electronics

MDPI

Article

Multitasking Learning Model Based on Hierarchical Attention Network for Arabic Sentiment Analysis Classification

Muath Alali [1], Nurfadhlina Mohd Sharef [1,2,*], Masrah Azrifah Azmi Murad [1], Hazlina Hamdan [1] and Nor Azura Husin [1]

[1] Intelligent Computing Research Group, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; baniatamuath@gmail.com (M.A.); masrah@upm.edu.my (M.A.A.M.); hazlina@upm.edu.my (H.H.); n_azura@upm.edu.my (N.A.H.)

[2] Laboratory of Computational Statistics and Operational Research, Institute of Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia

* Correspondence: nurfadhlina@upm.edu.my

Abstract: Limited approaches have been applied to Arabic sentiment analysis for a five-point classification problem. These approaches are based on single task learning with a handcrafted feature, which does not provide robust sentence representation. Recently, hierarchical attention networks have performed outstandingly well. However, when training such models as single-task learning, these models do not exhibit superior performance and robust latent feature representation in the case of a small amount of data, specifically on the Arabic language, which is considered a low-resource language. Moreover, these models are based on single task learning and do not consider the related tasks, such as ternary and binary tasks (cross-task transfer). Centered on these shortcomings, we regard five ternary tasks as relative. We propose a multitask learning model based on hierarchical attention network (MTLHAN) to learn the best sentence representation and model generalization, with shared word encoder and attention network across both tasks, by training three-polarity and five-polarity Arabic sentiment analysis tasks alternately and jointly. Experimental results showed outstanding performance of the proposed model, with high accuracy of 83.98%, 87.68%, and 84.59 on LABR, HARD, and BRAD datasets, respectively, and a minimum macro mean absolute error of 0.632% on the Arabic tweets dataset for five-point Arabic sentiment classification problem.

Keywords: Arabic sentiment analysis; multitask learning; ordinal classification; Arabic language

Citation: Alali, M.; Mohd Sharef, N.; Azmi Murad, M.A.; Hamdan, H.; Husin, N.A. Multitasking Learning Model Based on Hierarchical Attention Network for Arabic Sentiment Analysis Classification. *Electronics* **2022**, *11*, 1193. https://doi.org/10.3390/electronics11081193

Academic Editors: Miguel A. Alonso and David Vilares

Received: 27 November 2021
Accepted: 24 December 2021
Published: 9 April 2022

1. Introduction

Sentiment analysis (SA) is a natural language processing (NLP) task that has gained great importance in recent years in the data analysis and information extraction field [1]. The primary objective of SA is to detect sentiments articulated in text and classify the polarities of these sentiments as either binary or ternary polarity. Social media has become the main data source for analyzers to study internet users' expressed opinions on a specific topic, thus allowing them to predict and adjust their strategies.

Opinions expressed in the Arabic language are estimated to populate 5% of the Internet language population [2]. It is also seen as one of the most active up-and-coming languages on the internet in recent years [3]. In the Arabic language, users can convey their views or ideas using either modern Arabic or dialectal Arabic, which can vary from one country to another. Alternatively, both standard and dialectal Arabic are combined on social media. As a result of the morphology, orthography, and complex nature of the language, the detection of sentiment words in Arabic dialects has been found to be particularly challenging. In addition, all Arabic-speaking states have their own dialect, which increases the ambiguity level of the language [4]. That is, many textual contents have become available online, and they are written in the modern standard Arabic (MSA) and informal contexts, with a different meaning for the same word and expression. In addition, the root and the character

of Arabic words can have many forms based on the context, such as (كلمات,كلام,يتكلم). Many words in Arabic also have different meanings with the same spelling depending on their diacritics. Hence, analyzing the sentence's sentiment requires efforts beyond the approaches that only focus on syntactical and semantic features [5].

Moreover, the Arabic dialect is considered a low-resource and non-structured language, making information extraction a difficult task [6]. Furthermore, most tools and resources for MSA do not consider the Arabic dialects' features and are not adapted to them [7]. Besides that, lexical resources such as lexicons are not considered the best method to analyze Arabic sentiment due to the huge number of words from different dialects and the reality of covering all words in the lexicon [2]. Moreover, developing resources and tools for Arabic dialects is considered arduous and time daunting [8].

The existing approaches on Arabic sentiment analysis (ASA) have mainly focused on classifying tweets and reviews in either binary or ternary polarity. Most of these approaches [9–14] are based on handcrafted features, lexicon, and tweet-specific features that are utilized as inputs for machine learning (ML) algorithms, such as support vector machines (SVM), Naive Bayes (NB), multinomial Naive Bayes (MNB), logistic regression (LR), random forest (RF), and clustering. Other approaches utilized a rule-based approach such as the role of lexicalization by developing and prioritizing a set of heuristics rules that could be used in a chaining fashion to classify tweets as negative or positive [15] while Arabic sentiment ontology (ASO) [16] is introduced, which contains sentiment words with varying intensities. The ASO was used to discover user attitudes and classify tweets.

In contrast, deep learning (DL) approaches for SA, such as recurrent neural network (RNN) [17], convolutional neural network (CNN) [18–21], and recursive auto encoder (RAE) [22], have been identified as having the ability to provide superior adaptability and robustness in the past few years by extracting features automatically. However, deep neural network (DNN) approaches in Arabic dialect SA achievement are still limited in number compared with its applications in other areas, including chatbot [23], recommendation systems [24,25], remote sensing [26], and load monitoring [27]. However, most of the approaches applied to ASA focus on binary and ternary classifications.

Therefore, we focus on the problem of the five-polarity ASA in this work. We evaluate our proposed model on four datasets, namely, book reviews from the Large-Scale Arabic Book Reviews Dataset (LABR) [28], Book Reviews in Arabic Dataset (BRAD) [29], Hotel Arabic-Reviews Dataset [30], and Twitter Arabic dialect dataset (SemEval-2017) [31], which are categorized based on five-point scales ranging from highly positive to highly negative. The SemEval-2017 dataset consists of three tasks. Tasks (B) and (A) focus on the binary (positive and negative) and ternary (positive, neutral, and negative) scales. Task (C) concentrates on classifying tweets based on five scales (highly positive to highly negative). Most of the approaches that are applied on these tasks are focused on binary and ternary classifications [12–14,32], while for a five-point classification, only two approaches have addressed this problem, which is based on supervised and unsupervised conventional machine learning [33] and DNN [34] (details will be provided in Section 2).

A five-point polarity scale has low attention in ASA, and only a limited number of studies have tackled this problem. For example, the LABR comprises reviews ranked by users on a scale of 1 to 5. Most of the approaches that use this dataset address binary or ternary classifications, and only limited approaches tackle this problem using traditional ML algorithms, such as MNB and hierarchical classifier [35,36]. Similarly, for BRAD and HARD datasets, the existing approaches on both datasets are based on traditional ML algorithms [29,30]. The deep learning approaches applied on five-polarity classification tasks are distinctly insufficient.

This study mainly aims to manipulate the relation between ASA tasks (ternary and five polarities) and learning them simultaneously. We illustrate the benefit on two domains, namely, tweets and reviews. Multitasking learning (MTL) [37] has demonstrated great potential in various fields, such as human action recognition [38], lane image detection [39], scene classification [40], emotion detection [41], text-video retrieval [42], and image super-

resolution [43]. MTL improves the learning capabilities, encoder quality, and performance of sentiment classification of a conventional single-task classifier by learning the relative tasks in parallel using the shared representation of text sequences [44]. In addition, the key advantage of multitasking learning is that it offers a sophisticated way to use various resources for similar tasks.

For instance, while data can be labeled with distant supervision using emoticons for ternary classification, the fine-grained classification does not offer a straightforward way to achieve it. Learning related tasks such as binary or ternary with five polarities jointly improve encoder quality in producing effective sentence representation and the performance on the five-point task [45,46]. ASA has shown a lack of DL approaches. Moreover, the current works are based on traditional ML algorithms that do not produce a robust latent feature representation [46] and are based on single-task learning. In addition, the reported performance of existing works has a lot of room for improvement.

Our objective in this study is two-fold. First, we propose a multitasking model based on a hierarchical attention network with a shared private layer scheme to transfer and share the common knowledge between two ASA tasks (three and five polarities) during the training. The intent of the proposed model is to learn the significant sentence representation, increase the learning capabilities, and improve the final performance for five-point classification. Second, we evaluate and investigate the performance of two multitasking training techniques by alternate and joint training.

To the best of our knowledge, no study has used MTL for learning five-point ASA classification. The existing approaches that have tackled this classification problem are based on single-task learning. Moreover, a noticeable gap has been observed in the DL approaches applied to this task. In summary, our contributions are as follows:

- The model proposed in this study is the first model that adopts MTL for ASA. The multitask learning model based on a hierarchical attention network (MTLHAN) is developed to exploit the relation between three and five polarities in ASA using a shared private layer. We show how learning two tasks (binary and ternary classification) simultaneously in MTL improves the text representation capability for each task and increases the usability of features. ASA has been demonstrated to lack DL approaches, particularly on the five-polarity classification. The existing DL works are based on single-task learning. In contrast, traditional ML algorithms based on extracted features are considered laborious and time consuming.
- We propose a shared private layer consisting of a word encoder and word attention networks between ASA classification tasks, which add greater flexibility to share complementary features between tasks.
- Through the results obtained from the experiment, the proposed MTLHAN model has been identified to achieve a lower macro average mean absolute error (MAEM) and greater accuracy (ACC) compared with benchmark approaches.

The rest of the paper is organized as follows. The related works are briefly presented in Section 2. Then, we propose and discuss the MTL-HAN model in detail in Section 3. Section 4 presents the current study's results and interpretations. Finally, we discuss the conclusion of our research in Section 5.

2. Related Works

Most of the approaches applied to the Arabic dialect sentiment classification observed are based on traditional ML. Only three classifiers have regularly demonstrated exceptional results: SVM, k-nearest neighbor (KNN), and NB. The combination of bigrams feature and stemmer with term frequency–inverse document frequency (TF–IDF) functioned as a weighting schema to classify the tweets in Jordanian dialect using SVM and NB. They identified that an SVM with these combi-nations outperforms NB [47]. Additionally, the SVM was used to classify 3015 Arabic tweets from the TAGREED corpus [48]. Meanwhile, Meanwhile, three classifiers, namely, SVM, NB, and KNN, with various features and preprocessing steps have been used [49] to study the impact of preprocessing techniques and

n-gram features on the performance of ASA classification. They found that preprocessing, bigram, and character gram improves performance. Moreover, [50] applied SVM on tweets written in Arabic dialect without the preprocessing step; their method achieved an accuracy of 96.1%. Conversely, the same classifier removed Latin letters as preprocessing for the text and achieved a lower accuracy of 95%.

Another work [51] developed an ASA tool for Arabic dialects. Reviews from social media were gathered, which included Saudi, Iraqi, Lebanese, Egyptian, Syrian, and Jordanian dialects. Similarly, AraSenTi-Tweet dataset consisting of 17,573 tweets written in MSA and the Saudi dia-lect were presented [52], where the tweets were manually annotated as negative, neu-tral, positive, or mixed. Several ML classifiers, namely, NB, DT, and SVM were evaluated with TF–IDF and stemming as preprocessing on Arabic tweets written in MSA to identify the simple and workable approach for ASA [53]. The experimental results showed that DT achieved the best performance.

Emphasizing the scarcity of available lexicons for Algerian dialect, efforts on lexicon construction with three semi-automatically created lexicons (a nega-tion words lexicon, keywords lexicon, and intensification words lexicon) using MSA dictionary and Egyptian lexicons [54]. Furthermore, they added the polarity for all the lexicons and used a list of common phrases with their polarities and emoticons. Another work [55] presented the first Tunisian Sentiment Analysis Corpus (TSAC) collected from Facebook user comments. TSAC consists of 17,000 comments manually annotated to negative and positive. The previous approaches for ASA mostly focused on feature selection and creating sentiment resources.

DL models have been successfully used for ASA. Long short-term memory (LSTM) and CNN are the most recognized models. Several DL models on ASA, including a deep auto-encoder (DAE), deep belief networks (DBN), a recursive auto-encoder (RAE), and DNN were explored [56]. They used the ArSenl lexicon and bag-of-words [57] as feature vectors for DNN, DAE, and RAE. The results demonstrated that RAE outperformed all other models. The same authors also proposed an improved RAE model to come up with morphological complexity in Arabic text [22]. Opinions from a Twitter dataset on health services were analyzed [58] to investigate the performances of two DL models (DNN and CNN) and compared them with other ML algorithms (NB, LR, and SVM). The DL models demonstrated encouraging results with word embedding, where CNN and DNN had an accuracy of 90% and 85%, respectively. Arabic dialect of tweets and CNN were used with embedding features to address the highly imbalanced dataset [17]. The same team [59] evaluated several models of CNN, CNN-LSTM, and stacked LSTM. They used word embedding (CBOW and skip-gram (SG)) as features with two settings: dynamic and static. The results demonstrated that CNN-LSTM trained by CBOW achieves higher accuracy. However, in its performance, it exhibited sensitivity toward various datasets A character level with the CNN model compared with other ML classifiers, namely, SVM, LR, KNN, NB, decision tree (DT), and RF. CNN outperformed other ML algo-rithms with the highest accuracy of 94.33% [60]. Another work [61] studied two-word embedding (CBOW and SG) models using a corpus of 3.4 billion Arabic words. Then, CNN was used to classify sentiments. A corpus of 100,000 comments written in Algerian dialect on Facebook were collected manually by annotating the collected comments to negative, neutral, or positive [62].They evaluated two neural network models, CNN and MLP. The authors reported the best performance of 89.5% accuracy was achieved by CNN.

One of the tasks in the SemEval-2017 challenge that utilized the Arabic Twitter dialect dataset ASA was created by [31]. The state-of-the-art performance for Task A (three-polarity) [12] used NB with several features, including lexicon scores, word embedding, unigrams, lexical features (positive word, negative word, emoticon, question, question mark, negative and positive word numbers, and a flag to show that the tweet ends or begins with the hashtag), and bigrams. Furthermore, numerous features were extracted [13], including word embedding, bag of negated words, lexicons, POS, and POS with bigrams to enrich the sentiment. Then, SVM was used to classify the sentiment. The approaches

here were mainly centered on the analysis of features to choose the discriminative features. One study investigated the efficacy of the unsupervised and supervised approaches and their hybrid. Two models that included NB with n-gram features as a supervised model and lexicon features were used [14] to identify the tweet polarity. The researchers applied several weighting schemas (e.g., double and sum polarity) to assign the sentiment weights. Ultimately, the supervised model achieved higher performance.

Another study [33] proposed four models: supervised topic, unsupervised topic, supervised domain, and direct sentiment models. Moreover, the overall accuracy had been decreased by these mixtures of low and high variance features [33]. Most of the above approaches are based on handcrafted features and lexicon features. Using lexicon for Arabic colloquial terms involves high concentricity, as the words can have many scores (sentiment strength). In addition, the process of feature selection and extraction for Arabic dialects is considered time-daunting and extremely arduous to define, which might cause an incomplete specificity or features of the tasks [63]. For Task C (five polarities), limited works have been addressed compared with Task A in the SemEval-2017 challenge as shown in Table 1. Three RNNs with a convolutional network were used [34], where the word 'embedding' is used as a feature. Each CNN network is followed by an RNN; three inputs were used in their model: in and out domain embeddings and the lexicon score of the words. Furthermore, a combination of supervised and unsupervised models were used in [33] to classify the tweet into a five-point scale.

Table 1. A list of approaches that addressed Arabic tweets' SA. The evaluations metrics are macro average recall (P), where higher is better for three polarities, and macro mean absolute error (MAE^M), where lower is better for five polarities.

Model	Method	Polarity	P	MAE^M
NileTMRG [12]	NB	3	0.583	-
SiTAKA [13]	SVM	3	0.550	-
ELiRF-UPV [34]	RCNN	3 5	0.478 -	- 1.264
Tw-StAR [14]	NB	3	0.431	
INGEOTEC [32]	GA	3	0.477	
OMAM [33]	LR Unsupervised and supervised	3 5	0.438 -	- 0.943
NCNN [19]	CNN	3 5	0.620	- 0.914

ASA studies on five-point scales have gained the least popularity compared with other classification tasks of ternary and binary polarity. In addition, most of the approaches applied to this dataset are based on conventional ML algorithms, for example, the LABR dataset where the reviews were labeled from 1–5 (high negative to high positive). The lexicon-based approach and the corpus-based approach were experimented with n-gram features and evaluated several ML algorithms, including SVM, NB, stochastic gradient descent (SGD), passive aggressive (PA), LR, KNN, and perceptron [64]. The impact of balancing and stemming on the LABR dataset using ML classifiers with bag-of-words was studied [65].

Hierarchical classifier (HC) structures have been proposed in [35] to handle a five-polarity classification problem. The HC model is based on the divide-and-conquer approach in which the five classes are divided into subproblems, where each node exemplifies a different classification subproblem. They found that hierarchical classifiers can outperform the flat classifier (FC). The same team [36] suggested an enhanced version of the previous model by studying six different HC structures. They compared these structures with four ML classifiers (DT, KNN, NB, and SVM). The results revealed that the proposed HC

enhanced the performance. However, not all HC structures outperform FC, whereas most HCs decrease the accuracy. The above-mentioned approaches have shown a noticeable lack of DL approaches applied on the LABR fine-grained dataset. All the approaches proposed are based on ML algorithms. Table 2 summarizes the approaches applied to the LABR dataset. The HC [36] is the best existing work on the imbalanced dataset, and MNB [65] is the best existing work on the balanced dataset to date.

Table 2. Approaches applied to the LABR dataset.

Method	Polarity	Acc	Dataset
MNB	5	42.6%	balanced
SVM	5	50.3%	imbalanced
MNB	5	45.0%	imbalanced
HC	5	57.8%	imbalanced
HC	5	72.64%	imbalanced

Several researchers created their own datasets in the style of the LABR dataset. The Book Reviews in Arabic Dataset (BRAD 1.0) [29] consists of 510,600 reviews, where the reviews were labeled from 1–5 (high negative to high positive).Several classifiers have been examined, including SVM, LR, PA, and perceptron with n-gram features. They found LR and SVM achieved higher performance than perceptron [29]. Similarly, the Hotel Arabic-Reviews Dataset (HARD) [30] consists of 409,562 reviews labeled from 1–5 (high negative to high positive).They examined six sentiment classifiers, including AdaBoost, random forest (RF), PA, LR, SVM, and perceptron. They found SVM and LR produced the best performance with unigram and bigram features. Table 3 summarizes the approaches that applied on BRAD and HARD. The LR [30] is the best existing work on the HARD dataset, and LR [29] is the best existing work on the BRAD dataset.

Table 3. Approaches applied on HARD and BRAD.

Model	Dataset	Features	Polarity	Acc	F1-Score
LR [30]	HARD	N-gram, TFIDF	5	76.1%	75.9%
LR [29]	BRAD	N-gram, TFIDF	5	47.7%	48.9%

Other works have utilized the MTL approach to study the problem of five-point sentiment classification. For example, an MTL model based on a recurrent neural network by learning the ternary and five-point classification tasks jointly [45] consists of BiLSTM followed by one hidden layer.They also used additional features such as counts of elongated words, punctuation symbols, emoticons, and word membership features in sentiment lexicons to enrich the sentence representation. They found that learning the related sentiment classification tasks jointly improved the performance on the five-point task.

Another effort in the same direction [46] exploited the relation between binary and five-point sentiment classification tasks by learning them simultaneously. Their model consisted of LSTM as encoder with variational auto-encoder (VAE) as decoder, where the decoder parameters were shared among both tasks. The results revealed that their proposed model enhanced the performance on the five-point task. Furthermore, adversarial multitasking learning (AMTL) specifically on the encoder's framework, consisting of three LSTM as a sentence encoder, two encoders represented the task-specific layers, and one encoder represented the shared layers [44]. In their work, they added a multi-scale CNN as encoder beside the LSTM, and the output of both encoders was fused and concatenated with the output of the shared encoder to produce the final sentence representation. They found that the MTL model improves the encoder quality and the final performance of sentiment classification. The above-mentioned MTL approaches were applied to the English language. However, there is a lack of MTL and DL approaches usage for five-point ASA, and the

existing works applied on this task are based on single task learning using ML algorithms. Thus, the performance of the current ASA on five polarities could be improved, as the performance is still relatively low.

3. Proposed Approach

This research proposes a multitasking model, in which the goal is to exploit the relation between ASA classification tasks (ternary and five polarities) to improve the final performance for the five-point Arabic sentiment classification problem. Recently, the Arabic literature review [66] emphasized the need to use modernized deep learning techniques in ASA, such as hierarchical attention network (HAN) models [67]. Therefore, we used the hierarchical attention model, as it accommodates and simultaneously learns various classification tasks. Consequently, the proposed MTLHAN is realized on the HAN model, as it accommodates and simultaneously learns various types of classification tasks.

Multitask learning has been demonstrated to deliver a more effective model than single classification tasks. MTL can simultaneously capture the intrinsic relativity of the tasks learned. MTL uses the relatedness and the shared representation with multiple loss functions by learning sentiment classification tasks in parallel, such as five and three polarities, to enhance the representation of features produced on a neural network. The information learned for each task can assist other tasks to learn effectively. An important benefit of MTL is that it offers an excellent way to access resources developed for similar tasks, enhancing the learning performance of the current task and increasing the amount of usable data. During learning, the correlated task-sharing layers can enhance the generalization performance, the pace of learning, and the intelligibility of learned models. A learner can learn several related tasks and, while doing so, use these tasks as an inductive bias for one another, which, in return, enhances the learning of the domain's regularities. This feature allows better learning of the sentiment classification tasks with a minimum amount of training data.

To the best of our knowledge, no research has used MTL for learning a five-point ASA classification. The existing models that have tackled ordinal classification are based on traditional ML algorithms and a few DNNs. These approaches lack the ability to learn the relativity between different tasks. Based on this issue, we propose an MTLHAN to learn text representation for tweets and reviews, with ternary classification and five-point classification in parallel. The proposed MTL model relies on a general hierarchical attention network architecture [67]. However, it accommodates different classification tasks and learns them simultaneously, i.e., ternary and five-polarity classification tasks. The shared private layer scheme in MTL allows for the transfer of the knowledge from a ternary task to a five-point task during the training, thus improving the capabilities of learning on the current task.

Moreover, informative text features are shared between tasks. Knowing that a text sequence is positive in a ternary task narrows the classification decision in a five-point task between high positive and positive. The overall structure of our proposed model, MTLHAN, consists of two parts, as shown in Figure 1, where the first part is the shared private layers and the second is the task-specific layers.

Our objective is to construct an MTL model based on BiLSTM and attention mechanism for learning ternary and five-point classifications in parallel. The function of the proposed model is to learn the mapping $F : X \rightarrow (\hat{X}, Y_1, Y_2)$, where X and $\hat{X}$ are the text input and text sequence prediction, respectively. Y_1 represents three polarity scales, e.g., positive, neutral, and negative, and Y_2 represents five-polarity scales, e.g., high negative, negative, neutral, positive, and high positive. Our model consists of three attention models as follows:

1. Attention model for shared private layer;
2. Attention model for ternary polarities (task-specific layer);
3. Attention model for five polarities (task-specific layer).

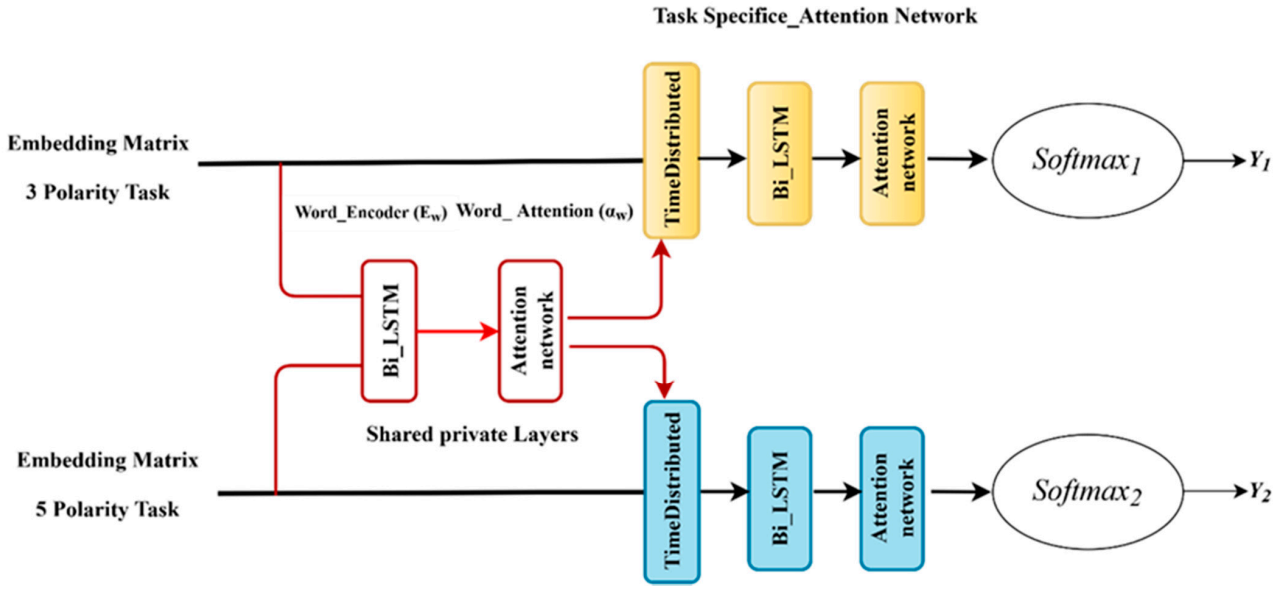

Figure 1. Architecture of the proposed MTLHAN model.

Each attention model consists of BiLSTM and attention networks. Our model intends to learn the significant representation of text sequences for tweets and reviews and improve the final performance for five-point classification. We focused on five-point classification, as it has gained less attention, with limited existing works that have tackled this task in ASA. To enable multitask learning, we propose a distinct way of sharing parameters between tasks. We used BiLSTM as a word encoder (E_w) to obtain the annotation of words followed by word attention (α_w) to distinguish salient features in a given text sequence. Both networks represented the shared private layers. The representations of informative words from (α_w) were aggregated and then fed to task-specific layers, where the yellow and blue boxes in Figure 1 represent ternary classification and fine-grained classification tasks, respectively. Each task structure consisted of BiLSTM on the sentence level (E_s) and attention model α_s on the sentence level. This model has the capability to attain better and competitive performance. In addition, our model can produce a robust latent representation and extract the most important words in a text sequence. The elaboration on each component in the model is provided below.

3.1. Arabic Dialect Encoding with Bi-LSTM

The RNN [68] is a deep network used to process sequential data. It can preserve the previous information on account. However, vanilla RNN cannot handle the long dependencies in input sequences due to the exploding and vanishing gradients problem. Through LSTM, this issue has fortunately been addressed. LSTM can withstand the previous information for long dependencies, thus helping to preserve more information. Therefore, it is the best option in training text classification. LSTM networks consist of four layers that interact in a unique way. These layers include the input gate, forget gate, output gate, and memory cell, which are defined as i_t, f_t, o_t, and c_t. $\hat{g}_t$ is a vector to generate candidate values. We used Bi-LSTM in the shared private layer and the task-specific layer. Below is the computation at each step:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (3)$$

$$\hat{g}_t = tanh\left(W_g x_t + U_g h_{t-1} + b_g\right), \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{5}$$

$$f_t = \sigma\Big(W_f x_t + U_f h_{t-1} + b_f\Big), \tag{6}$$

where x_t is the input at the current time t, σ is the logistic sigmoid function and $\odot$ denotes the element-wise multiplication. Each vector, i_t, f_t, $o_t \in [0,1]^n$ is equal to the dimension of hidden layer h. In our model, we used bidirectional LSTM. Forward LSTM computes the representation of the sentence at every word from the left context $\overrightarrow{h_t}$ and backward LSTM computes the representation of the sentence at every word from the right context $\overleftarrow{h_t}$, which will add useful information. The final representation is obtained by concatenating its left and right context representations $[\overrightarrow{h_t}; \overleftarrow{h_t}]$.

3.2. Word Embedding and Sentence Representation

Word representations, namely, one-hot vectors, are able to attain great performance in text classification [69]. Nevertheless, as the problem is sparse, this method of word encoding encounters the dimensionality problem when used to classify short sentences. Recent research has shown that the continuous representations of words, e.g., word embedding, provide the addition of powerful DL models for SA classification. Typically, these representations encode the syntactic and semantic features of the words. GloVe [70], Word2Vecv [71], and FastText [72] are the three most commonly used pre-trained word embedding methods. One study [20], which evaluated various word embeddings on Arabic dialect SA, discovered that the model CBOW performs better than other models, namely, GloVe and SG. In this study, we used FastText pre-trained word embedding, where the word embeddings are produced by the CBOW model (enhanced by sub-word information and position weights), which generates high-quality word vectors and captures positional information.

The input to MTLHAN were reviews or tweets, where their contents were each treated as a sequence of words. Given a sentence $s = [w_1, w_2, w_3, \ldots\ldots w_m]$ with length m, for each word w_i, we could obtain a low-dimensional distributional vector $w_e \in R^{d_w}$ by look-up operation from $W^{V \times d_w}$, where d_w is the embedding dimension and V is the vocabulary size.

3.3. Shared Private Layers

The core of designing a multitasking network is the scheme for parameter sharing. The shared knowledge scheme enables the sharing and transferring of knowledge between task 1 K_1 and task 2 K_2 by sharing and exploiting the common features between these tasks. The shared parameters help the model learn a universal representation for the text sequence inputs, improve the learning performance on the current tasks, and increase the amount of usable data. The MTLHAN uses Bi-LSTM as a word encoder E_w with parameters H_w and attention mechanism (α_w) with parameters W_w as shared private layers between ternary and five-polarity classification tasks. The shared knowledge can be considered as the hidden states of the word encoder and the attention network, which are denoted as $\theta_{enc,att} = \left\{H_w^{K1}, W_w^{K1}, H_w^{K2}, W_w^{K2}\right\}$ for a given sentence with words w_{it} , $t \,\epsilon [0, T]$. The structure of the shared private layers is depicted in Figure 2.

3.3.1. The Shared Word Encoder

A word encoder is used to obtain word annotations by summarizing information for words from both directions (forward and backward) and therefore incorporating the contextual information in the annotation. The bidirectional LSTM contains the forward $\overrightarrow{E_w}$, which computes the representation of the sentence s_i from $w_{i1,}$ to $w_{it,}$, and backward $\overleftarrow{E_w}$, which computes the representation from $w_{it,}$ to w_{i1} :

$$\overrightarrow{h_{it}^w} = \overrightarrow{E_w}(w_{it}), t \,\epsilon [1, T], \tag{7}$$

$$\overleftarrow{h_{it}^w} = \overleftarrow{E_w}(w_{it}), t \,\epsilon [1, T], \tag{8}$$

For a given word W_{it}, we obtain its annotation by concatenating the forward hidden state $\overrightarrow{h_{it}^{w}}$ and backward hidden state $\overleftarrow{h_{it}^{w}}$, i.e., $h_{it}^{w} = [\overrightarrow{h_{it}^{w}}, \overleftarrow{h_{it}^{w}}]$, which summarizes the information of the whole sentence centered around w_{it} . The shared knowledge can be considered the word encoder's hidden states between ADSA classification tasks, as depicted in Figure 3. Formally, for any sentence in task $k = \{binary, ternary, or\ five - point\}$, the shared hidden representation for each task has been computed as follows:

$$h_{it}^{k} = BiLSTM\ (w_{it}, h_{it}^{k} - 1,\ \theta_{k}), \tag{9}$$

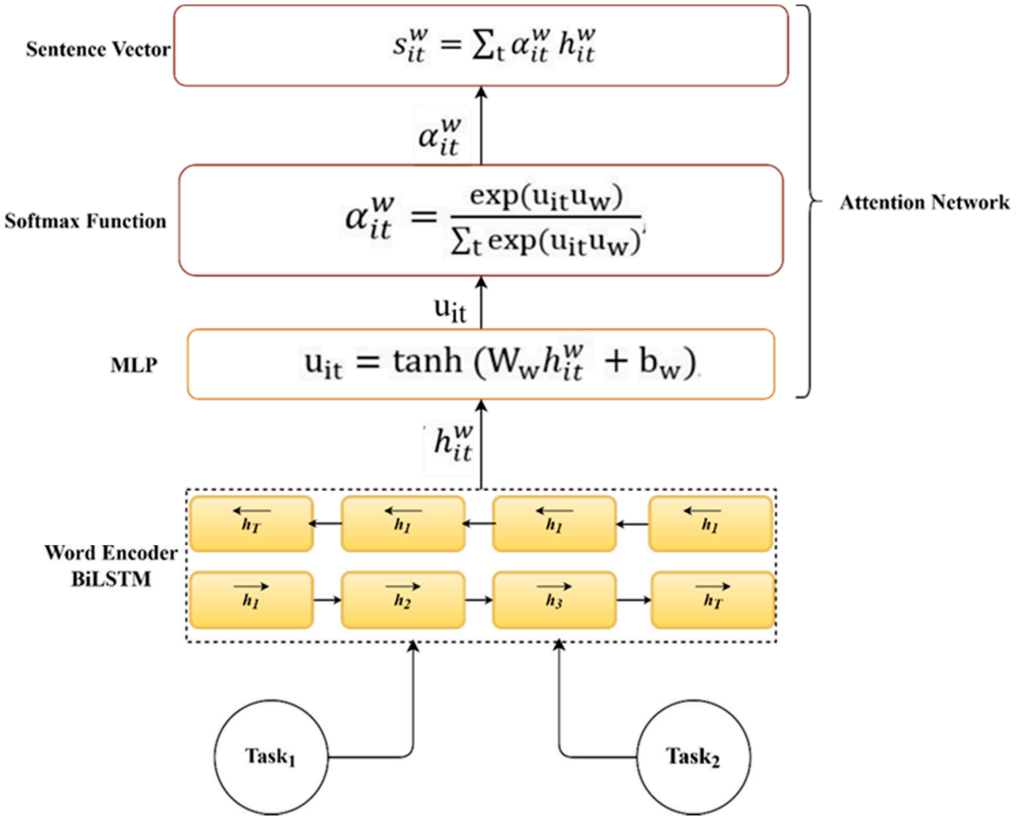

Figure 2. The scheme of the shared parameters in MTLHAN.

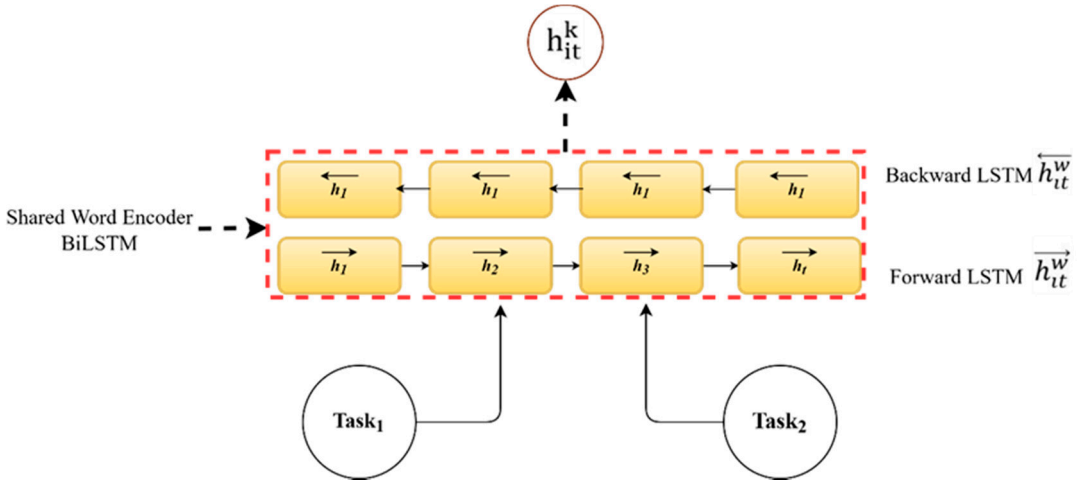

Figure 3. The shared word encoder (Bi-LSTM).

3.3.2. The Shared Attention Networks

The MTLHAN utilizes the attention network on two levels. The word level enables the model to pay less or more attention to words that contribute to the sentence meaning when constructing the representation of sentences. The attention on a sentence level is used to allow the task to learn task-dependent features by rewarding the words that are indicators to classify the sentence accurately.

The word level attention network (α_w) is the second component in the shared private layers in MTLHAN. The shared word encoder can be regarded as a shared feature pool, and the attention mechanism is used to determine the importance of the shared features; besides, these informative features are shared between tasks. Attention mechanisms help to improve the global sentence representation by focusing on and attending to a smaller part of the data [73,74]. The attention mechanism (α_w) is used to extract the important words that contribute to the representation of sentence meaning and then form the sentence vector by aggregating the representation of those informative words. Figure 4 illustrates the components of the shared word level attention network. Specifically, the hidden annotation of words h_{it}^w from the shared word encoder feed through a fully connected neural network (MLP) with parameters W_w.

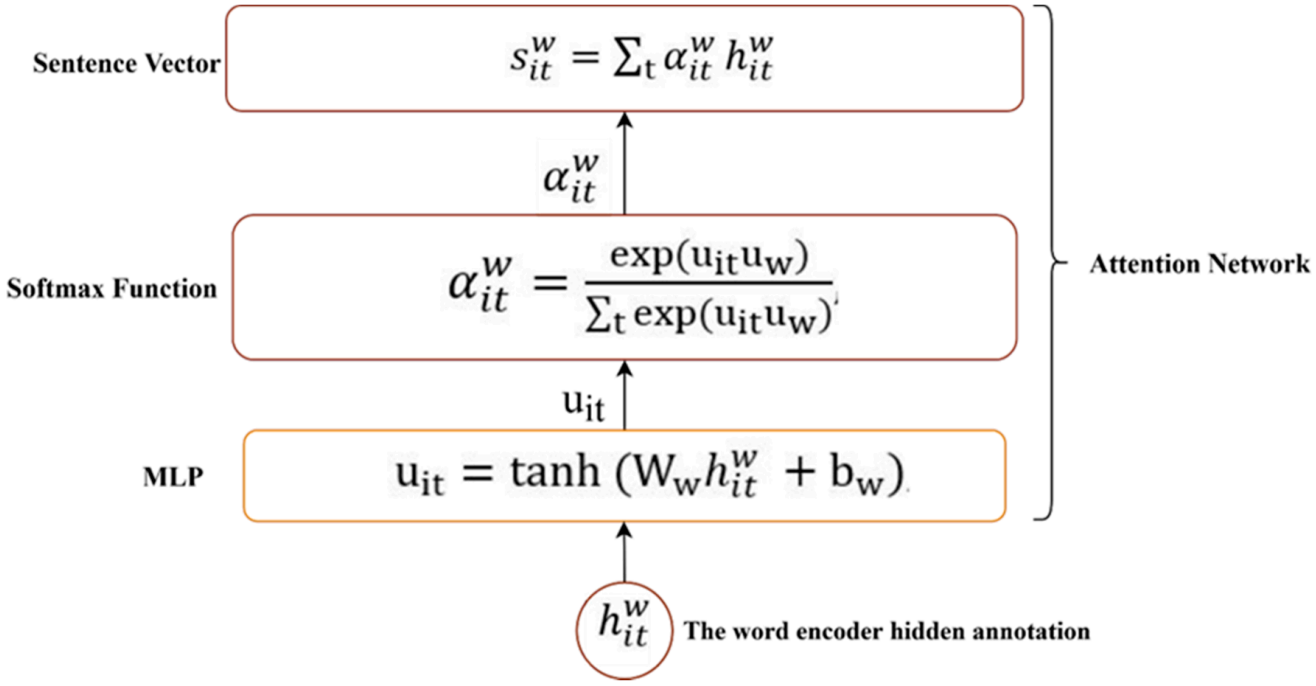

Figure 4. The shared word level attention network.

The idea is to allow the model to learn through training with randomly assigned weights and biases. The new annotations of MLP are represented as u_{it}. u_{it}, which is computed by Equation (10).

$$u_{it} = tanh\ (W_w h_{it}^w + b_w), \tag{10}$$

The importance of a word is measured by multiplied u_{it} with trainable context vector u_w and then passed to the SoftMax function to obtain the normalized importance weights α_{it}^w. The context vector u_w is randomly initialized and learned during the training. α_{it}^w is computed by Equation (11).

$$\alpha_{it}^w = \frac{exp(u_{it}u_w)}{\sum_t exp(u_{it}u_w)}, \tag{11}$$

Subsequently, the sentence vector s_{it}^{w} is produced, as the weighted sum of the word annotations h_{it}^{w} with importance weights α_{it}^{w}, which can be interpreted as a high-level representation of the informative word. s_{it}^{w} is computed by Equation (12).

$$s_{it}^{w} = \sum_{t} \alpha_{it}^{w} h_{it}^{w}, \tag{12}$$

3.4. Task-Specific Layers

Given the sentence vectors s_{it}^{w}, we used two Bi-LSTM as sentence encoders with different parameters; one for ternary classification and another for five-point classification. Similarly, we acquire the text sequence vectors by concatenating $\overrightarrow{h_{it}^{s}}$ and $\overleftarrow{h_{it}^{s}}$ to obtain the annotation of sentence $h_{it}^{s} = [\overrightarrow{h_{it}^{s}}, \overleftarrow{h_{it}^{s}}]$. Formally, for any sentence in task k, we can compute task-specific representation as follows:

$$S_{s}^{k} = BiLSTM\ (w_{s}, h_{it}^{s} - 1,\ \theta_{k}), \tag{13}$$

The attention mechanism α_{s} is also used here to allow the task to learn task-dependent features by rewarding the words that are indicators to classify the sentence accurately. The normalized importance weights of sentences are similarly computed as in Equations (10)–(12). Subsequently, the final sentence vector s_{it}^{s} is produced as the weighted sum of the word annotations.

3.5. Training

To learn the parameters of the proposed MTLHAN model, the cycle of training can be summarized as Algorithm 1. The proposed model trains and learns the ternary and binary classification tasks jointly. For example, HARD dataset, the MTLHAN train, the five-polarity and ternary classification tasks jointly, where $K_1 = HARD_{five} = (X_{(five)}, Y_{(five)})$ and $K_2 = HARD_{ternary} = (X_{ternary}, Y_{ternary})$. In the last layer of task K, the final vector representation s_{it}^{s} is fed into the corresponding SoftMax layers to fit the number of classes.

$$\hat{y}_{(ternary)} = softmax\ (W_{(ternary)}\ s_{it(ternary)}^{s} + b_{(ternary)}), \tag{14}$$

$$\hat{y}_{(five)} = softmax\ (W_{(five)}\ s_{it(five)}^{s} + b_{(five)}), \tag{15}$$

where $\hat{y}_{(ternary)}$ denotes the ternary-classification prediction probabilities, $\hat{y}_{(five)}$ represents the fine-grained prediction probabilities. b and W are the bias and weight to be learned, respectively.

Two techniques have been evaluated for the model training process through alternate [75,76] and joint learning. We can conduct MTL by alternately calling each task loss and optimizer, which means the training process runs for a specified number of iterations on the ternary classification task and then continues to five-polarity classification tasks. Both tasks are trained to reduce cross-entropy. Thus, we acquire:

$$\hat{y}_{(ternary)} = softmax\ (W_{(ternary)}\ s_{it(ternary)}^{s} + b_{(ternary)}), \tag{16}$$

$$\hat{y}_{(five)} = softmax\ (W_{(five)}\ s_{it(five)}^{s} + b_{(five)}), \tag{17}$$

where $\hat{y}_{j}^{i}$ and y_{i}^{j} are the predicted probabilities and ground-true label, respectively. N_1 and N_2 are the number of training samples in the ternary and five-point classification tasks, respectively. To implement the joint training of the ternary and five-point classifications to train the MTLHAN model, we obtain the following global loss function:

$$\text{Total Loss (L)} = \lambda_1 L_{ternary}(\hat{y}, y) + \lambda_2 L_{five}\ (\hat{y}, y), \tag{18}$$

where λ_1 and λ_2 are the weights for ternary and five-point tasks, respectively. The parameters λ_1 and λ_2 are used for balancing both losses by the equal weighting scheme ($\lambda = 1$).

Algorithm 1: Multitask Learning-based Hierarchical Attention

Require: training dataset X ($X_{(ternary)}$, $Y_{(ternary)}$, $X_{(five)}$, $Y_{(five)}$), learning rate l;
Ensure: model Ω: $\{W_{(ternary)}, W_{(five)}, b\}$;
1: Initialize model Ω: $\{W_{(ternary)}, W_{(five)}, b\}$;
2: Repeat
3: Select the ternary task.
4: Select mini-batch samples from task k.
5: Word _encoder (Bi-LSTM(**X**)).
6: Word _attention (α_w)
7: Ternary classification: Bi-LSTM ($X_{(ternary)}$)
8: Attention network ($\alpha_{ternary}$)
9: Five-polarity classification Bi-LSTM ($X_{(five)}$)
10: Attention network (α_{five})
11: **IF** training = = "Jointly" then
12: Calculate the Loss: J(Ω) by Equation (18).
13: **ELSE**
14: **IF** training = ='Alternately" and Task = = "ternary"
15: Calculate the Loss for each task: J(Ω) by Equation (16).
16: **ELSE**
17: Calculate the Loss for each task: J(Ω) by Equation (17).
18: Calculate gradient: ∇(Ω).
19: Update model: Ω = Ω − l∇(Ω).
20: Until maximum iteration

Shared Private layers
Ternary classification task
Five-point classification task

3.6. Datasets

The model was trained on four benchmark datasets. The first one was LABR [64]. The reviews were gathered from Goodreads website and ranked by users on a scale of 1 to 5. They provided two datasets, namely, balanced and imbalanced. Tables 4 and 5 summarize the class distribution for Arabic Book Reviews on imbalanced and balanced datasets, respectively.

The second dataset was the Arabic Twitter dataset provided by SemEval-2017 [31]. The dataset was annotated according to three and five scales. The dataset had multiple dialects, including Levantine, Egyptian, and Gulf. In these dialects, the same word comes in different forms, such as suffixes and prefixes, and holds various definitions. In turn, this variation adds more complexity to the task of classification. In addition, the training data size is very small and highly imbalanced. Table 6 summarizes the polarity distribution for each task.

Table 4. Statistics about imbalanced LABR training and testing dataset.

Dataset	Task	Highly Positive	Positive	Neutral	Negative	Highly Negative	Total
Train	3	-	15,216	9841	4197	-	29,254
	5	19,015	15,216	9841	4197	2337	50,606
Test	3	-	3838	2360	1088	-	7286
	5	4763	3838	2360	1088	602	12,651

Table 5. Statistics about balanced LABR training and testing dataset.

Dataset	Task	Highly Positive	Positive	Neutral	Negative	Highly Negative	Total
Train	3	-	2352	2352	2352	-	7056
	5	2352	2352	2352	2352	2352	11,760
Test	3	-	587	587	587	-	1761
	5	587	587	587	587	587	2935

Table 6. Statistics of tweets training dataset and testing datasets.

Dataset	Task	Highly Positive	Positive	Neutral	Negative	Highly Negative	Total
Train	3	-	743	1470	1142	-	3355
	5	175	884	1699	770	210	3738
Test	3		1514	2364	2222		6100
	5	1	1548	3343	1175	1	6309

The third dataset was BRAD [29]. The reviews were gathered from the same source as the LABR dataset and annotated according to five scales. The fourth dataset was HARD [30]. The reviews were collected from the booking website and annotated to five scales. Tables 7 and 8 summarize the class distribution for the BRAD and HARD datasets.

Table 7. Statistics about BRAD datasets.

Dataset	Task	Highly Positive	Positive	Neutral	Negative	Highly Negative	Total
BRAD	3	-	158,461	106,785	47,133	-	251,001
	5	166,972	158,461	106,785	47,133	31,247	510,598

Table 8. Statistics about HARD balanced and imbalanced datasets.

Dataset	Task	Highly Positive	Positive	Neutral	Negative	Highly Negative	Total
Imbalanced	3	-	132,208	80,326	38,467	-	251,001
	5	144,179	132,208	80,326	38,467	14,382	409,562

3.6.1. Data Preprocessing

We first perform Sentence Breaker (https://github.com/disooqi/ArabicProcessingCog, accessed on 1 November 2021) to break the reviews into sentences. Then the following preprocessing steps were applied to the datasets:

1. Diacritics, punctuation, non-Arabic words and letters, hashtags, and URL were removed.
2. Emoticons were replaced with their meaning.
3. Letters were normalized.
4. Elongated words and Kashida were normalized.

4. Experimental Results and Discussion

4.1. Evaluation Metrics

The official evaluation metric used for the Arabic Twitter dataset, which was provided by [31], is the macro mean absolute error MAE^M, which takes into account the order of the five classes in which each text sequence is to be classified into only one of the classes in the

dataset by numbers i in C = {1,2,3,4,5} with a total given order on C. The MAE^M is defined as [77]:

$$MAE^M(h, Te) = \frac{1}{|c|}\sum_{j=1}^{C}\frac{1}{Te}\sum_{X_i \in Te_j}|h(X_i) - Y_i|, \quad (19)$$

where Te_j is the set of a test sample whose true class is c_j, y_j denotes the true label of item x_i, $h(x_i)$ is its predicted label, and $|h(x_i - y_i)|$ denotes the "distance" between classes $h(x_i)$ and y_j.

The total accuracy is used as the evaluation metric for LABR datasets (balanced and imbalanced), which can be defined as:

$$\text{Accuracy} = \frac{\text{N}_{\text{correct}}}{\text{N}} \quad (20)$$

where N is the number of test datasets and $\text{N}_{\text{correct}}$ is the number of correct classifications.

4.2. Benchmark Approaches

The proposed MTLHAN model has been compared with the existing state-of-the-art approaches on the five-point dataset of Arabic dialect tweets:

- NCNN: narrow convolutional neural network structure trained on top of word embedding. The model consists of three convolutional layers, each one followed by max-pooling proposed by [18].
- CRNN: three convolutional layers followed with an RNN; three inputs were used in their model: the in and out domain embeddings and the lexicon score of the word, which were proposed by [34].

The proposed MTLHAN model has also been compared with benchmark approaches on LABR datasets:

- SVM: support vector machine classifier with n-gram feature proposed by [65].
- MNB: multinomial Naive Bayes with bag-of-words features proposed by [64].
- HC: hierarchical classifiers model based on the divide-and-conquer approach proposed by [35].
- HC(KNN): improved hierarchical classifiers model based on the divide-and-conquer approach proposed by [36].

Similarly, for the BRAD and HARD datasets, we compare our model with the following benchmark approaches:

- BRAD dataset: logistic regression with unigrams, bigrams, and TF–IDF proposed by [29].
- HARD dataset: logistic regression with unigrams, bigrams, and TF–IDF proposed by [30].

The Bidirectional Encoder Representation from Transformers (BERT) [78] recently achieved outstanding performance in the NLP tasks. In the proposed architecture, Arabic pre-trained BERT models(AraBERT) were proposed [79] which was trained on three corpora, namely, Arabic Wikipedia, Modern Standard Arabic Corpus (1.5 billion words), and OSIAN [80]. We also compare our proposed model with AraBERT, consisting of 12 encoder layers, 12 attention heads, and 768 hidden dimensions.

4.3. Experiment Setting

We have followed the official split for Arabic dialect tweets and LABR datasets. For BRAD and HARD datasets, we followed the baseline approaches for both datasets, where these datasets split into 80% for training and 20% for testing. Therefore, the hyperparameters for optimizations were chosen empirically: (i) Adam optimizer was used to train each task in the MTLHAN; (ii) batch size is 50, number of epochs was set to 60 with early stopping, the dropout rate was set to (0.1 to 0.3) for regularizing the networks; (iii) the dimension of pre-trained word embedding is 300, the hidden layer size of BiLSTM was set

to 150; (iv) as mentioned in Section 3.6.1, the sentence breaker has been used to break the reviews into a sentence, therefore, the max length of the sentence was set to 200 for the BRAD and HARD datasets, and 50 for the HARD and Arabic dialect tweet datasets. The max number of sentences was set to 9 for the BRAD and LABR datasets, and 5 and 1 for the HARD and Arabic dialects tweets, respectively.

The training data were highly imbalanced; some classes had more training samples than others, thus introducing bias in our proposed model. We used the class weights method introduced in [81] to penalize the errors made on the rare classes more to deal with this problem. We obtained class weights with $CW_i = \frac{\max(x)}{x_i + \alpha \times \max(x)}$, where x is a vector with class counts. α is the smoothing factor used to smooth out the class weights in case of very strong imbalances in the training data (which otherwise could lead to exceedingly large class weights). Keras, TensorFlow, and scikit-learn [82] were used to perform all the experiments.

4.4. Results and Findings

The experimental results are promising, and they demonstrate the superiority of the proposed model over benchmark approaches. To evaluate our proposed MTLHAN model, the model is first compared with two multitasking learning techniques jointly and alternately. The majority of datasets that have been used in this work are on review level (LABR, BRAD, and HARD), and only one dataset on sentence level (Arabic dialect tweets). Therefore, LABR and Arabic dialect tweets have been selected to evaluate both MTL learning techniques. Table 9 illustrates the performance of joint and alternate training, where the evaluation metrics are the total accuracy for the LABR datasets and the macro mean absolute error for Arabic dialect tweets.

Table 9. Performance of jointly and alternately training for five-polarity classification.

MTLHAN Training	LABR (Imbalanced) ACC	LABR (Balanced) ACC	Tweets MAE^M
Jointly	83.98	76.57	0.632
Alternately	80.86	72.13	0.671

Joint training achieves better performance than alternate training. However, both training techniques outperform the benchmark approaches. Therefore, the joint training performances have only been compared with the benchmarks approach for BRAD, HARD, LABR, and Arabic dialect tweet datasets, as presented in Tables 10–14. The evaluation findings can be summarized as follows:

- On the five-point dataset of Arabic tweets, the MTLHAN achieves the best results ($MAE^M = 0.632$) with a significant difference over current approaches, as depicted in Table 10. The results for current approaches are AraBERT ($MAE^M = 0.801$), NCNN ($MAE^M = 0.914$), OMAM ($MAE^M = 0.43$), and CRNN ($MAE^M = 1.264$).

Table 10. The performance of MTLHAN against benchmark approaches based on SemEval-2017 Arabic dialect tweet dataset.

Model	Polarity	MAE^M
RCNN [34]	5	1.264
OMAM [33]	5	0.943
NCNN [19]	5	0.914
AraBERT [79]	5	0.801
MTLHAN	5	0.632

- On the LABR imbalanced dataset, the MTLHAN outperforms all other approaches with (Acc = 83.98%), as presented in Table 11. The performances of the related work

approaches are MNB (Acc = 45.0%), SVM (Acc = 50.3%), HC (Acc = 57.8%), AraBERT (Acc = 58.96%), and improved HC (Acc = 72.64%).

Table 11. The performance of the MTLHAN compared with benchmark approaches on the LABR imbalanced dataset.

Model	Polarity	ACC	F1-Score
SVM [65]	5	50.3%	49.1%
MNB [64]	5	45.0%	42.8%
HC (KNN) [35]	5	57.8%	63.0%
AraBERT [79]	5	58.96%	55.88%
HC (KNN) [36]	5	72.64%	74.82%
MTLHAN	5	83.98%	80.81%

- The MTLHAN achieves the best performance (Acc = 76.57%) with a large difference over the benchmark approach, as depicted in Table 12. The performance of the benchmark approach is MNB (Acc = 42.6%).

Table 12. The performance of the MTLHAN compared with benchmark approaches on LABR balanced dataset.

Model	Polarity	Acc
MNB [64]	5	42.6%
AraBERT [79]	5	56.8%
MTLHAN	5	76.57%

- On HARD dataset, the performance of MTLHAN exceeds other approaches with (Acc = 87.68%), as depicted in Table 13.

Table 13. The performance of the MTLHAN compared with benchmark approaches on the HARD dataset.

Model	Polarity	ACC	F1-Score
LR [1]	5	76.1%	75.9%
AraBERT [79]	5	80.85%	77.88%
MTLHAN	5	87.68%	84.56%

- On the BRAD dataset, the performance of MTLHAN achieves higher performance (Acc = 84.59%) with significant differences over current approaches, as presented in Table 14.

Table 14. The performance of the MTLHAN compared with benchmark approaches on the BRAD dataset.

Model	Polarity	ACC	F1-Score
LR [29]	5	47.7%	48.9%
AraBERT [79]	5	60.85%	58.79%
MTLHAN	5	84.59%	81.28%

5. Discussion

The evaluation results show that the proposed MTLHAN model with joint and alternate learning achieves superior performance. The performance of joint training is higher

than that of alternate training, with a difference of 3.12% and 5.12% in the LABR balanced and imbalanced datasets, respectively, and 0.39% in the Arabic tweet dataset. The difference in the performance between both methods is that the alternate training is affected by the dataset size of each task. More information will be dominant in the shared private layers when the task has a larger dataset. In some cases, alternate training can easily become biased if one of the tasks has datasets much larger than the other. Therefore, joint training is more preferred in SA tasks. Conversely, alternate training is more suitable if we have two different datasets for each of the different tasks, for example, machine translation tasks translating from Arabic dialect to MSA and MSA to English [75]. By designing a network in an alternate setting, the performance of each task can be improved without having to find more training data [76].

When comparing the proposed MTLHAN model performance with the best performing model in the Arabic tweet dataset, results obtained by MTLHAN outperform AraBERT [79], with a difference of 0.169%. In addition, our model outperforms other approaches on the same dataset, NCNN [19], OMAM [33], and RCNN [34] with clear differences of 0.282%, 0.311%, and 0.632%, respectively. NCNN uses a convolutional network trained on top of word embedding. On the other hand, OMAM uses a combination of supervised and unsupervised models based on ML algorithms and lexicons. However, using this combination does not produce a robust sentence representation [46]. The RCNN surprisingly has the worst performance. Combining the convolutional and Bi-LSTM enables the model to obtain comprehensive representation, namely, the historical, future, and local context of any position in a sentence. Despite the small dataset, given the high complexity and complicated nature of the model, the performances might be affected by over-fitting, which loses the semantic and sentiment representations [34]. Therefore, multitask learning is more suited when the dataset size is small. Learning-related tasks simultaneously increase the amount of usable data, and the risk of over-fitting is reduced [83].

The proposed model results show that the MTLHAN model achieves the best performance on the LABR imbalanced dataset, thus outperforming the HC model [36] with a significant difference of 11.34% on the imbalanced dataset. The HC model is based on the divide-and-conquer approach, where the five classes are divided into subproblems. However, the authors only focus on selecting core classifiers without considering sentence representation. Meanwhile, the other approaches, namely AraBERT [79], SVM [65], and MNB [64], achieved the worst performance on the same dataset. Our proposed model outperforms AraBERT [79] and MNB [64], with a huge difference of 19.77% and 33.97%, respectively, on the balanced dataset. Moreover, our proposed MTLHAN outperforms all competing approaches on the five-polarity classification for the LABR balanced and imbalanced datasets. Similarly, the proposed MTLHAN achieves the best performance on the BRAD imbalanced dataset, thus outperforming the AraBERT [79] and LR [29], with a difference of 23.74% and 36.89%, respectively.

All of the approaches applied on LABR and BRAD datasets are not DL models that have no capabilities of producing a richer sentence representation compared with DL models [46] and are based on single-task learning. Conversely, the performance of AraBERT is very low. This is due to the fact that the model does not learn the decision boundaries between polarity classes well, which is justified by a large number of false-positives and high positives, as well as for negatives and high negatives. Similarly, on the HARD dataset, the MTLHAN outperforms AraBERT and LR [30], with a difference of 6.83% and 11.58%, respectively.

Other relative tasks can improve the performance of the five-point classification. The comparison analysis with benchmark approaches directly elucidates that joint learning in five-polarity classification can learn additional rich feature representations among the text sequence than the single-learning task. This outcome also indicates that joint learning is better suited to solving complex classification tasks and can learn and produce a more robust latent representation in fine-grained tasks for Arabic colloquial SA.

6. Conclusions

The current study has successfully developed the first multitask learning model for five-point classification in Arabic dialect SA. The proposed multitask architecture with shared private parameters helps to improve the global text sequence representation. Moreover, the attention mechanism can extract the most informative words in text sequences. Limited works on colloquial Arabic that are applied to this task are based on single task learning and do not consider the relative tasks. Moreover, these studies are based on conventional ML algorithms, feature selection, and sentiment resources that are time-consuming, arduous, and unable to produce a richer feature representation. In addition, Arabic dialect SA, in particular, are still suffering from a lack of sentiment resources. In contrast, the limited number of deep-neural networks used for this task for the Arabic dialect are based on the single-learning task. These approaches are highly complicated and complex for a small amount of data.

We have conducted several experiments on five-point ASA datasets. We empirically determine the best training technique (alternate and joint) for multitask learning in ASA. The experiment results show that joint learning achieves higher performance than alternate training, as the latter is influenced by the dataset size of each task. The empirical results demonstrate that our model outperforms other state-of-the-art approaches on three datasets. We have found that we can significantly enhance the performance of five-point classification through jointly learning the tasks of fine-grained and ternary classifications with a multitasking model. By determining that a text is "negative" in the ternary setting, the classification decision between the high negative and negative categories in the five-point setting can be narrowed down.

Furthermore, the performance of fine-grained tasks with joint learning is greater than ternary tasks, thus showing that joint learning is better suited in solving complex classification tasks. In addition, it uses a shared private layer to reduce over-fitting and increase the amount of usable data. This ability demonstrates the effectiveness of our proposed model structure. Our model produces a robust latent feature representation for text sequence. The proposed model is trained end-to-end. Based on the total accuracy of 83.98%%, 87.68%, and 84.59% on the LABR, HARD, and BRAD datasets, respectively, the results of the experiments show that our model enhances the existing state-of-the-art approaches. On the Arabic tweet dataset, the proposed model achieved minimum MAE^M with a performance of 0.632. It is noted that the performance of the proposed model is not comparable with the state-of-art approaches for LABR, HARD, and BRAD, such as LR, MNB, and HC. However, the authors insist that the performance of the five-polarity ASA classification has been largely improved. The time complexity of the proposed MTLHAN is also not the main concern since the SA application can always be performed offline.

Plans for future work include redeveloping a multitask architecture based on transformers and evaluating other multitasking frameworks based on transformers such as the Framework for Adapting Representation Models (FARM) (https://farm.deepset.ai/, accessed on 25 November 2021). Moreover, we plan to incorporate the character level to the proposed approaches and use convolutional neural networks as a sentence level encoder in the proposed approach. Moreover, incorporating the sarcasm detection task is another crucial area to work on to enhance ASA's performance, and to evaluate the proposed model and transformer performance on other domains such as Arabic aspect level sentiment analysis, Arabic text categorization, and Arabic text entailment.

Author Contributions: Conceptualization: M.A. and N.M.S.; methodology, M.A. and N.M.S.; software: M.A.; validation: N.M.S., M.A.A.M. and H.H.; investigation: M.A. and N.M.S.; resources: N.M.S.; writing—original draft preparation: M.A.; writing—review and editing: M.A.A.M., H.H. and N.A.H.; supervision: N.M.S.; project administration: N.M.S.; funding acquisition: N.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research and publication are partly sponsored by the United States Air Force Office of Scientific Research grant FA2386-18-1-4079.

Acknowledgments: This work is part of a study that focuses on Arabic sentiment analysis conducted at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. The authors thanks the funder for the support provided.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pang, B.; Lee, L. *Opinion Mining and Sentiment Analysis, Foundations and Trends® in Information Retrieval*; Now Publishers: Boston, MA, USA, 2008; pp. 1–135. [CrossRef]
2. El-Masri, M.; Altrabsheh, N.; Mansour, H. Successes and challenges of Arabic sentiment analysis research: A literature review. *Soc. Netw. Anal. Min.* **2017**, *7*, 54. [CrossRef]
3. Boudad, N.; Faizi, R.; Thami, R.O.H.; Chiheb, R. Sentiment analysis in Arabic: A review of the literature. *Ain Shams Eng. J.* **2018**, *9*, 2479–2490. [CrossRef]
4. Hazrina, S.; Sharef, N.M.; Ibrahim, H.; Murad, M.A.A.; Noah, S.A.M. Review on the advancements of disambiguation in semantic question answering system. *Inf. Process. Manag.* **2017**, *53*, 52–69. [CrossRef]
5. Sharef, N.M.; Shafazand, Y.M.; Nazri, M.Z.A.; Husin, N.A. Self-adaptive based model for ambiguity resolution of the Linked Data Query for Big Data Analytics. *Int. J. Integr. Eng.* **2018**, *10*, 176–182. [CrossRef]
6. Salloum, S.A.; AlHamad, A.Q.; Al-Emran, M.; Shaalan, K. A survey of Arabic text classification models. *Int. J. Electr. Comput. Eng.* **2018**, *8*, 4352–4355. [CrossRef]
7. Harrat, S.; Meftouh, K.; Smaili, K. Machine translation for Arabic dialects (survey). *Inf. Process. Manag.* **2019**, *56*, 262–273. [CrossRef]
8. Elnagar, A.; Yagi, S.M.; Nassif, A.B.; Shahin, I.; Salloum, S.A. Systematic Literature Review of Dialectal Arabic: Identification and Detection. *IEEE Access* **2021**, *9*, 31010–31042. [CrossRef]
9. Abdul-Mageed, M. Modeling Arabic subjectivity and sentiment in lexical space. *Inf. Process. Manag.* **2019**, *56*, 291–307. [CrossRef]
10. Al-Smadi, M.; Al-Ayyoub, M.; Jararweh, Y.; Qawasmeh, O. Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels' reviews using morphological, syntactic and semantic features. *Inf. Process. Manag.* **2019**, *56*, 308–319. [CrossRef]
11. Baly, R.; Badaro, G.; El-Khoury, G.; Moukalled, R.; Aoun, R.; Hajj, H.; El-Hajj, W.; Habash, N.; Shaban, K.; Diab, M.; et al. A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-Art Opinion Mining Models. In Proceedings of the Third Arabic Natural Language Processing Workshop, Valencia, Spain, 3 April 2017; pp. 110–118. [CrossRef]
12. El-Beltagy, S.R.; El Kalamawy, M.; Soliman, A.B. NileTMRG at SemEval-2017 Task 4: Arabic Sentiment Analysis. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 790–795. [CrossRef]
13. Jabreel, M.; Moreno, A. SiTAKA at SemEval-2017 Task 4: Sentiment Analysis in Twitter Based on a Rich Set of Features. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 692–697.
14. Mulki, H.; Haddad, H.; Gridach, M.; Babaoğlu, I. Tw-StAR at SemEval-2017 Task 4: Sentiment Classification of Arabic Tweets. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 664–669. [CrossRef]
15. Siddiqui, S.; Monem, A.A.; Shaalan, K. Evaluation and enrichment of arabic sentiment analysis. *Stud. Comput. Intell.* **2017**, *740*, 17–34. [CrossRef]
16. Tartir, S.; Abdul-Nabi, I. Semantic Sentiment Analysis in Arabic Social Media. *J. King Saud Univ.—Comput. Inf. Sci.* **2017**, *29*, 229–233. [CrossRef]
17. Al-Azani, S.; El-Alfy, E.-S. Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text. *Procedia Comput. Sci.* **2017**, *109*, 359–366. [CrossRef]
18. Alali, M.; Sharef, N.M.; Hamdan, H.; Murad, M.A.A.; Husin, N.A. Multi-layers convolutional neural network for twitter sentiment ordinal scale classification. *Adv. Intell. Syst. Comput.* **2018**, *700*, 446–454. [CrossRef]
19. Alali, M.; Sharef, N.M.; Murad, M.A.A.; Hamdan, H.; Husin, N.A. Narrow Convolutional Neural Network for Arabic Dialects Polarity Classification. *IEEE Access* **2019**, *7*, 96272–96283. [CrossRef]
20. Gridach, M.; Haddad, H.; Mulki, H. Empirical evaluation of word representations on arabic sentiment analysis. *Commun. Comput. Inf. Sci.* **2018**, *782*, 147–158. [CrossRef]
21. Al Omari, M.; Al-Hajj, M.; Sabra, A.; Hammami, N. Hybrid CNNs-LSTM Deep Analyzer for Arabic Opinion Mining. In Proceedings of the 2019 6th International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 364–368. [CrossRef]
22. Al-Sallab, A.; Baly, R.; Hajj, H.; Shaban, K.; El-Hajj, W.; Badaro, G. AROMA: A Recursive Deep Learning Model for Opinion Mining in Arabic as a Low Resource Language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2017**, *16*, 1–20. [CrossRef]
23. Palasundram, K.; Sharef, N.M.; Nasharuddin, N.A.; Kasmiran, K.A.; Azman, A. Sequence to Sequence Model Performance for Education Chatbot. *Int. J. Emerg. Technol. Learn.* **2019**, *14*, 56–68. [CrossRef]
24. Khairudin, N.; Sharef, N.M.; Noah, S.A.M.; Mustapha, N. Embedded learning for leveraging multi-aspect in rating prediction of personalized recommendation. *Malays. J. Comput. Sci.* **2018**, *31*, 31–47. [CrossRef]

25. Nilashi, M.; Ahani, A.; Esfahani, M.D.; Yadegaridehkordi, E.; Samad, S.; Ibrahim, O.; Sharef, N.M.; Akbari, E. Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach. *J. Clean. Prod.* **2019**, *215*, 767–783. [CrossRef]
26. Xu, R.; Tao, Y.; Lu, Z.; Zhong, Y. Attention-mechanism-containing neural networks for high-resolution remote sensing image classification. *Remote Sens.* **2018**, *10*, 1602. [CrossRef]
27. Salerno, V.M.; Rabbeni, G. An Extreme Learning Machine Approach to Effective Energy Disaggregation. *Electronics* **2018**, *7*, 235. [CrossRef]
28. Nabil, M.; Aly, M.; Atiya, A. LABR: A Large Scale Arabic Sentiment Analysis Benchmark. *arXiv* **2014**, arXiv:1411.6718.
29. Elnagar, A.; Einea, O. BRAD 1.0: Book reviews in Arabic dataset. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016. [CrossRef]
30. Elnagar, A.; Khalifa, Y.S.; Einea, A. Hotel Arabic-reviews dataset construction for sentiment analysis applications. *Stud. Comput. Intell.* **2018**, *740*, 35–52. [CrossRef]
31. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 502–518. [CrossRef]
32. Miranda-Jiménez, S.; Graff, M.; Tellez, E.S.; Moctezuma, D. INGEOTEC at SemEval 2017 Task 4: A B4MSA Ensemble based on Genetic Programming for Twitter Sentiment Analysis. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 771–776. [CrossRef]
33. Baly, R.; Badaro, G.; Hamdi, A.; Moukalled, R.; Aoun, R.; El-Khoury, G.; Al Sallab, A.; Hajj, H.; Habash, N.; Shaban, K.; et al. OMAM at SemEval-2017 Task 4: Evaluation of English State-of-the-Art Sentiment Analysis Models for Arabic and a New Topic-based Model. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 603–610. [CrossRef]
34. González, J.-À.; Pla, F.; Hurtado, L.-F. ELiRF-UPV at SemEval-2017 Task 4: Sentiment Analysis using Deep Learning. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 723–727. [CrossRef]
35. Al-Ayyoub, M.; Nuseir, A.; Kanaan, G.; Al-Shalabi, R. Hierarchical Classifiers for Multi-Way Sentiment Analysis of Arabic Reviews. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 531–539. [CrossRef]
36. Nuseir, A.; Al-Ayyoub, M.; Al-Kabi, M.; Kanaan, G.; Al-Shalabi, R. Improved hierarchical classifiers for multi-way sentiment analysis. *Int. Arab J. Inf. Technol.* **2017**, *14*, 654–661.
37. Thrun, S. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75.
38. Mehmood, F.; Chen, E.; Akbar, M.A.; Alsanad, A.A. Human Action Recognition of Spatiotemporal Parameters for Skeleton Sequences Using MTLN Feature Learning Framework. *Electronics* **2021**, *10*, 2708. [CrossRef]
39. Li, J.; Zhang, D.; Ma, Y.; Liu, Q. Lane image detection based on convolution neural network multi-task learning. *Electronics* **2021**, *10*, 2356. [CrossRef]
40. Shao, X.; Zhang, X.; Tang, G.; Bao, B. Scene recognition based on recurrent memorized attention network. *Electronics* **2020**, *9*, 2038. [CrossRef]
41. De Bruyne, L.; De Clercq, O.; Hoste, V. Mixing and Matching Emotion Frameworks: Investigating Cross-Framework Transfer Learning For Dutch Emotion Detection. *Electronics* **2021**, *10*, 2643. [CrossRef]
42. Wu, X.; Wang, T.; Wang, S. Cross-modal learning based on semantic correlation and multi-task learning for text-video retrieval. *Electronics* **2020**, *9*, 2125. [CrossRef]
43. Yang, J.; Wei, F.; Bai, Y.; Zuo, M.; Sun, X.; Chen, Y. An Effective multi-task two-stage network with the cross-scale training strategy for multi-scale image super resolution. *Electronics* **2021**, *10*, 2434. [CrossRef]
44. Jin, N.; Wu, J.; Ma, X.; Yan, K.; Mo, Y. Multi-task learning model based on multi-scale cnn and lstm for sentiment classification. *IEEE Access* **2020**, *8*, 77060–77072. [CrossRef]
45. Balikas, G.; Moura, S.; Amini, M.-R. Multitask Learning for Fine-Grained Twitter Sentiment Analysis. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, 7–11 August 2017; pp. 1005–1008. [CrossRef]
46. Lu, G.; Zhao, X.; Yin, J.; Yang, W.; Li, B. Multi-task learning using variational auto-encoder for sentiment classification. *Pattern Recognit. Lett.* **2020**, *132*, 115–122. [CrossRef]
47. Alomari, K.M.; Elsherif, H.M.; Shaalan, K. Arabic Tweets Sentimental Analysis Using Machine Learning. In Proceedings of the 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, Arras, France, 27–30 June 2017; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10350, pp. 602–610. [CrossRef]
48. Abdul-Mageed, M.; Diab, M.; Kübler, S. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Comput. Speech Lang.* **2014**, *28*, 20–37. [CrossRef]
49. Duwairi, R.; El-Orfali, M. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *J. Inf. Sci.* **2014**, *40*, 501–513. [CrossRef]
50. Nabil, M.; Aly, M.; Atiya, A. ASTD: Arabic Sentiment Tweets Dataset. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portuga, 17–21 September 2015; pp. 2515–2519. [CrossRef]
51. Al-Kabi, M.N.; Gigieh, A.H.; Alsmadi, I.M.; Wahsheh, H.A.; Haidar, M.M. Opinion Mining and Analysis for Arabic Language. *Int. J. Adv. Comput. Sci. Appl.* **2014**, *5*, 181–195. [CrossRef]

52. Al-Twairesh, N.; Al-Khalifa, H.; Al-Salman, A.; Al-Ohali, Y. AraSenTi-Tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets. *Procedia Comput. Sci.* **2017**, *117*, 63–72. [CrossRef]
53. Altawaier, M.; Tiun, S. Comparison of machine learning approaches on Arabic twitter sentiment analysis. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2016**, *6*, 1067. [CrossRef]
54. Mataoui, M.; Zelmati, O.; Boumechache, M. A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic. *Res. Comput. Sci.* **2016**, *110*, 55–70. [CrossRef]
55. Medhaffar, S.; Bougares, F.; Estève, Y.; Hadrich-Belguith, L. Sentiment Analysis of Tunisian Dialects: Linguistic Ressources and Experiments. In Proceedings of the Third Arabic Natural Language Processing Workshop (WANLP), Valence, Spain, 3–4 April 2017; pp. 55–61. [CrossRef]
56. Al Sallab, A.; Hajj, H.; Badaro, G.; Baly, R.; El Hajj, W.; Shaban, K.B. Deep Learning Models for Sentiment Analysis in Arabic. In Proceedings of the Second Workshop on Arabic Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 9–17. [CrossRef]
57. Badaro, G.; Baly, R.; Hajj, H.; Habash, N.; El-Hajj, W. A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining. In Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), Doha, Qatar, 25 October 2014; pp. 165–173. [CrossRef]
58. Alayba, A.M.; Palade, V.; England, M.; Iqbal, R. Arabic language sentiment analysis on health services. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), Nancy, France, 3–5 April 2017; pp. 114–118.
59. Al-Azani, S.; El-Alfy, E.-S.M. Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs. In Proceedings of the Neural Information Processing, ICONIP 2017, Guangzhou, China, 14–18 November 2017; Volume 10635, pp. 491–500. [CrossRef]
60. Omara, E.; Mosa, M.; Ismail, N. Deep Convolutional Network for Arabic Sentiment Analysis. In Proceedings of the 2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC), Alexandria, Egypt, 17–19 December 2018; pp. 155–159. [CrossRef]
61. Dahou, A.; Xiong, S.; Zhou, J.; Haddoud, M.H.; Duan, P. Word embeddings and convolutional neural network for Arabic sentiment classification. In Proceedings of the COLING 2016—26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2418–2427.
62. Soumeur, A.; Mokdadi, M.; Guessoum, A.; Daoud, A. Sentiment Analysis of Users on Social Networks: Overcoming the challenge of the Loose Usages of the Algerian Dialect. *Procedia Comput. Sci.* **2018**, *142*, 26–37. [CrossRef]
63. Alhumoud, S.O.; Altuwaijri, M.I.; Albuhairi, T.M.; Alohaideb, W.M. Survey on Arabic Sentiment Analysis in Twitter. *Int. J. Comput. Inf. Eng.* **2015**, *9*, 364–368.
64. Aly, M.; Atiya, A. LABR: A large scale Arabic book reviews dataset. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria, 4–9 August 2013; Volume 2, pp. 494–498. [CrossRef]
65. Al Shboul, B.; Al-Ayyoub, M.; Jararweh, Y. Multi-way sentiment classification of Arabic reviews. In Proceedings of the 2015 6th International Conference on Information and Communication Systems (ICICS), Amman, Jordan, 7–9 April 2015; pp. 206–211. [CrossRef]
66. Nassif, A.B.; Elnagar, A.; Shahin, I.; Henno, S. Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. *Appl. Soft Comput.* **2021**, *98*, 106836. [CrossRef]
67. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489. [CrossRef]
68. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors David. *Nature* **1986**, *323*, 533–536. [CrossRef]
69. Johnson, R.; Zhang, T. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 103–112. [CrossRef]
70. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
71. Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T. Learning Word Vectors for 157 Languages. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation Conference (LREC 2018), Miyazaki, Japan, 7–12 May 2018; pp. 1–5.
72. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
73. Cho, K.; Courville, A.; Bengio, Y. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Trans. Multimed.* **2015**, *17*, 1875–1886. [CrossRef]
74. Raffel, C.; Ellis, D.P.W. Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. *arXiv* **2015**, arXiv:1512.08756.
75. Baniata, L.H.; Park, S.; Park, S.-B. A multitask-based neural machine translation model with part-of-speech tags integration for Arabic dialects. *Appl. Sci.* **2018**, *8*, 2502. [CrossRef]

76. Baniata, L.H.; Park, S.; Park, S.-B. A Neural Machine Translation Model for Arabic Dialects That Utilizes Multitask Learning (MTL). *Comput. Intell. Neurosci.* **2018**, *2018*, 7534712. [CrossRef]
77. Baccianella, S.; Esuli, A.; Sebastiani, F. Evaluation measures for ordinal regression. In Proceedings of the ISDA 2009-9th International Conference on Intelligent Systems Design and Applications, Pisa, Italy, 30 November–2 December 2009; pp. 283–287.
78. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. [CrossRef]
79. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. In Proceedings of the LREC 2020 Workshop Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 9–15.
80. Zeroual, I.; Goldhahn, D.; Eckart, T.; Lakhouaja, A. OSIAN: Open Source International Arabic News Corpus—Preparation and Integration into the CLARIN-infrastructure. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019; pp. 175–182. [CrossRef]
81. Baziotis, C.; Pelekis, N.; Doulkeridis, C. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 747–754. [CrossRef]
82. Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; Mueller, A. Scikit-learn: Machine Learning in Python. *GetMobile Mob. Comput. Commun.* **2015**, *19*, 29–33. [CrossRef]
83. Liu, S.; Johns, E.; Davison, A.J. End-to-end multi-task learning with attention. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1871–1880. [CrossRef]