif Fadel. Email: afadl@stu.kau.edu.sa

Received: 23 October 2022; Accepted: 30 January 2023

Abstract: Aspect-based sentiment analysis (ABSA) is a fine-grained process. Its fundamental subtasks are aspect term extraction (ATE) and aspect polarity classification (APC), and these subtasks are dependent and closely related. However, most existing works on Arabic ABSA content separately address them, assume that aspect terms are preidentified, or use a pipeline model. Pipeline solutions design different models for each task, and the output from the ATE model is used as the input to the APC model, which may result in error propagation among different steps because APC is affected by ATE error. These methods are impractical for real-world scenarios where the ATE task is the base task for APC, and its result impacts the accuracy of APC. Thus, in this study, we focused on a multi-task learning model for Arabic ATE and APC in which the model is jointly trained on two subtasks simultaneously in a single model. This paper integrates the multi-task model, namely Local Cotext Foucse-Aspect Term Extraction and Polarity classification (LCF-ATEPC) and Arabic Bidirectional Encoder Representation from Transformers (AraBERT) as a shred layer for Arabic contextual text representation. The LCF-ATEPC model is based on a multi-head self-attention and local context focus mechanism (LCF) to capture the interactive information between an aspect and its context. Moreover, data augmentation techniques are proposed based on state-of-the-art augmentation techniques (word embedding substitution with constraints and contextual embedding (AraBERT)) to increase the diversity of the training dataset. This paper examined the effect of data augmentation on the multi-task model for Arabic ABSA. Extensive experiments were conducted on the original and combined datasets (merging the original and augmented datasets). Experimental results demonstrate that the proposed Multi-task model outperformed existing APC techniques. Superior results were obtained by AraBERT and LCF-ATEPC with fusion layer (AR-LCF-ATEPC-Fusion) and the proposed data augmentation word embedding-based method (FastText) on the combined dataset.

Keywords: Arabic aspect extraction; arabic sentiment classification; AraBERT; multi-task learning; data augmentation

1 Introduction

Aspect-based sentiment analysis (ABSA) plays an important role in sentiment analysis, where the sentiment polarities of a review or comment text are predicted based on the aspect terms [1,2]. The two fundamental subtasks of ABSA are aspect term extraction (ATE) and aspect polarity classification (APC). Aspect terms are the features of reviews about services or products, and APC expresses the sentiment polarities toward the extracted aspect terms (positive, negative, or neutral). For example, the review in Arabic about a hotel, “The location of the hotel is great, but the service is bad” (“موقع الفندق رائع لكن الخدمة سيئة”), provides positive and negative opinions about a hotel relative to two aspect terms, i.e., “location” and “service,” respectively. Note that ATE is treated as a sequence labeling problem, and APC is a classification problem.

Traditional machine learning and deep learning approaches, such as recurrent neural networks (RNN) and their variants, long short-term memory (LSTM), and the gated recurrent unit (GRU) [3], or a combination of traditional methods and deep learning models such as integrating of deep models and dependency rules [4], are frequently used to solve the sentiment analysis, ATE and APC problems. Machine learning requires handcrafted feature extraction methods. However, deep learning methods incur lower costs for feature extraction but require a large dataset to train the models.

Recently, the requirement for large training datasets has been solved using transfer learning based on pre-trained language models (PTM), e.g., Flair [5] and bidirectional encoder representations from transformers (BERT) [6]. PTMs are trained on huge volumes of data transferred to downstream tasks using weights that are learned from PTM as the initial weights in the other tasks.

Most traditional methods separately perform ATE and APC, and some studies have assumed that the aspect terms are given or preidentified [7–10], which is impractical as aspect terms are not preidentified in real-world scenarios. Pipeline solutions design different models for each task, and the output from the ATE model is used as the input to the APC model. ATE and APC tasks are sequentially performed using different models [11,12]. The models are stacked one after another, i.e., one model is used for ATE and another model is employed to classify the extracted aspect terms. The limitation of pipeline methods is that they require two steps to achieve ATE and APC, i.e., the models for the subtasks are independently trained, and the relationship between ATE and APC is not explicitly modeled; thus, error propagation is possible among the steps. Here, an error in the ATE step will propagate to the APC step, negatively impacting the ABSA process’s overall performance. In ATE and APC, multi-task learning has overcome these limitations by jointly training a single model, where the aspect term and its polarity are simultaneously extracted. Here, model performance is improved using the similarities between two tasks in multi-task learning. The advantages of multi-task learning are its ability to share knowledge between many tasks and train multiple tasks in parallel using a shared representation, where learning each task can enhance the learning and generalizability of other tasks. Multi-task learning is achieved by optimizing multiple loss functions for multiple tasks in a single model rather than a single task (i.e., the objective functions are altered between tasks).

Data augmentation techniques generate extra data to train machine learning and deep learning models, which is beneficial for models that require a large, labeled dataset to realize sufficient training. Data augmentation improves the generalizability of models and helps avoid overfitting problems. Data augmentation techniques have been successfully applied in computer vision and image classification tasks [13]. However, applying data augmentation techniques in natural language processing (NLP) tasks is a relatively new concept. Several alternative approaches have been proposed, ranging from methods that only slightly modify the data to data generation and paraphrasing methods. For example, data modification can be realized using easy data augmentation (EDA) methods [14]. Word replacement (using dictionaries) is employed to replace an original word in a sentence with synonymous terms, delete some words from the given sentence, or swap random words [15]. Note that removing or swapping words may change the meaning and provide low-quality samples, which will negatively affect performance. Back translation methods employ data paraphrasing. Recent studies have used semantic embedding augmentation based on word embedding techniques, e.g., AraVec [15] and FastText [16]. Semantic embedding methods overcame the limitations of the thesaurus-based method (i.e., WordNET) by suggesting replacement words that are closest to the original words in the vector space. Moreover, generation methods generate new data from the original data using language models, e.g., generation pre-trained transformer (GPT)s [17]. For contextual augmentation, pre-trained language models are utilized for data augmentation, where a random mask where random masked words are predicted based on the context using masked language models [6].

The Arabic language's complexity distinguishes it from other languages, such as English. Arabic has a richer inflectional and derivative morphology. There is a limited number of resources and tools for Arabic, so there is a lack of language resources and tools [18,19]. Thus, only a limited number of studies have attempted to develop Arabic ABSA (AABSA) methods. This paper inspired by a previous study [20] that proposed the multi-task learning model Local Context Focus for Aspect Term Extraction and Polarity classification (LCF-ATEPC) for English and Chinese and proved its effectiveness. This paper adopted a multi-task learning model (LCF-ATEPC) and integrated it with AraBERT [21] to extract aspect terms and classify their sentiment polarities in an Arabic dataset. The proposed model employs AraBERTV02 as a contextual representation layer, and the local context focus mechanism (LCF) technique is used to consider the local context for APC [22].

Furthermore, the proposed model benefits from state-of-the-art data augmentation methods based on PTM (i.e. contextual and noncontextual), offering better word coverage than the traditional thesaurus-based method (WordNET). Here, we employ two data augmentation techniques to increase diversity in the training dataset. These data augmentation techniques are based on pre-trained word embedding substitution techniques (i.e., FastText and AraVec) and a pre-trained language model (i.e., AraBERT). This paper examined the impact of the proposed data augmentation techniques on the performance of the multi-task model in the ATE and APC tasks. Unlike previous methods that did not augment aspect terms, aspect terms are augmented in the proposed method by applying constraints on how the suggested replacement words are selected.

The primary contributions of this paper are summarized as follows.

- This paper proposes a multi-task learning model that integrates LCF-ATEPC and a pre-trained Arabic BERT language model (AraBERT) to realize simultaneous Arabic ATE and APC. To the best of our knowledge, this is the first study to develop a multi-task learning model for the Arabic ATE and APC tasks.
- This paper introduces two data augmentation strategies using state-of-the-art methods, i.e., word embedding and contextual models to increase training data diversity. Moreover, this paper investigate how the proposed data augmentation strategies can improve the effectiveness of the multi-task model. To the best of our knowledge, this is the first study applying data augmentation to the AABSA task, especially on the Arabic Hotels' reviews dataset [23].

- We evaluate the impact of the proposed data augmentation strategies on the performance of the proposed multi-task model. Experimental results demonstrate that the proposed multi-task model achieved better results using a combination of the original and augmented datasets compared to using only the original dataset (Hotels' reviews).

The remainder of this paper is organized as follows. Section 2 summarizes related work, and Section 3 describes the proposed approaches in detail. The experimental process and settings are described in Section 4, and the experimental results are presented and analyzed in Section 5. Finally, conclusions are given in Section 6.

2 Related Work

2.1 Aspect-based Sentiment Analysis

ABSA comprises several primary subtasks, i.e., ATE, APC, and aspect category detection. In this section, we focus on the ATE and APC subtasks. In addition, we summarize previous studies on ABSA tasks with a focus on ABSA for the Arabic language.

2.1.1 Aspect Term Extraction

There are three types of ATE methods that are based on existing techniques, i.e., rule-based or dictionary-based methods, traditional machine learning methods, and more recent approaches based on deep learning and transformers. Automatic or manual extraction of aspect phrases can be realized using rule-based techniques, which depend on specified rules to achieve ATE [24,25]. Rule-based techniques are highly dependent on external resources that are not supported and unavailable in low-resource languages. Here, more frequent elements, e.g., nouns or noun phrases, are extracted using frequency-based algorithms with less grammatical information [26]. Moreover, previous studies have employed the topic model concept for aspect extraction [27,28]. Traditional machine learning techniques handle the ATE task as a sequence labeling problem, which has been realized using various machine learning-based methods, e.g., support vector machines (SVM) [29] and conditional random fields (CRF) [30,31]. Furthermore, aspect extraction from e-commerce comments was accomplished by combining traditional methods with machine learning [32].

Recently, studies have attempted to adopt deep learning to the ATE task to reduce the human labor and feature extraction costs incurred by machine learning models. For example, a paper [33] suggested a model with two layers, bidirectional LSTM (BiLSTM)and CRF layers, to extract explicit and unsupervised aspect terms. Moreover, an RNN and CRF have been integrated to enhance the ATE task to identify the aspect boundary [1,9]. Previous studies have also enhanced existing conventual neural networks (CNNs) to improve ATE tasks [11,34]. A previous study [35] employed multi-layers of CNN with word embedding to tag each word in the given sentence to determine if it is an aspect or nonaspect. Moreover, attention mechanisms have been used to help models learn representations more effectively by emphasizing words that correlate to the aspects [7,36–38]. Recently, with the emergence of transfer learning and the development of contextual embedding and pre-trained language models to improve NLP tasks, many studies have employed contextual embedding to achieve superior results in ABSA. For example, in a previous study [39], many datasets from different domains were combined, and the labels were unified. Here, a fine-tuned BERT model was employed to extract aspects from the multi-domain datasets. Moreover, Lopes et al. [40] utilized BERT for the ATE in Portuguese. Winatmoko et al. [41] extracted aspect terms from Bahasa Indonesian hotel reviews using the multilingual BERT, extra auxiliary label, and CRF as the output layer, and the results demonstrated improved performance.

2.1.2 Aspect Polarity Classification

Various techniques have been proposed for the second subtask of ABSA (APC): Traditional machine learning-based and deep learning-based models. Note that conventional machine learning-based approaches require extensive feature engineering; thus, most recent studies considered deep learning techniques, and various deep learning-based approaches have been proposed to improve APC. For example, Tang et al. [42] proposed two LSTM variants, i.e., Target-Dependent LSTM (TD-LSTM) and Target-Connection LSTM (TC-LSTM). In the ABSA task, the attention mechanism can capture the significance of each context word relative to a target by modeling their semantic correlation [43]. Moreover, certain methods have integrated various other techniques with deep learning methods for the APC task. For example, Meng et al. [44] proposed the CNN-BiLSTM approach. Here, a CNN was employed to obtain a higher-level representation of the sequence, and BiLSTM was used for local and global feature extraction after highlighting the aspect terms using attention, where the attention for each word was calculated to identify words with high weights. A previous study [43] developed a model of LSTM and attention with aspect embedding. Based on a dependency tree and affective common sense, Liang al. [45] constructed a graph convolution network to capture sentiment dependencies with distinct aspects.

The limitations of using a word embedding with deep learning because of its context-free nature (where many words have the same embedding regardless of context) have led to using pre-trained contextual models, and such models have demonstrated superior performance in most recent studies. For example, Li et al. [11] utilized an auxiliary sentence to convert the ABSA issue from a single-sentence classification challenge to a sentence-pair classification task. In [46], they employed a BERT model for text representation with CRF for ABSA in an end-to-end manner. Target-dependent with BERT (TD-BERT) modifies the original version of BERT for aspect classification. They proposed using position rather than using the first token BERT as input for APC [10]. A previous study [47] used a post-training BERT model with a masked language model (MLM) model to extract domain-specific features.

2.1.3 Multi-Task Learning for ABSA

As previously stated, most previous studies individually or sequentially handled the ATE and APC tasks, where independent models were designed for each task. However, ATE and APC are performed jointly in multi-task learning. For example, a previous study [48] proposed a joint model for ATE and APC in an end-to-end network. They conducted experiments using several deep neural networks and word embedding on a German dataset. In addition, Nguyen et al. [49] proposed a unified model that simultaneously handled the ATE and APC tasks based on the BiLSTM and CRF techniques. They evaluated their model on four benchmark datasets from SemEval-2014 [50], SemEval-2015 [51], and SemEval-2016 [23], and the results proved the effectiveness of the model. Wang et al. [52] proposed a multi-task model for aspect extraction and polarity classification based on an attention mechanism. In [53], the researchers introduced a multi-task model using BiLSTM with a self-attention mechanism for the ATE task and a CNN for the APC task. In addition, in [54], the authors proposed a multi-task model based on deep learning for ATE and APC tasks on the Vietnamese dataset for restaurant and hotel domains. Another study [20] employed a fine-tuned BERT model as a multi-task model for ABSA with a self-attention layer on top of the BERT model.

Note that most previously proposed ABSA methods focus on English because of the availability of English resources. Moreover, there are strong NLP tools for English. Thus, the best existing methods for ABSA are adapted or modified to Arabic and other languages. Arabic is a challenging and low-resource language [18]; thus, few methods have been proposed for this language. Table 1 lists and compares previous studies that have considered the Arabic language. These studies are compared in terms of the AABSA tasks, proposed models, dataset domains (if a multi-task task model was used), and data augmentation techniques.

2.2 Data Augmentation for ASBA

Data augmentation is widely used in image classification [13,55]. Several data augmentation approaches have been proposed to increase the size of training datasets for various NLP tasks, such as text classification [14,56] named entity recognition [57], and machine translation [58]. However, few studies have investigated data augmentation techniques for ABSA in English or other languages, and we could not identify any previous study that investigated data augmentation for an Arabic ATE and APC.

A previous study [59] used naïve EDA techniques (e.g., random swap, random deletion, random insertion, and synonym replacement) for data augmentation with some adaptations to be compatible with ABSA. Then, to select accurate words for augmentation, they extended EDA with some adjustments based on word sense disambiguation. Moreover, Li et al. [60] proposed two techniques for synonym replacement. The first technique was based on using part of speech information to determine the tokens to be replaced. Then, using WordNET, they selected the most similar words to the original words and replaced them with synonyms with high cosine similarity. The second method is based on syntactic dependency. Here, in each sentence, the tokens were swapped with the tokens with tokens that have the same part of speech (PoS) and the same label with dependency arc with root. Li et al. [61] introduced the conditional augmentation strategy of ATE; mask sequence to sequence was employed to generate new sentences. Some words from the input sentence are masked and replaced with generated words. They used an encoder to encode the masked word and their label as input (to preserve the original label) and a decoder to reform this input.

Table 1: Existing Arabic ABSA methods

Reference	Evaluated task	Dataset domain	Proposed model	Multi-task model	Data augmentation
[62]	Aspect sentiment classification	Large-scale Arabic Book Review (LABR)	Naïve Bayes Bernoulli naïve Bayes	✗	✗
[63]	-Aspect category detection -Aspect category detection	Selected reviews from LABR dataset (1513 reviews)	Lexicon-based approaches	✗	✗
[64]	Aspect term extraction	Restaurants, movies, and hotels datasets	N-grams and a PoS tagger were used to extract aspects	✗	✗
[65]	-Aspect term extraction -Polarity term extraction	Arabic news affected readers dataset	-PoS, NER, N-gram for feature extraction -CRF, decision tree, and Naïve Bayes and K-nearest neighbors for classifiers	✗	✗
[66]	-Aspect category detection -Aspect polarity extraction -Sentiment polarity classification	Arabic Hotels' reviews	Naïve Bayes, Decision tree, and K-nearest neighbors	✗	✗

(Continued)

Table 1: Continued

Reference	Evaluated task	Dataset domain	Proposed model	Multi-task model	Data augmentation
[67]	-Aspect detection	Arabic airline tweeter	-Word embedding for feature representation -SVM for classification	✕	✕
[68]	-Aspect-based sentiment detection -Aspect category detection -Aspect extraction -Sentiment polarity classification	Arabic Hotels' reviews	-RNN -SVM	✕	✕
[69]	-Aspect term extraction -Aspect polarity classification	Arabic Hotels' reviews	-Bi-LSTM and CRF For ATE -LSTM for APC	✕	✕
[70]	-Aspect extraction -Sentiment polarity classification	Arabic Hotels' reviews	-Combination of BiGRU and CNN and CRF (Bi-GRU-CNN-CRF) for aspect extraction -interactive attention with GRU for aspect extraction	✕	✕
[71]	-Aspect term extraction	Arabic Hotels' reviews	-Utilizes BiLSTM as the encoder-an LSTM as the decoder alongside an attention mechanism and CRF	✕	✕
[72]	-Sentiment polarity classification	Arabic Hotels' reviews Arabic News	Fine-tuned BERT model with linear layer for classification	✕	✕
[73]	-Aspect term extraction	Arabic News data set	Combination of BERT, BiLSTM, and CRF	✕	✕
[74]	-Aspect category detection -Aspect extraction	Human Annotated Arabic Dataset of Book Review (HAAD)	Using rule-based and ontology methods	✕	✕
[75]	-Aspect extraction	Arabic Hotels' reviews	Combination of contextual embedding (AraBERT and Flair) with deep learning and CRF BERT-Flair-BiLSTM/BiGRU-CRF	✕	✕
[76]	-Aspect term polarity classification -Aspect category polarity classification	-Human Annotated Arabic Dataset of Book Review (HAAD) -Arabic Hotels' reviews	Using sequence to sequence model for normalizing text befor classification	✕	✕
[77]	-Aspect polarity classification	-Human Annotated Arabic Dataset of Book Review (HAAD) -Arabic Hotels' reviews -Arabic News	Using BERT with liner layer for aspect sentiment classification	✕	✕
[78]	-Aspect category detection	-Arabic News	Combination of BERT and Temporal conventional network and BiGRU	✕	✓

(Continued)

Table 1: Continued

Reference	Evaluated task	Dataset domain	Proposed model	Multi-task model	Data augmentation
Our Proposed study	-Aspect extraction	-Arabic Hotels' reviews	multi-task model LCF-APTEPC with AraBERTvo2	✓	✓
	-Aspect polarity classification	-A combination of Arabic Hotels' reviews and augmented datasets			

Unlike the methods proposed in previous studies, we propose a multi-task learning model that simultaneously handles Arabic ATE and APC tasks. The proposed model evaluated the Arabic Hotels' reviews dataset. Additionally, the proposed model implements data augmentation techniques based on word embedding substitution and a pre-trained AraBERT model. The augmented dataset was combined with the original dataset. Finally, we evaluated the impact of data augmentation techniques on the proposed multi-task model.

3 Proposed Methods

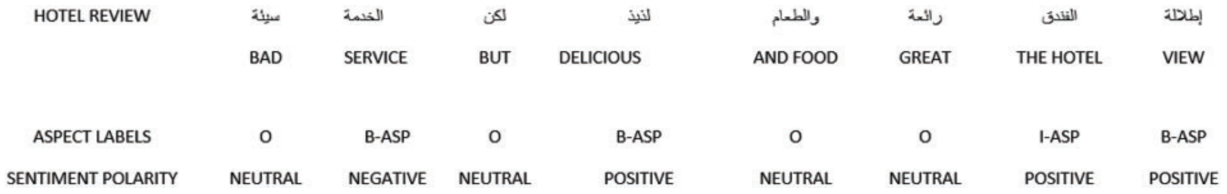
3.1 Task Definition

This study aimed to extract ATE from Arabic hotel review text and predict the sentiments of the extracted aspect terms. For instance, consider the following review text: "the location is very good, but the service is bad." Here, the model should extract the "location" and "service" aspects and their respective sentiment polarities, i.e., positive, negative, or neutral.

Here, to implement multi-task learning, the aspect term and its sentiment polarity are simultaneously retrieved. Thus, the input sequence is tokenized, giving each token two labels. The first label indicates whether the token is part of the aspect term, whereas the second label identifies the aspect term's sentiment polarity.

3.2 Data Preprocessing

The dataset used in this study contains user reviews of hotels. However, the text data include various elements that impact the performance of ATE and APC tasks, e.g., hyperlinks or uniform resource locator (URLs), numbers, English letters, and inappropriate symbols and punctuation. Thus, to reduce noise and the size of the vectors, we removed all unnecessary punctuation marks and special characters (@, #, %, &, ∗,?, ^, !), sequences of English letters, and numbers. The dataset was then reformatted from extensible markup language (XML) format to a text file with two labels, i.e., one label for aspect terms and another label for sentiment polarity. Note that the beginning of aspect (B-ASP) indicates the first word of the aspect term, inside of aspect (I-Asp) indicates a word inside the aspect term, and outside (O) is a nonaspect word. In addition, the sentiment polarity of each aspect term can be positive, negative, or neutral. For example, the following review was annotated as shown in Fig. 1: "إطلالة الفندق رائعة والطعام لذيذ لكن الخدمة سيئة" "the view of the hotel is great, the food is delicious, but the service is bad."



HOTEL REVIEW	سيئة	الخدمة	لكن	لذيذ	والطعام	رائعة	الفندق	إطلالة
	BAD	SERVICE	BUT	DELICIOUS	AND FOOD	GREAT	THE HOTEL	VIEW
ASPECT LABELS	O	B-ASP	O	B-ASP	O	O	I-ASP	B-ASP
SENTIMENT POLARITY	NEUTRAL	NEGATIVE	NEUTRAL	POSITIVE	NEUTRAL	NEUTRAL	POSITIVE	POSITIVE

Figure 1: Example of two label annotation

3.3 Data Augmentation

In the following, we describe the data augmentation techniques employed in the proposed model.

3.3.1 Word Embedding-Based Data Augmentation

By modifying the original training dataset, data augmentation techniques are used to enrich and increase the training dataset's size and data diversity and improve model performance.

In the proposed model, we employ two substitution techniques based on PTM for data augmentation, i.e., word embedding and contextual augmentation. In word embedding, words can be represented with low-dimensional dense vectors (i.e., most elements are nonzero). Here, the distance and direction of the vectors represent the semantic relationships among words. The closer the words are in meaning, the closer the distance between them. For example, synonyms are observed to be near each other, whereas antonyms are noticeably distant from each other.

This paper uses two word embedding techniques, i.e., FastText and AraVec, to augment the dataset. A general overview of the proposed data augmentation architecture is shown in Fig. 2. The data augmentation algorithm using word embedding is described as follows. First, vectors for each word in the training dataset are obtained using FastText or AraVec (excluding stop words and punctuation).

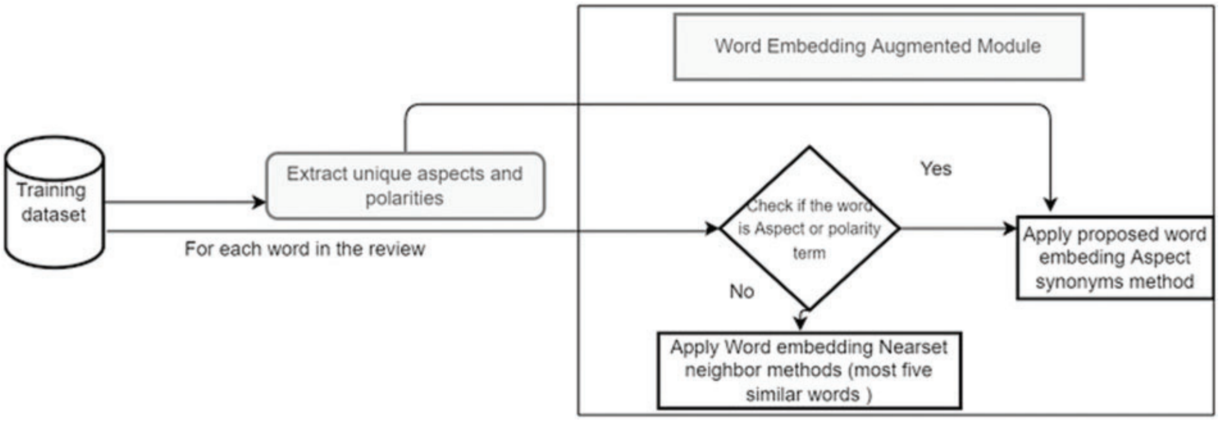
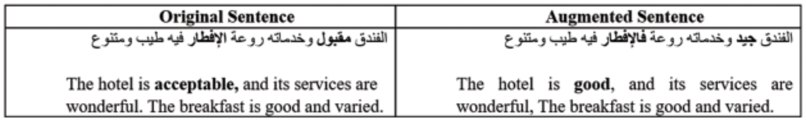

Word Embedding Augmented Module
Training dataset
Extract unique aspects and polarities
For each word in the review
Check if the word is Aspect or polarity term
Yes
No
Apply proposed word embeding Aspect synonyms method
Apply Word embedding Nearset neighbor methods (most five similar words)

Figure 2: The architecture of word embedding substitution approach

Here, aspect synonyms are generated as follows: 1) a list of unique aspects is created from the original dataset; 2) a vector is generated for each aspect (in a unique aspect list) using FastText or Word2Vec; 3) words with the highest similarity to each aspect in terms of the cosine similarity of the word embeddings are obtained to create a list of synonyms, where words with higher cosine similarities are selected as synonyms for replacement.

Then, we select accurate, high-quality, and relevant synonyms for the original aspect by placing additional constraints on the words' synonyms. First, we select two random words from the aspect synonym list for each aspect in a unique list (extracted by the word embedding) to obtain the top 10 neighbors for each word (if available). Then, we obtain the words present in both lists and add them to the list of synonyms. This technique creates a new synonym list for aspect terms where the original word is replaced by one of these synonyms. If there is no intersection, we add the top three neighbors of the original word. Here, we apply the same method to sentiment polarity terms to enrich the augmented data and improve robustness. We create a synonyms list for other words (nonaspect and nonpolarity) by selecting the top five similar words (nearest neighbors) from the vector space. For augmentation, we randomly replace 30% of the words in the review (with one of its synonyms).

This process is repeated five times for each sentence. Consequently, we obtain five times as much data, i.e., we generate 24000 extra reviews) five times the original training dataset). Finally, we combine the original and augmented data. Here, the most important factor to consider is the number of words to be replaced in each review. Thus, we replace 30% of each review. Moreover, for selecting accurate synonyms for the original words, we realized that by applying constraints on how to select the replacement words. Fig. 3 shows an example sentence before and after data augmentation via FastText, where bold words identify the replacement words proposed by the data augmentation.



Original Sentence	Augmented Sentence
الفندق **مقبول** وخدماته روعة **الإفطار** فيه طيب ومتنوع The hotel is **acceptable,** and its services are wonderful. The breakfast is good and varied.	الفندق **جيد** وخدماته روعة **فالإفطار** فيه طيب ومتنوع The hotel is **good**, and its services are wonderful, The breakfast is good and varied.

Figure 3: Example of data augmentation, replacement words in bold

3.3.2 Contextual Data Augmentation

The second data augmentation technique is based on contextual augmentation. Here, used AraBERTV02 to mask four random words in the sentence, and then new words are predicted as new replacement words. First, the sentence is tokenized, a single random word from the sentence is then masked (while ignoring sub-tokens after tokenization), and the sentence is regrouped and input to the contextual model to predict the masked word. The suggested new words are based on the context, i.e., the suggested new words are based on the other words in the sentence and the position of the masked token word. Finally, the predicted word from BERT model with the highest score is used to replace the original word.

To preserve the semantic meaning of the sentence, we avoid replacing greater than 50% of the words in short sentences, especially those with fewer than five words. This process is repeated with the other three random words and then generates the augmented sentence, and each sentence is augmented five times. Generally, with all augmentation techniques, we found that some suggested word is the same word with an added prefix or suffix that does not make a great notable change in the augmented text.

3.4 Multi-Task Model Architecture

The proposed model was inspired by the success of the multi-task LCF-ATEPC [20] and pre-trained Arabic language AraBERT models. The proposed model employs an LCF-ATEPC model for Arabic ATE and APC tasks. This paper integrated this model with AraBERTv02 for Arabic word representation. This model aims to extract aspect terms from the review text and simultaneously classify their sentiment polarity. The main techniques used in the proposed model are AraBERT and the LCF mechanism for AABSA. The model comprises an embedding layer and two independent pre-trained AraBERT layers for global and local contexts to extract the local and global context features, respectively. In addition, the model contains three more layers: multi-head self-attention (MHSA), interactive learning, and output. The proposed model contains the following main components (see Fig. 4).

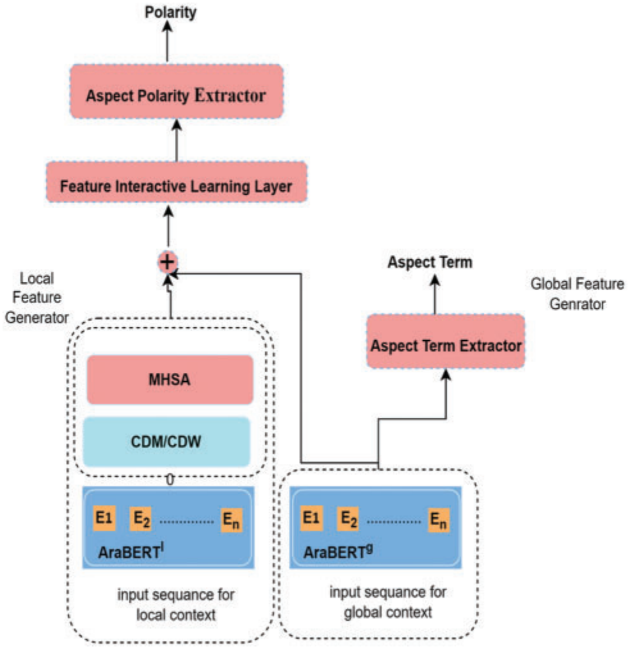


Figure 4: Architecture of the proposed multi-task model integrating LCF-ATEPC [20] and AraBERT [21]

3.4.1 Embedding Layer

Recall that we reformat the Hotels' reviews dataset from XML format to a dual-label dataset. Here, each review is tokenized, and each token is assigned a label to indicate whether the token is an aspect term or not, and the other label indicates the sentiment polarity of the given aspect term. To realize effective feature representation, we employ the AraBERT model [21], which is a pre-trained BERT transformer model, as a shared BERT layer. This model encodes a word using a transformer encoder in a bidirectional manner; it indicates the semantics of the word in the context depending on its semantic relationship with relevant words in the text [21]. Here, a contextual embedding vector is output for each word [79].

In the proposed model, AraBERTv02 is employed to generate two parallel embedding layers. The first layer is used to extract the local context features $AraBERT^{l}$, and the second layer is used to extract the global context features $AraBERT^{g}$. Note that fine-tuning of each layer is separately performed according to the joint loss function of multi-task learning. The embedding layers are represented by the following equations:

$$O^{l}_{BERT} = AraBERT^{l}\ \left(S^{l}\right) \quad (1)$$

$$O^{g}_{BERT} = AraBERT^{g}\ (S^{g}) \tag{2}$$

Here, O^{l}_{BERT} is the local input representation, and O^{g}_{BERT} is the global context representation. In addition, S^{l} and S^{g} are the sequence inputs for the local context and global context, respectively.

3.4.2 Multi-Head Self-Attention Mechanism

The attention head h is calculated for each semantic word using several weight matrices and different representations using MHSA, which performs multiple-scale products in parallel and concatenates them. The MHSA technique can avoid feature loss due to the long-distance dependencies between relative words in the sequence during feature learning. It is based on the self-attention mechanism, which is a special type of attention that helps detect relationships between different words in the input that indicate the syntactic and contextual structure of a sentence.

3.4.3 Local Context Focus Mechanism

The LCF mechanism has been used to extract local context features [22]. It is adapted to identify more semantic relative contextual words to aspect terms. Context words that are close to the aspect are more relevant than those that are distant; thus, we adopt the semantic relative distance (SRD) [22]. SRD is key to local context; it is based on threshold α, which determines how many words around the target can be considered local context. For example, if the SRD value is six, each contextual word with an SRD value less than or equal to six words will be considered local context; otherwise, it will be considered global context. The SRD value is calculated as follows.

$$SRD_i = \left|i - p^{t}\right| - \frac{\lfloor L_n \rfloor}{2} \tag{3}$$

where i is the word's position in the context, and p^{t} is the central position of the aspect term. In addition, L_n is the aspect length, and SRD represents the distance between the targeted aspect and the i th contextual word.

Local context features can be trained using a context dynamic mask (CDM), context dynamic weight (CDW), or a fusion of CDM and CDW [22,20]. CDM masks nonlocal features that are unrelated to the targeted aspect learned by the local AraBERT layer. Here, the features of less semantic words (nonlocal context words) will be masked and assigned zero vectors for feature vectors. Nonlocal contexts are tokens whose SRD for the targeted aspect is greater than a threshold value. Subsequently, the local context outputs are calculated as follows.

$$V_i = \begin{Bmatrix} E, & rd_i \le \alpha \\ 0, & rd_i > \alpha \end{Bmatrix} \tag{4}$$

$$M = [V_1, V_2 \ldots V_n] \tag{5}$$

$$O^{l}_{CDM} = O^{l}_{BERT} . M \tag{6}$$

$$O^{l} = MHSA\left(O^{l}_{CDM}\right) \tag{7}$$

where α represents the SRD threshold, M is the feature mask matric, V_i is the mask vector of each word, O is the zero vector, and E is ones vector. In addition, O^{l}_{CDM} represents the local features learned by the local context.

The CDW is another mechanism of the local context that preserves semantically relative features and assigns small weights to less semantic relative features (in terms of the target aspect) based on their SRD. The following equations are used to compute the CDW.

$$V_i = \left\{ \begin{matrix} E, & rd_i \le \alpha \\ \dfrac{SRD_i - \alpha}{n}.E & rd_i > \alpha \end{matrix} \right\} \tag{8}$$

$$W = \left[V_1^w, V_2^w \ldots V_2^w \right] \tag{9}$$

$$O^l_{CDw} = O^l_{BERT}.W \tag{10}$$

$$O^l = MHSA\left(O^l_{CDw}\right) \tag{11}$$

where, SRD_i is the SRD between the ith word in the input sequence and the target aspect, and n is the number of tokens in the input sequence. The CDW output is O^l_{Cw}.

Here, either the CDM or CDW approach can be used to learn the local context features. In addition, we can employ an approach that fuses the CDM and CWD techniques. After concatenation, the output from fusion layer passes into the linear layer to create the final fusion layer O^l_{fus}. Then, the local context is learned by the MHSA module to improve the relationship between the aspect and the context.

$$O^l_{fus} = \left[O^l_{CDM};\ O^l_{Cw} \right] \tag{12}$$

$$O^l_{fus} = \left[W^f O^f + b^f \right] \tag{13}$$

$$O^l = MHSA\left(O^l_{fus}\right) \tag{14}$$

Here, b^f is the bias vector, and W^f is the weight matrix.

3.4.4 Feature Interactive Learning Layer

In the APC process, feature interactive learning is first applied by concatenating the local and global context features (to enrich the feature representation. Then, the learned features are passed to the linear layer and another MHSA encoding. The following equations express these steps.

$$O^{lg} = \left[O^l;\ O^g \right] \tag{15}$$

$$O^{lg}_{dens} = \left[W^{lg} O^{lg} + b^{lg} \right] \tag{16}$$

$$O^{lg}_{FL} = MHSA\left(O^l_{fus}\right) \tag{17}$$

where, O^g represents the global context features learned by model, and O^l represents the local context features. In addition, b^{lg} is the bias vector, and W^{lg} is the weight matrix.

3.4.5 Sentiment Polarity Extractor

The sentiment Polarity Extractor represents the last output layer that predicts the sentiment polarity. Head pooling is performed on the output features learning by feature interactive learnings to extract the last hidden state from the position of the first token. Then, the output is given into softmax to classify the sentiment polarity for the aspect term. The following equations express these steps:

$$O_{pool} = \left(O_{FL}^{lg}\right) \tag{18}$$

$$Y_{polarity} = \frac{\exp\left(O_{pool}\right)}{\sum_{k=1}^{c} \exp\left(O_{pool}\right)} \tag{19}$$

where c is the classes {positive, negative, neutral} and $y_{polarity}$ is the sentiment polarity of the aspect term.

3.4.6 Aspect Term Extraction

The ATE process is a token-level problem where classification is performed at the token level. Here, the following SoftMax activation function is used to predict the tags of the input sequence.

$$A_{term} = \frac{\exp\left(term_i\right)}{\sum_{k=1}^{n} \exp\left(term_i\right)} \tag{20}$$

Note that the cross-entropy loss function is used for both ATE and APC. To jointly train the model on the two subtasks simultaneously, the joint loss function is the sum of the loss functions of each task.

$$L_{ATEPC} = L_{ATE} + L_{APC} \tag{21}$$

L_{ATEPC} means the joint loss function, L_{ATE} means the loss function of task ATE; L_{APC} means the loss function of task APC.

4 Experiments

4.1 Dataset and Experimental Settings

This paper used the benchmark dataset from the SemEval-2016 challenges to evaluate the proposed multi-task model, and the hotel dataset is from the Arabic Hotels domain on subtasks 2 and 3 from the SemEval 2016 task 5. The dataset was annotated at the text level with 2029 reviews divided into 1839 training instances and 425 testing instances, and at the sentence level with 6029 sentences (4082 sentences for training and 1227 sentences for testing), we will use 90% of training data for training set 10% as a validation set. Additional information about the dataset can be found in the literature [23]. Note that we focused on the sentence-level data in our experiments.

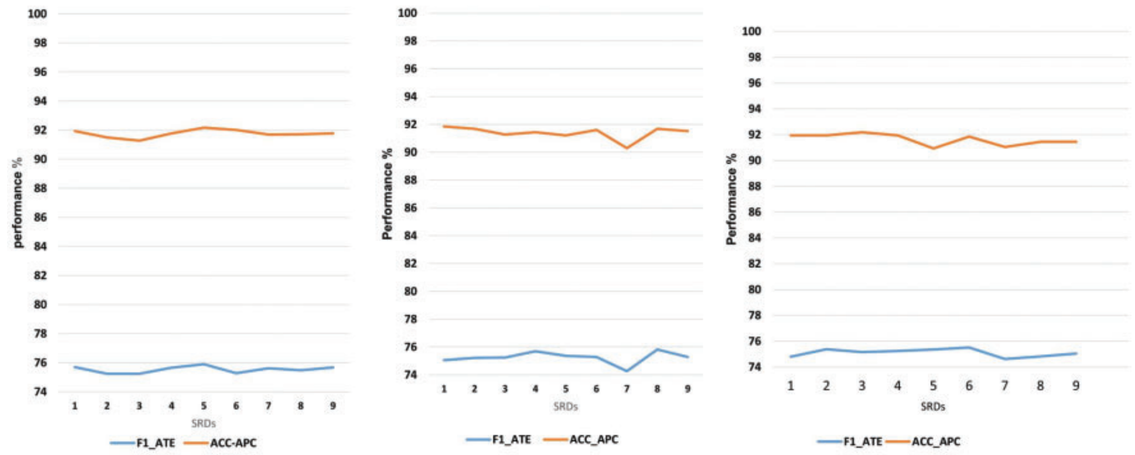
This paper evaluated the proposed model on new training datasets generated by combining the original and augmented datasets using three augmentation methods. Comprising 24005 training and 1227 testing data points, we use the original testing data.

As mentioned in the previous section, we reformatted the original dataset into two labels for each token in the sentence. Here, we employed a beginning, inside, and outside (BIO) annotation strategy for ATE if the word was an aspect term (word or phrase). Note that B-ASP indicates the first word of the aspect term, I-Asp indicates a word inside the aspect term, and O is a nonaspect word. Moreover, the sentiment polarity of each aspect term can be positive, negative, or neutral.

All experiments are implemented on Google Colaboratory (https://colab.research.google.com/) using NVIDIA T4 Tensor Core Graphics Processing Unit. Google Colab enables easier and faster implementation of machine learning algorithms. Here, a Python library was utilized for all implementations. Bert-base-arabertv02 version with 768 embedding dimensions was used as the contextual embedding layer in all experiments. After repeated experiments to discover the optimal settings of different hyperparameters (using different hyperparameters), we observed that the best results are obtained with the hyperparameters shown in Table 2 (hyperparameters for the model in all experiments). We tested the model with different learning rates and observed that the best results were obtained when the learning rate was 3×10^{-5}. Therefore, the learning rate was fixed at 3×10^{-5} in all experiments. The batch size was set to 16, and the maximum number of epochs was 5. After several experiments, the results in Fig. 5 demonstrate that varying SRD threshold values slightly affect the results obtained on the experimental dataset. For all experiments, we set the SRD threshold to 5 (the default value of LCF-ATEPC).

Table 2: Experimental hyperparameter settings

Parameter	Values
Learning rate	3×10^{-5}
Mini-batch size	16
Max. number of epochs	5
SRD threshold	5
Hidden size	768
Max. sequence length	140



(a) (b) (c)

Figure 5: Impact of SRD threshold on: (a) AR-LCF-ATE-Fusion (b) AR-LCF-ATEPC–CDW and (c) AR-LCF-ATEPC–CDW (F1_ATE: F1-score for ATE; ACC_APC: accuracy value for APC)

4.2 Evaluation

A set of measures can be used to evaluate the performance of text classification models. Precision, Recall, and F1 Score are the most popular performance measurements. These metrics are calculated based on the prediction results provided by the confusion matrix. This matrix quantifies the number of (in)correct predictions made by the model for each class. There are four types of prediction results in the confusion matrix when considering an individual class label k:

- True Positive (TP): denotes the number of observations that belong to class k and were correctly classified as k.
- True Negative (TN): denotes the number of observations that do not belong to class k and were not classified as k.
- False Positive (FP): denotes the number of observations that do not belong to class k but were incorrectly classified as k.
- False Negative (FN): denotes the number of observations that belong to class k but were incorrectly classified as another class.

The ratio of successfully predicted to total aspect terms or aspect sentiment polarities is called precision, whereas the ratio of correctly predicted to the number of aspect terms in the standard dataset is called recall.

F1 score is a metric that considers the precision and recall taking their harmonic mean. The best value of this score is 1 (best precision and recall), and the worst value is 0 [80]. The precision, recall and F1 score are computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

$$F1 = \frac{2 \cdot Precision \cdot Recal}{Precision + Recal} \tag{24}$$

In the case of APC, the precision, recall and F1-score are computed as follows (they are computed by macro-average per-class metrics):

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{25}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{26}$$

$$F1_k = \frac{2 \cdot Precision_k \cdot Recall_k}{Precision_k + Recall_k} \tag{27}$$

where $k \in$ [positive, negative, neutral]

The macro average calculates the global scores for these metrics as follows:

$$x_m = \frac{1}{3}\sum_{i=1}^{n} x_i \tag{28}$$

Where n is the number of classes.

Also, accuracy is used as the evaluation metric for APC. Accuracy was obtained by dividing the number of successfully identified reviews by the total number of reviews.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{29}$$

5 Results and Discussion

5.1 Performance of Proposed Multi-Task Model on the Original Dataset

Extensive experiments were conducted on the Arabic benchmark dataset from SemEval-2016 Task 5 Arabic Hotels' reviews dataset, and the proposed model was applied to this dataset to assess the performance of ATE and APC subtasks. A series of experiments are conducted to verify the proposed AraBERT and LCF-ATEPC with variants of local context layers Fusion, CDW, and CDM (AR-LCF-ATEPC-Fusion, AR-LCF-ATEPC-CDW and AR-LCF-ATEPC-CDM) on Arabic ATE and APC. All models were tested on the same training and testing dataset. Table 3 shows the experimental result of the three variants of the proposed model, where Precession (P_{ATE}) Recall (R_{ATE}) and F1 score ($F1_{ATE}$) represent the evaluation metrics of ATE; Precision (P_{APC}) Recall (R_{APC}) and F1 score ($F1_{APC}$) and Accuracy (ACC_{APC}) represent the evaluation metrics of APC.

Table 3: The Experimental Results of LCF-ATEPC + AraBERT model

Model	ATE			APC			
	P_{ATE}	R_{ATE}	$F1_{ATE}$	P_{APC}	R_{APC}	$F1_{APC}$	ACC_{APC}
AR-LCF-ATEPC_Fusion	75.97	75.90	**75.94**	75.09	79.45	**76.74**	**91.5**
AR-LCF-ATEPC_ CWD	77.11	73.93	75.48	75.63	77.33	76.4	91.04
AR-LCF-ATEPC_ CMD	72.64	76.83	74.68	75.89	76.21	76.01	90.55

As shown in Table 3, for the ATE task, the AR-LCF-ATEPC_Fusion model achieved 75.97% of precision, indicating that out of all positive predicted aspect terms, 75.97% of them are extracted correctly; the recall score was 75.90%, indicating that out of all correct aspect terms in the dataset, 75%, 90% were correctly extracted by the model. The higher precision achieved by AR-LCF-ATEPC-CDW was 77.11, which means lower false positive; the recall was lowest precession of all models (73.93%). However, AR-LCF-ATEPC-CMD achieved the lowest precession (72.64) and outperformed the other models in recall score (76.83%). F1 score is considered a better metric than precision and recall when precision and recall results are different as it is a harmonic mean of precision and recall, where it considers false positive and false negative (choosing the best model in this work depends on F1). Therefore, the F1 score provided a better assessment of model performance. Among all models, AR-LCF-ATEPC-Fusion achieved the highest F1 score of 75.94%. The second best-performing model was AR-LCF-ATEPC-CDW, with 75.48 of F1 score. AR-LCF-ATEPC-CDM achieved the lowest F1 score of all models.

In terms of APC, Table 3 shows that all models achieved good accuracy; the best accuracy (91.5%) was achieved by AR-LCF-ATEPC-Fusion model, which indicates that 91.5% of aspect sentiment polarities were classified correctly. The precision for all three models was between 75%–76%, indicating that all models can correctly predict 75%–76% of all correct aspect polarities predicted by the mode. AR-LCF-ATEPC-Fusion achieved the highest recall of all the models. Hence, we employed the F1 measure to select the best model among the three models. Among all models, AR-LCF-ATEPC-Fusion achieved the best results for both tasks, achieving an F1 score of 75.94% for ATE while achieving an F1 score of 76.74 and an accuracy of 91.5% for APC. AR-LCF-ATEPC_CDW was the second-best performing model. However, AR-LCF-ATEPC_CMD performed the worst in both tasks, with an F1 score of 74.36% for ATE and an F1 score of 76.07%, and an accuracy of 90.93% for APC. This is because CDM may lose some important information by masking all words outside the local context. However, CDW assigns low weights to non-local context words that perceive the semantic information in them.

For further performance analysis, the confusion matrix for three models resulting from the experiments is depicted in Tables 4–6. For the AR-LCF-ATEPC_Fusion model, as shown in Table 4, 93.08% of negative aspect terms are accurately classified as negative, while 2.73% and 4.19% were incorrectly classified as neutral and positive, respectively.

Table 4: Confusion matrix of AR-LCF-ATEPC_Fusion model

		Predicted		
		Negative	Neutral	Positive
Actual	Negative	93.08%	2.73%	4.19%
	Neutral	9.1%	52.27%	38.63
	Positive	2.9%	4.1%	93%

Table 5: Confusion matrix of AR-LCF-ATEPC_CDW model

		Predicted		
		Negative	Neutral	Positive
Actual	Negative	94.37%	1.88%	3.75%
	Neutral	18.18%	45.45%	36.36%
	Positive	3.29%	4.54%	92.17%

Table 6: Confusion matrix of AR-LCF-ATEPC_CDM model

		Predicted		
		Negative	Neutral	Positive
Actual	Negative	91.79%	3.03%	5.18%
	Neutral	9.84%	42.62%	47.54%
	Positive	2.31%	3.47%	94.22%

52.27% of the neutral aspect terms are correctly classified as neutral, whereas 47.73% of the neutral aspect polarities were incorrectly classified. 93% of the positive aspect terms are correctly classified as positive, with only 7% was misclassified by AR-LCF-ATEPC-Fusion. As shown in Tables 4–6, from all confusion matrices for all models, we observed the effectiveness of the model in classifying the positive and negative sentiment polarities; we can notice misclassification of the neutral class; This is because there is insufficient annotated data for this class or the difficulty in constructing sentences that contain natural sentiment.

Regarding APC, we found that using different attention layers allowed the model to capture important information from the concatenated local and global context words and learn more interactive aspects and sentence representations. Thus, the proposed multi-task model outperformed the traditional neural network model. we found that using AraBERT as a shared contextual layer plays an important role in extracting local and global semantic features and generating a better representation where AraBERT is large language model can learn contextualized sentiment features in a bidirectional way. Moreover, AraBERT solves Out of Vocabulary (OVV) issues by splitting the unknown words into known sub-words.

As for the overall models' performance, in all experimental results, AR-LCF-ATEPC-Fusion outperformed other models based on LCF and AraBERT for both tasks. This proves the positive effect of concatenating the learned features of CMD and CDW to enrich the features and improve performance. Therefore, AR-LCF-ATEPC-Fusion model will be used for comparison with previous works models and with multi-task model variants on the augmented dataset in terms of F1 for ATE; F1, and accuracy for APC.

To validate the effectiveness of multi-task model, we compared the best multi-task model (AR-LCF-ATEPC-Fusion) with state-of-the-art Deep-based and transformer-based approaches that used the same benchmark dataset: RNN-BiLSTM-CRF [69], BiGRU [70], attention mechanism with neural network [71], BERT [72], and Bert-Flair-BiLSTM/BiGRU-CRF [75], Sequence to Sequence mode for preprocessing and BERT for classification (Seq-seq BERT) [76] and BERT with liner layer (Bert-liner-pair) [77]. The results demonstrated that LCF-ATEPC model outperformed other comparable models. Note none of the previous models for Arabic worked on multitask learning, so the comparisons were with a single task. As shown in Table 7, AR-LCF-ATEPC-Fusion achieved the best results with an F1 score of 75.94% for the ATE task, thereby outperforming all comparison models (except our previous single-task ATE method [75]). For the APC task, AR-LCF-ATEPC-Fusion outperformed all comparison models with an accuracy of 91.5% and an F1 score of 76.74%, improving the accuracy by 2%. This proves the effectiveness of the multi-task model to boost performance; the multi-task learning allows the propagation of information between ATE and APC, so they complement each other and can capture the semantic alignment between them to improve performance. In addition, utilizing LCF and AraBERT improved the ATE and APC results by capturing rich and complex features.

Table 7: Comparison of LCF-ATEPC + AraBERT model on the original dataset with existing methods

Model	$F1_{ATE}$	$F1_{APC}$	ACC_{APC}
Bi-LSTM-CRF for ATE and LSTM-PC for APC [69]	69.9	-	82.6
Attention-Based Neural Model [71]	72.8	-	-
APC using BERT [72]	-	-	89.51
BiGRU-CNN-CRF for ATE IAN-GRU for APC [70]	70.67	-	83.98
Flair-BERT-BiLSTM-CRF [75]	79.7	-	-
Seq-Seq-BERT [76]	-	-	84.65
BERT-liner-pair [77]	-	-	89.51
AR-LCF-ATEPC_Fusion	**75.94**	**76.74**	**91.5**

For the importance of the SRD threshold to identify the local context, we evaluate the effectiveness of different SRD thresholds, study their impact on model performance, and evaluate the best SRD threshold for all variations of LCF layers (i.e., CDM, CDW, and Fusion) on model performance (Fig. 5). Here, we fixed all hyperparameters with $SRD_{threshold}$ ranges from 0–9. As shown in Fig. 5, the highest F-1 score of ATE and the accuracy of APC were obtained via AR-LCF-ATEPC_Fusion model when $\text{SRD}_{\text{threshold}} = 5$ and $\text{SRD}_{\text{threshold}} = 1$. The highest F1-score of ATE and the accuracy of ATE values of the AR-LCF-ATEPC-CDW method were achieved with as $\text{SRD}_{\text{threshold}} = 8$. Finally, the best F1-score of ATE and the accuracy of APC were obtained with $\text{SRD}_{\text{threshold}} = 8$. In general, these results demonstrate that varying the $\text{SRD}_{\text{threshold}}$ value has a slight effect on the results obtained on the experimental dataset.

5.2 Performance of Proposed Multi-Task Model with Augmentation Techniques

The proposed data augmentation techniques based on word and contextual embedding augmentation were evaluated using the proposed LCF-ATEPC model on the three combined datasets. The first two datasets are combinations of the original and augmented datasets generated by word embedding techniques FastText and AraVec, respectively. The third dataset is a combination of the original and augmented dataset generated using AraBERT. The results are shown in Table 8 and illustrated in Fig. 6, which explains the effectiveness of the augmented dataset. Here, the effectiveness of each augmentation technique was assessed based on the improvement of the F1-score of ATE and that accuracy and F1-score of APC tasks. To demonstrate the effectiveness of the proposed data augmentation techniques, This paper compared the performance of LCF-ATEPC with and without applying the data augmentation techniques. Extensive experiments were then conducted to verify the performance of the LCF-ATEPC model + AraBERTv02, different LCF layers (i.e., fusion, CDW, or CDM), and different data augmentation techniques (i.e., word embedding and AraBERT) in the Arabic ATE and APC tasks. We found that the models based on AR-LCF-ATEPC-Fusion on the combined first dataset (with the augmented dataset by using FastText) achieved the best results. However, AR-LCF-ATEPC-CDW on the combined dataset (augmented dataset using AraBERT) achieved the lowest performance.

Table 8: Comparison of LCF-ATEPC + AraBERT model with and without data augmentation techniques (w/o: without, DA: data augmentation)

Model	F1_{ATE}	F1_{APC}	ACC_{APC}
AR-LCF-ATEPC-Fusion (w/o DA)	75.94	76.74	91.5
AR-LCF-ATEPC–CDW (w/o DA)	75.48	76.45	91.04
AR-LCF-ATEPC-CDM (w/o DA)	74.68	76.01	90.55
AR-LCF-ATEPC-Fusion(w/DA FastText)	78.56	78.87	93.18
AR-LCF-ATEPC-CDW(w/DA FastText)	77.01	77.42	92.2
AR-LCF-ATEPC-CDM(w/DA FastText)	76.37	77.32	91.2
AR-LCF-ATEPC-Fusion(w/DA AraVec)	76.98	77.49	91.97
AR-LCF-ATEPC-CDW(w/DA AraVec)	76.37	77.29	92.2
AR-LCF-ATEPC-CDM(w/DA AraVec)	76.66	77.13	91.44
AR-LCF-ATEPC-Fusion(w/DA AraBERT)	77.98	77.62	92.36
AR-LCF-ATEPC-CDW(w/DA AraBERT)	76.37	77.33	91.2
AR-LCF-ATEPC-CDM(w/DA AraBERT)	76.66	77.03	91.44

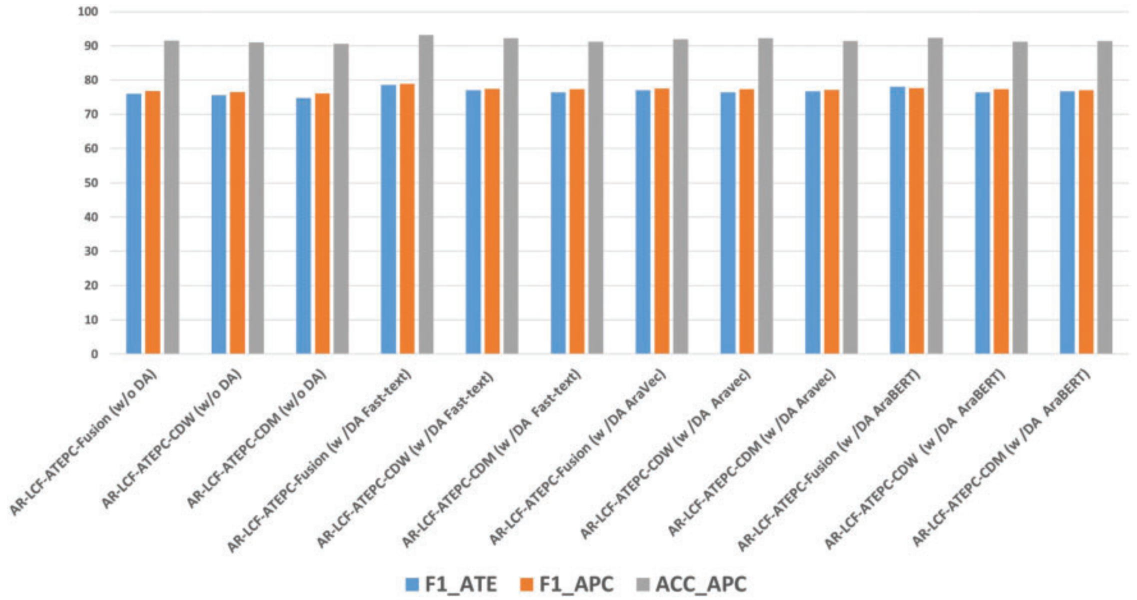


Figure 6: Comparison of LCF-ATEPC + AraBERT model with and without data augmentation techniques

AR-LCF-ATEPC-Fusion achieved the best results with the FastText-based data augmentation, achieving an F1-score of 78.56% for the ATE task. In addition, it achieved an accuracy of 93.18% and F1-score of 78.87% for the APC task, and it outperformed the existing methods and best model without data augmentation (AR-LCF-ATEPC-Fusion) with improvement in accuracy (1.48%) and F1-score (2.05), that prove the effectiveness of utilizing data augmentation with Multi-task model to improve the performance. However, the AR-LCF-ATEPC-Fusion model with AraBERT-based data augmentation was found to be the second-best data augmentation technique among all techniques in terms of the F1-score for the ATE task and accuracy value and F1-score for the APC task.

In contrast to our expectations, we found that the AraBERT-based data augmentation did not obtain the best results as the FastText data augmentation technique (i.e., word embedding). This may have occurred as a result of the fact that words with very different meanings can be predicted and used in a given context, thereby adding some noisy examples. We carefully selected the word suggestions for replacement with original words with the highest score and manually evaluated most of the examples. However, when using word embedding for data augmentation, we compared each original word directly with the candidate replacements and found that we could add constraints to select more accurate words. In addition, we observed a slight performance improvement with the AR-LCF-ATEPC-Fusion model with AraVec-based data augmentation. The other models, i.e., the AR-LCF-ATEPC-CDW and AR-LCF-ATEPC-CDM models, with all augmented data, obtained results that also showed a slight performance improvement compared to those on the original dataset.

6 Conclusion and Future Work

Existing methods treat the Arabic ATE and APC separately as single tasks or solve the problem in a pipeline manner (using independent models for each task). This paper proposes a multi-task learning model to address the Arabic ATE and APC tasks. The proposed model employs the multi-task learning LCF-ATEPC model with AraBERTv02 as a shared layer for text representation. In addition, the multi-task model employs the MHSA and local context mechanisms. To increase the diversity of the training dataset, this paper apply several data augmentation methods to generate additional training data automatically. The proposed data augmentation technique is based on word embedding substitution and a pre-trained language model (i.e., AraBERTV02). In an extensive set of experiments, we evaluated the proposed multi-task model on the common Arabic Hotels' reviews dataset. Thereafter, we evaluated the proposed model on the combined dataset of the original and augmented datasets. This paper found that the proposed multi-task model outperformed previous methods on the same dataset by achieving the best accuracy of APC. In addition, we found that the FastText-based data augmentation technique (i.e., word embedding) with AR-LCF-ATEPC-Fusion improved the ATE performance by increasing the F1-score from 75.94% to 78.56% and improved APC performance by increasing the accuracy values from 91.5% to 93.18% and an F1-score from 76.74 to 78.87. In the future, we plan to employ multi-task learning with different Arabic domains; and utilize data augmentation with different Arabic domains to reduce the time and effort required to collect new datasets, particularly for models that require large training datasets.

Funding Statement: The authors received no funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Banjar, Z. Ahmed, A. Daud, R. A. Abbasi and H. Dawood, "Aspect-based sentiment analysis for polarity estimation of customer reviews on twitter," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2203–2225, 2021.

[2] R. Bensoltane and T. Zaki, "Aspect-based sentiment analysis: An overview in the use of arabic language," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 2325–2363, 2023.

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

[4] A. Diwali, K. Dashtipour, K. Saeedi, M. Gogate, E. Cambria *et al.,* "Arabic sentiment analysis using dependency-based rules and deep neural networks," *Applied Soft Computing*, vol. 127, pp. 109377, 2022.

[5] A. Akbik, D. Blythe and R. Vollgraf, "Contextual string embeddings for sequence labeling for sequence labeling," in *Proc. of the 27th Int. Conf. on Computational Llinguistics*, Santa Fe, New Mexico, USA, pp. 1638–1649, 2018.

[6] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, pp. 4171–4186, 2019.

[7] X. Li, L. Bing, P. Li, W. Lam and Z. Yang, "Aspect term extraction with history attention and selective transformation," in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, pp. 4194–4200, 2018.

[8] H. Gandhi and V. Attar, "Extracting aspect terms using CRF and Bi-LSTM models," *Procedia Computer Science*, vol. 167, no. 2019, pp. 2486–2495, 2020.

[9] Z. Gao, A. Feng, X. Song and X. Wu, "Target-dependent sentiment classification with BERT," *IEEE Access*, vol. 7, pp. 154290–154299, 2019.

[10] C. Sun, L. Huang and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in *Proc. the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL)*, Minneapolis, Minnesota, USA, pp. 380–385, 2019.

[11] X. Li, L. Bing, W. Lam and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. the 56th Annual Meeting of the Association for Computational Linguistic*, Melbourne, Australia, pp. 1109–1114, 2018.

[12] M. Hu, Y. Peng, Z. Huang, D. Li and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *Pro. of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Florence, Italy, pp. 537–546, 2019.

[13] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 Int. Interdisciplinary PhD Workshop (IIPhDW)*, Swinoujście, Poland, pp. 117–122, 2018.

[14] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," arXiv preprint arXiv:1901.11196, 2019.

[15] T. Mikolov, I. Sutskever and K. Chen, "Distributed representations of Words and phrases and their compositionality tomas," arXiv preprint arXiv:1310.4546, 2013.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei *et al.,* "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, pp. 9, 2019.

[17] M. El-Masri, N. Altrabsheh and H. Mansour, "Successes and challenges of arabic sentiment analysis research: A literature review," *Social Network Analysis and Mining*, vol. 7, no. 1, pp. 1–22, 2017.

[18] T. H. Alwaneen, A. M. Azmi, H. A. Aboalsamh, E. Cambria and A. Hussain, "Arabic question answering system: A survey," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 207–253, 2022.

[19] O. Oueslati, E. Cambria, M. Ben HajHmida and H. Ounelli, "A review of sentiment analysis research in arabic language," *Future Generation Computer Systems*, vol. 112, pp. 408–430, 2020.

[20] H. Yang, B. Zeng, J. Yang, Y. Song and R. Xu, "A Multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction," *Neurocomputing*, vol. 419, pp. 344–356, 2021.

[21] W. Antoun, F. Baly and H. Hajj, "AraBERT: Transformer-based model for arabic language understanding," in *Proc. of the 12th Language Resources and Evaluation Conf. (LREC 2020 Workshop)*, Marseille, France, pp. 9–15, 2020.

[22] B. Zeng, H. Yang, R. Xu, W. Zhou and X. Han, "Lcf: A local context focus mechanism for aspect-based sentiment classification," *Applied Sciences*, vol. 9, no. 16, pp. 3389, 2019.

[23] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar *et al.,* "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. of the 10th Int. Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, USA, pp. 19–30, 2016.

[24] S. Poria, E. Cambria, L. -W. Ku, C. Gui and A. Gelbukh, "A Rule-based approach to aspect extraction from product reviews," in *Proc. of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, Dublin, Ireland, pp. 28–37, 2015.

[25] T. A. Rana and Y. N. Cheah, "A Two-fold rule-based model for aspect extraction," *Expert Systems with Applications*, vol. 89, pp. 273–285, 2017.

[26] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng *et al.,* "Red opal: Product-feature scoring from reviews," in *Proc. of the 8th ACM Conf. on Electronic Commerce*, San Diego, California, USA, pp. 182–191, 2007.

[27] B. Ma, D. Zhang, Z. Yan and T. Kim, "An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews," *Journal of Electronic Commerce Research*, vol. 14, no. 4, pp. 304, 2013.

[28] M. T. Khan, M. Durrani and S. Khalid, "Aspect-based sentiment analysis on a large-scale data: Topic models are the preferred solution," *Bahria University Journal of Information & Communication Technologies*, vol. 8, no. 2, pp. 22–27, 2015.

[29] A. S. Manek, P. D. Shenoy, M. C. Mohan and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, 2017.

[30] Y. Xiang, H. He and J. Zheng, "Aspect term extraction based on MFE-CRF," *Information*, vol. 9, no. 8, pp. 1–15, 2018.

[31] H. Wen and J. Zhao, “Aspect term extraction of E-commerce comments based on model ensemble,” in *Proc. 2017 14th Int. Computer Conf. on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, pp. 24–27, 2017.

[32] A. Rietzler, S. Stabinger, P. Opitz and S. Engl, “Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification,” in *Proc. of the 12th Language Resources and Evaluation Conf.*, Marseille, France, pp. 4933–4941, 2020.

[33] A. Giannakopoulos, C. Musat, A. Hossmann and M. Baeriswyl, “Unsupervised aspect term extraction with b-lstm & crf using automatically labelled datasets,” arXiv preprint arXiv:1709.05094, 2017.

[34] P. Barnaghi, G. Kontonatsios, N. Bessis and Y. Korkontzelos, “Aspect extraction from reviews using convolutional neural networks and embeddings,” in *Proc. of 24th Int. Conf. on Applications of Natural Language to Information Systems, (NLDB 2019)*, Salford, UK, pp. 409–415, 2019.

[35] H. Xu, B. Liu, L. Shu and S. Y. Philip, “Double embeddings and CNN-based sequence labeling for aspect extraction,” in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 592–598, 2018.

[36] P. Chen, Z. Sun, L. Bing and W. Yang, “Recurrent attention network on memory for aspect sentiment analysis,” in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 452–461, 2017.

[37] W. Wang, S. J. Pan, D. Dahlmeier and X. Xiao, “Coupled multi-layer attentions for co-extraction of aspect and opinion terms,” in *Proc. of the Thirty-First Association for the Advancement of Artificial Intelligence (AAAI) Conf. on Artificial Intelligence*, San Francisco, California, USA, pp. 3316–3322, 2017.

[38] A. Kumar, A. S. Veerubhotla, V. T. Narapareddy, V. Aruru, L. B. M. Neti *et al.,* “Aspect term extraction for opinion mining using a hierarchical self-attention network,” *Neurocomputing*, vol. 465, pp. 195–204, 2021.

[39] B. N. Dos Santos, R. M. Marcacini and S. O. Rezende, “Multi-domain aspect extraction using bidirectional encoder representations from transformers,” *IEEE Access*, vol. 9, pp. 91604–91613, 2021.

[40] E. Lopes, U. Correa and L. Freitas, “Exploring BERT for aspect extraction in Portuguese language,” in *Int. FLAIRS Conf. Proc.*, North Miami Beach, Florida, USA, vol. 34, pp. 1–4. 2021.

[41] Y. A. Winatmoko, A. A. Septiandri and A. P. Sutiono, “Aspect and opinion term extraction for hotel reviews using transfer learning and auxiliary labels,” arXiv preprint arXiv:1909.11879, 2019.

[42] D. Tang, B. Qin, X. Feng and T. Liu, “Effective LSTMs for target-dependent sentiment classification,” in *Proc. of COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 3298–3307, 2016.

[43] Y. Wang, M. Huang, X. Zhu and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing*, Austin, Texas, USA, pp. 606–615, 2016.

[44] W. Meng, Y. Wei, P. Liu, Z. Zhu and H. Yin, “Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM,” *IEEE Access*, vol. 7, pp. 167240–167249, 2019.

[45] B. Liang, H. Su, L. Gui, E. Cambria and R. Xu, “Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional,” *Knowledge-Based Systems*, vol. 235, pp. 107643, 2022.

[46] X. Li, L. Bing, W. Zhang and W. Lam, “Exploiting BERT for end-to-end aspect-ased sentiment analysis∗,” arXiv preprint arXiv:1910.00883, 2019.

[47] H. Xu, B. Liu, L. Shu and P. S. Yu, “BERT Post-training for review reading comprehension and aspect-based sentiment analysis,” in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, vol. 1, pp. 2324–2335, 2019.

[48] M. Schmitt, S. Steinheber, K. Schreiber and B. Roth, “Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks,” in *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 1109–1114, 2018.

[49] H. Nguyen and K. Shirai, "A joint model of term extraction and polarity classification for aspect-based sentiment analysis," in *2018 10th Int. Conf. on Knowledge and Systems Engineering (KSE)*, Springer, Cham, pp. 323–328, 2018.

[50] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos *et al.,* "Semeval-2014 task 4: Aspect based sentiment analysis," in *Proc. of the 8th Int. Workshop on Semantic Evaluation*, Dublin, Ireland, pp. 27–35, 2014.

[51] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proc. of the 9th Int. Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, USA, pp. 486–495, 2015.

[52] Y. Wang, Q. Chen and W. Wang, "Multi-task BERT for aspect-based sentiment analysis," in *2021 IEEE Int. Conf. on Smart Computing (SMARTCOMP)*, Irvine, CA, USA, pp. 383–385, 2021.

[53] M. S. Akhtar, T. Garg and A. Ekbal, "Multi-task learning for aspect term extraction and aspect sentiment classification," *Neurocomputing*, vol. 398, pp. 247–256, 2020.

[54] S. Van Thin, D. -V. Nguyen, K. Van Nguyen, N. L. -T. Nguyen and A. H. -T. Nguyen, "Multi-task learning for aspect and polarity recognition on Vietnamese datasets," in *16th Int. Conf. of the Pacific Association for Computational Linguistics*, Springer, Singapore, pp. 169–180, 2019.

[55] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conf. on Computer Vision and Pattern Recognition (CVRP)*, Providence, RI, USA, pp. 3642–3649, 2012.

[56] X. Zhang, J. Zhao and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in Neural Information Processing Systems*, vol. 28, pp. 649–657, 2015.

[57] H. -K. Kung, C. -M. Hsieh, C. -Y. Ho, Y. -C. Tsai, H. -Y. Chan *et al.,* "Data-augmented hybrid named entity recognition for disaster management by transfer learning," *Applied Sciences*, vol. 10, no. 12, pp. 4234, 2020.

[58] M. Fadaee, A. Bisazza and C. Monz, "Data augmentation for low-resource neural machine translation," arXiv preprint arXiv: 1705.00440, 2017.

[59] T. Liesting, F. Frasincar and M. M. Truşcǎ, "Data augmentation in a hybrid approach for aspect-based sentiment analysis," in *Proc. of the 36th Annual ACM Symp. on Applied Computing*, New York, NY, USA, pp. 828–835, 2021.

[60] G. Li, H. Wang, Y. Ding, K. Zhou and X. Yan, "Data augmentation for aspect-based sentiment analysis," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 125–133, 2023.

[61] K. Li, C. Chen, X. Quan, Q. Ling and Y. Song, "Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, online, pp. 7056–7066, 2020.

[62] M. Aly and A. Atiya, "Labr: A large scale arabic book reviews dataset," in *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, pp. 494–498, 2013.

[63] I. Obaidat, R. Mohawesh, M. Al-Ayyoub, A. -S. Mohammad and Y. Jararweh, "Enhancing the determination of aspect categories and their polarities in arabic reviews using lexicon-based approaches," in *2015 IEEE Jordan Conf. on Applied Electrical Engineering and Computing Technologies (AEECT)*, Amman, Jordan, pp. 1–6, 2015.

[64] S. Ismail, A. Alsammak and T. Elshishtawy, "A generic approach for extracting aspects and opinions of arabic reviews," in *Proc. of the 10th Int. Conf. on Informatics and Systems*, Giza Egypt, pp. 173–179, 2016.

[65] M. Al-Smadi, M. Al-Ayyoub, H. Al-Sarhan and Y. Jararwell, "An aspect-based sentiment analysis approach to evaluating arabic news affect on readers," *Journal of Universal Computer Science*, vol. 22, no. 5, pp. 630–649, 2016.

[66] 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," *Information Processing & Management*, vol. 56, no. 2, pp. 308–319, 2019.

[67] M. M. Ashi, M. A. Siddiqui and F. Nadeem, "Pre-trained word embeddings for arabic aspect-based sentiment analysis of airline tweets," in *Proc. of the Int. Conf. on Advanced Intelligent Systems and Informatics*, Springer, Cham, pp. 241–251, 2018.

[68] 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," *Journal of Computational Science*, vol. 27, pp. 386–393, 2018.

[69] 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," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 8, pp. 2163–2175, 2019.

[70] M. M. Abdelgwad, T. H. A. Soliman, A. I. Taloba and M. F. Farghaly, "Arabic aspect based sentiment analysis using bidirectional GRU based models," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6652–6662, 2022.

[71] S. Al-Dabet, S. Tedmori, and M. Al-Smadi, "Extracting opinion targets using attention-based neural model," *SN Computer Science*, vol. 1, no. 5, pp. 1–10, 2020.

[72] M. M. Abdelgwad, "Arabic aspect based sentiment analysis using BERT," arXiv preprint arXiv:2107.13290, 2021.

[73] R. Bensoltane and T. Zaki, "Towards arabic aspect-based sentiment analysis: A transfer learning-based approach," *Social Network Analysis and Mining*, vol. 12, no. 1, pp. 1–16, 2022.

[74] S. Behdenna, B. Fatiha and G. Belalem, "Ontology-based approach to enhance explicit aspect extraction in standard arabic reviews," *International Journal of Computing and Digital Systems*, vol. 11, no. 1, pp. 277–287, 2022.

[75] 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.

[76] 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, 2022.

[77] M. M. Abdelgwad, T. H. A. Soliman and A. I. Taloba, "Arabic aspect sentiment polarity classification using BERT," *Journal of Big Data*, vol. 9, no. 1, pp. 1–15, 2022.

[78] R. Bensoltane and T. Zaki, "Combining BERT with TCN-BiGRU for enhancing arabic aspect category detection," *Journal of Intelligent & Fuzzy Systems.*, no. Preprint, pp. 1–14, 2022.

[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5999–6009, 2017.

[80] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu *et al.,* "Deep learning–based text classification: A comprehensive review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–40, 2021.