applied sciences

MDPI

Article

Abusive Content Detection in Arabic Tweets Using Multi-Task Learning and Transformer-Based Models

Bedour Alrashidi [1,2,*], Amani Jamal [1] and Ali Alkhathlan [1]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2] Department of Information and Computer Science, College of Computer Science and Engineering, University of Ha'il, Ha'il 55436, Saudi Arabia
* Correspondence: bhamedalrashidi@stu.kau.edu.sa

Abstract: Different social media platforms have become increasingly popular in the Arab world in recent years. The increasing use of social media, however, has also led to the emergence of a new challenge in the form of abusive content, including hate speech, offensive language, and abusive language. Existing research work focuses on automatic abusive content detection as a binary classification problem. In addition, the existing research work on the automatic detection task surrounding abusive Arabic content fails to tackle the dialect-specific phenomenon. Consequently, this has led to two important issues in the automatic abusive Arabic content detection task. In this study, we used a multi-aspect annotation schema to tackle the automatic abusive content detection problem in Arabic countries, based on the multi-class classification task and the dialectal Arabic (DA)-specific phenomenon. More precisely, the multi-aspect annotation schema includes five attributes: directness, hostility, target, group, and annotator. We specifically developed a framework to automatically detecting abusive content on Twitter using natural language processing (NLP) techniques. The developed framework used different models of machine learning (ML), deep learning (DL), and pretrained Arabic language models (LMs) using the multi-aspect annotation dataset. In addition, to investigate the impact of the other approaches, such as multi-task learning (MTL), we developed four MTL models built on top of a pretrained DA language model (called MARBERT) and trained on the multi-aspect annotation dataset. Our MTL models and pretrained Arabic LMs enhanced the performance compared to the existing DL model mentioned in the literature.

Keywords: abusive content; dialectal Arabic (DA); NLP; DL; multitask learning

Citation: Alrashidi, B.; Jamal, A.; Alkhathlan, A. Abusive Content Detection in Arabic Tweets Using Multi-Task Learning and Transformer-Based Models. *Appl. Sci.* **2023**, *13*, 5825. https://doi.org/10.3390/app13105825

Academic Editor: Douglas O'Shaughnessy

Received: 24 March 2023
Revised: 5 May 2023
Accepted: 6 May 2023
Published: 9 May 2023

1. Introduction

At present, most people around the world are increasingly using social media platforms (such as Twitter, Facebook, YouTube, and others) to express their thoughts and share information. According to recent surveys, the Arab region receives more than 27 million tweets daily [1]. There are nearly 164 million active monthly users on Twitter in the Middle East and North Africa (MENA), demonstrating the platform's popularity. These users generate a huge number of Arabic tweets, many of which are in dialectal Arabic (DA) [2]. However, social networking platforms are increasingly being used to spread violent and abusive content because of their dynamic, democratic, and unregulated nature. The vast reach of social media platforms such as Twitter still needs an automatic method in order to recognize such a toxic phenomenon and radicalization [3]. Thus, this study intends to focus on the automatic abusive Arabic content detection task in Twitter and address the two main existing issues.

The first issue is that the existing research work focuses on automatic abusive content detection as a binary classification problem. The automatic detection of abusive content (such as hate speech, offensive language, and abusive language) is a challenging task [4]

due to the paucity of high-quality annotated datasets. However, binary judgments of hate speech (e.g., hate or not hate) are known to be unreliable [5–7]. Furthermore, recent studies have recommended the use of multiclass detection tasks and a multilabel annotated dataset [5–8] to enhance the detection task beyond the binary classification. The second issue in the automatic abusive Arabic content detection task is that it fails to tackle the dialect-specific phenomenon. This is because the Arabic tweets are written in dialects, and it requires such research and solutions to detect abusive language in tweets.

Thus, this paper aims to tackle the problem using a fine-grained classification task in Arabic. Thus, we searched the available dataset that goes beyond the binary classification task and used it for automatic abusive content detection in Arabic. Generally, we found seven datasets (see Section 3.1) and discovered that the annotation of the available datasets has different schema, e.g., binary, ternary, multiclass, and multi-aspect abusive content. The only available dataset that used a fine-grained and multi-aspect annotation procedure was the muti-aspect hate speech dataset which was constructed by Ousidhoum et al. [9]. The multi-aspect hate speech dataset includes five attributes (directness, hostility, target, group, and annotator). These attributes, labeled as hierarchical annotation procedures, are illustrated in Figure 1, with each attribute found within the indication labels. Each tweet was assigned into one of five aspects, with each one including a multi-class classification task, except the directness aspect which used a binary task. This dataset was designed for multilingual study purposes; however, we selected the Arabic dataset in our study. We believe that this hierarchal annotation process can detect, categorize, and identified abusive content and targeted groups with the fine-grained detection procedure.

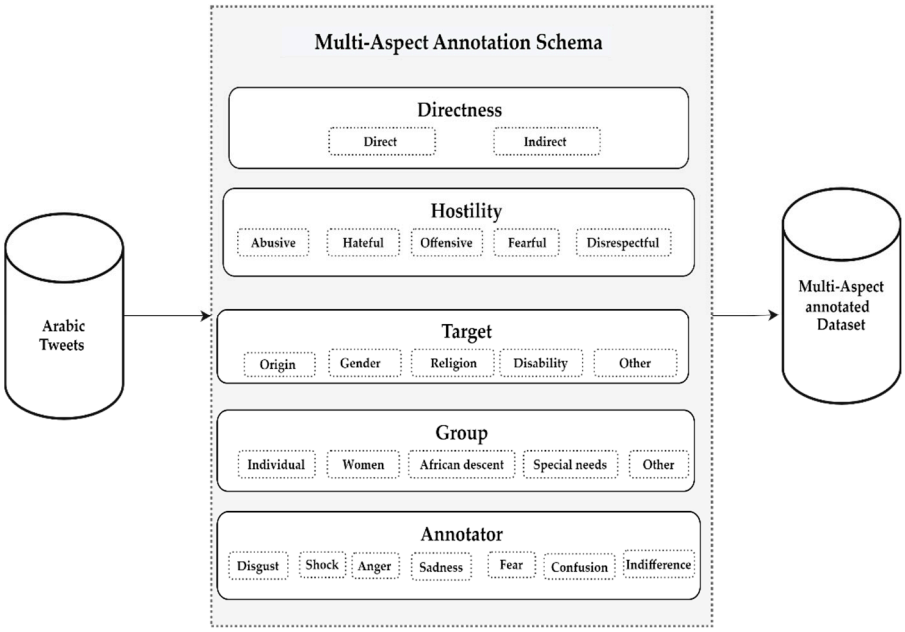


Figure 1. A multi-aspect annotation schema within its indication labels.

In this study, we developed a framework to automatically detect abusive content on Twitter using natural language processing (NLP) techniques. The developed framework used different models, including machine learning (ML), deep learning (DL), and Arabic pretrained language models (LMs), within a multi-aspect annotation dataset. Recently, there has been huge interest in adapting NLP techniques, together with ML and DL approaches, to address the problem of abusive content at large by developing automatic detection models. Due to the advancements in NLP techniques, for many natural language tasks, specifically deep learning models, transformer-based models have recently shown to be the most successful and extensively used method.

Transformer-based models analyze textual information in parallel and employ self-attention mechanisms to compute attention weights that quantify the influence of each word on another in place of the sequence word dependence architecture of recurrent neural network (RNN) models. Numerous pretrained models have been made available since 2018, including bidirectional encoder representations from transformers (BERTs) [10], which greatly aided the development of numerous NLP applications. More precisely, we intend to employ pretrained LMs in DA and MSA, such as MARBERT [11], ArabicBERT [12], CAMeLBERT [13], QARiB [14], and AraBERTv0.2 [15] (see Section 3.2).

In addition, to investigate the impact of the other approaches, such as multi-task learning (MTL), we developed four MTL models built on top of a pretrained DA language model (called MARBERT) and trained on the multi-aspect annotation dataset (see Section 3.3). Our MTL models and pretrained Arabic LMs enhanced the performance compared to the existing DL model in the literature.

Detecting abusive content is still a challenging task, not only in the Arabic language. Recently, many studies have been conducted for hate speech and abusive language detection in many languages. BERT models have recently demonstrated outstanding performance in abusive content identification. According to Zampieri et al. [16] and Ping Liu et al. [17], the BERT model was found to achieve better performance in SemEval-2019 Task6 compared to ML and RNN models. Die et al. [18] built an offensive language detection system, combining MTL with BERT-based models, which achieved a 91.51% F1 score for English SemEval-2020 Task12.

To the best of our knowledge, this study is the first to employ pretrained Arabic LMs in a multi-aspect annotated dataset using different fine-tuning strategies. In addition, no study has investigated the MTL approach with LMs in a multi-aspect annotated dataset.

The main contributions in this study can be summarized as follows:

- We present a framework for automatic abusive content detection in Arabic by using the multi-aspect annotated dataset and applying ML, DL, and pretrained LMs in DA and MSA. Then, we comprehensively evaluate the performance of each approach.
- We propose a MTL model that is built upon pretrained LMs in DA (called MARBERT) to investigate its impact on automatic abusive Arabic content detection.
- We apply four different neural network (NN) architectures to the MTL model and then comprehensively evaluate the performance of each experiment.

The rest of the paper is organized as follows. Section 2 is the literature review which reviews the related work on automatic abusive content detection in Arabic and the available datasets. In Section 3, we present the proposed framework for automatic abusive content detection and the MTL model in detail. Section 4 provides the experimental setup and the performance measurements. The results of the performed experiments are then presented in Section 5. Section 6 presents the conclusion and future prospects.

2. Literature Review

This section presents a review of the previous studies and available datasets for automatic abusive content detection in the Arabic language.

2.1. Automatic Abusive Content Detection in the Arabic Language

Researchers' interest in automatic abusive content detection on social media has recently grown. Whether traditional ML or DL techniques are used, the majority of research in the literature has essentially modeled the issue as a supervised classification task [8,19–22].

Early attempts at tackling this problem involved term frequency weighting which was used to extract n-gram features and then fed to naive Bayes (NB) and support vector machine (SVM) classifiers, according to Mulki et al. [21]. The results indicate the outperformance of NB over SVM by achieving F1 scores of 89.6 for binary classification and 74.4 for ternary classification. Albadi et al. [19] examined the combined effect of the gated recurrent unit (GRU) and SVM with AraVec embeddings [23], and the best results achieved 79% accuracy. Al-Hassan et al. [22] employed the SVM model as a baseline for comparing the

LSTM, CNN + LSTM, GRU, and CNN + GRU deep learning models. They concluded that the performance of detection improved with 72% precision, 75% recall, and a 73% F1 score when the CNN was added as a layer to the LSTM. According to Duwairi et al. [20], CNN models outperformed the CNN-LSTM and achieved an accuracy score of 81%. When a binary classification task was used on the ArHS dataset, CNN-BiLSTM counterparts identified hate speech with greater accuracy.

In recent years, there has been some competition among workshops and conferences that specialize in Arabic language processing. Among those competitions, there was a shared task for offensive language and hate speech detection. For instance, the shared task of offensive language identification was addressed in the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4) [24]. There were two subtasks: Subtask A focused on detecting offensive language and Subtask B focused on detecting hate speech. Most teams used many pre-processing steps, usually via character normalization, punctuation removal, diacritics, the repetition of letters, and tokens that were not Arabic. The teams used deep neural network (DNN) algorithms (such as a CNN and an RNN, including a LSTM, a BiLSTM, and GRUs with and without attention), as well as classic ML techniques (such as the SVM and logistic regression (LR)) and fine-tuning contextual embeddings (such as BERT and AraBERT) [15]. The highest-ranking submissions incorporated a variety of learning techniques (including standard ML) and deep neural networks (DNN) to reach F1 scores of 90.5 (Hassan et al. [25]) and 95.2 (Fatemah Husain [26]) for Subtasks A and B.

For the MTL approach, some studies employed this approach and achieved significantly better performance. For instance, the use of pretrained Arabic BERT was proposed by Djandji et al. [27] for the appropriate classification of various tweets. They used MTL to enhance the AraBERT model, such that it can efficiently learn both tasks together, even in the presence of little per-task labeled data. With macro-F1 scores of 90.0 for Subtask A and 83.41 for Subtask B, their results ranked second in both Subtasks A and B of OSACT4. Another study conducted by Abu Farha and Magdy [28] employed CNN-BiLSTM-based architecture and used MTL to detect hate speech and offensive language. The MTL model achieved a macro-F1 score of 0.76 for the hate speech task and 0.87 for the offensive language task. More recently, the multi-corpus-based learning strategy is explored by Aldjanabi et al. [29] and is built upon MARBERT [11]. For Arabic offensive and hate speech identification tasks, the created MTL model outperformed existing models in the literature on three of four datasets. AlKhamissi and Diab [5] proposed an ensemble of models employing multitask learning and a self-consistency correction method. The proposed model (called AraHS) was found to outperform the baseline models. AraHS is an ensemble of MARBERT [11] model trained with different hyperparameters using MTL. There have been also various studies that have employed the MTL approach to investigate another text classification task such as sentiment analysis (SA). One such study of the five-point sentiment classification issue [30] used a RNN to jointly classify the ternary and five-point sentiment classification tasks. They utilized a BiLSTM accompanied by a hidden layer and additional features, including punctuation symbols, emoticons, and word membership characteristics in sentiment lexicons to ameliorate the sentence representation. The study determined that the integration of related sentiment classification tasks using MTL led to better performance with respect to the five-point sentiment classification task. Another study [31] took an analogous approach, employing the relationship between binary and five-point sentiment classification tasks to jointly train both. The model involved an LSTM encoder with a variational autoencoder (VAE) decoder, where decoder parameters were shared between both tasks. The results showed that this proposed model improved the performance of the five-point sentiment classification task. Another approach presented by Ning et al. [32] included Adaptive Multi-task Learning (AMTL) on the encoding framework, which resulted in higher-quality encoder output and performance overall. Additionally, the current SA on five polarities based on ML algorithms is not performing substantially well, which suggests that the addition of MTL and DL approaches may enhance its performance.

2.2. The Available Datasets for Automatic Abusive Content Detection in the Arabic Language

Table 1 summarizes the available datasets for abusive content detection in the Arabic language. The table lists seven different datasets (DS), each with their own unique characteristics such as number of tweets/posts, language, classification type, and source. The DS1 was created by Albadi et al. [19] and contains 6100 tweets in DA collected from Twitter. The tweets are classified as binary type. DS2 was created by Alshalan and Al-Khalifa [8] and contains 9300 tweets in DA collected from Twitter. The tweets are classified as binary type and are specific to Saudi Arabia dialects. DS3 is called OSACT 4 [24] it was constructed in the 4th Workshop on OSACT and contains 10,000 tweets in DA/MSA collected from Twitter. The tweets are classified as binary type. The DS4 is called OSACT 5 [33], it was constructed in the 5th Workshop on OSACT and contains 12,600 tweets DA/MSA collected from Twitter. The tweets are classified as binary type and multi-class (based on factors such as gender, race, religion, etc.). DS5 was created by Mulki et al. [21] and contains 5800 tweets in Levantine Arabic dialect and the tweets are classified as ternary type. DS6 was created by Haddad et al. [34] and contains 6020 posts and comments from Facebook and YouTube in DA specifically focused on Tunisia dialect. The posts and comments are classified as ternary type. DS7 was created by Ousidhoum et al. [9] and contains 3300 tweets in DA collected from Twitter. The tweets are classified according to multiple aspects including directness, hostility, target, group, and annotator.

Table 1. Summary of the available datasets for automatic abusive content detection in Arabic *.

Dataset	Source	Language	Size	Classification Type	Link
Albadi et al. [19]	Twitter	Arabic (DA)	6.1 K tweets	Binary (HS, not HS)	DS1
Alshalan and Al-Khalifa [8]	Twitter	Arabic (DA) Saudi	9.3 K tweets	Binary (HS, not HS)	DS2
OSACT 4 [24]	Twitter	Arabic (DA)/ (MSA)	10 K tweets	Binary (HS, not HS) (OFF, not OFF)	DS3
OSACT 5 [33]	Twitter	Arabic (DA)/ (MSA)	12.6 K tweets	Binary (HS, not HS) (OFF, not OFF) Multi-HS (gender, race, religion, and others)	DS4
L-HSAB [21]	Twitter	Arabic (DA) Levantine	5.8 K tweets	Ternary (abusive, HS, and normal)	DS5
T-HSAB [34]	Facebook and YouTube	Arabic (DA) Tunisian	6.02 K posts and comments	Ternary (abusive, HS, and normal)	DS6
Ousidhoum et al. [9]	Twitter	Arabic (DA)	3.3 K tweets	Muti-Aspects (directness, hostility, target, group, and annotator)	DS7

* List of acronyms used in this summary: HS (hate speech), OFF (offensive language), MSA (modern standard Arabic), DA (dialectal Arabic), L-HSAB (Levantine hate speech and abusive), T-HSAB (Tunisian hate speech and abusive dataset), OSACT (Open-Source Arabic Corpora and Processing Tool), DS (dataset).

Noticeably, the majority of relevant works on abusive Arabic content detection used binary and ternary classification task, except the recently released OSACT 5 [33] dataset and the selected dataset used in this study [9]. Furthermore, the main dataset collected from Twitter and annotated using different strategies was the dataset with its annotation

uploaded in GitHub. In addition, the available datasets had some limitations, such as the number of instances in small size, most of which did not exceed 13,000 instances.

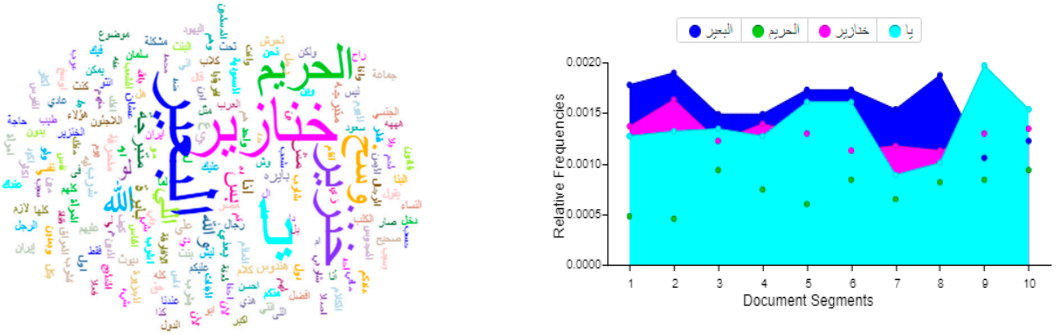
3. Materials and Methods

3.1. Dataset Description

The dataset used in this study was constructed by Ousidhoum et al. [9]. Essentially, they constructed three datasets for the multilingual study in English, French, and Arabic. In this study, we selected the Arabic dataset only containing 3353 Arabic tweets. They used Amazon Mechanical Turk (AMT) [35] to label 13,000 tweets into muti-aspect annotations schema. They designed annotation guidelines for the annotator to label each tweet and used the Krippendorff [36] average score, with an average score of 0.202 in the Arabic dataset. The distribution for each label and the number of instances are listed in Table 2. These include: (a) the identification of whether direct or indirect text is used; (b) the use of offensive, disrespectful, abusive, normal, fearful, or ignorant text; (c) the use of text that discriminates against an individual or a particular category of people; (d) a group's name; (e) the feelings of annotators on the annotated content of a range of negative-to-neutral sentiments. In this study, we used online Voyant tools [37] to explore the word cloud and the term frequencies of the selected dataset, as shown in Figure 2. The most frequent terms mainly referred to the names of animals, particularly in tweets with abusive content as well as the names of gender. For instance, "البعير/Camel ", "خنازير/Pigs", and "الحريم/women". Moreover, the frequent term "يا/yA/O" indicates vocative particle forms and is mainly used to direct speech to a specific person or group, and usually followed by hate or abusive words.

Table 2. The distribution for each label and the number of instances.

Attribute	Label	No. Instances
Directness	Direct	1684
	Indirect	754
Hostility	Abusive	610
	Hateful	755
	Offensive	1151
	Disrespectful	615
	Fearful	41
	Normal	1197
Target	Origin	877
	Gender	548
	Religion	145
	Disability	1
	Other	1782
Group	Individual	915
	Other	1470
	Women	722
	Special needs	2
	African descent	51
Annotator	Disgust	778
	Shock	917
	Anger	356
	Sadness	388
	Fear	35
	Confusion	115
	Indifference	1825



(**a**) (**b**)

Figure 2. The most frequent terms in Arabic tweets from the selected dataset: (**a**) a description of the word cloud; (**b**) the relative frequencies with document segments.

3.2. The Proposed Framework for Automatic Abusive Content Detection in the Arabic Language

The proposed framework in Figure 3 for automatic abusive content detection in Arabic is designed by adopting the text classification pipeline [38]. It is represented as three main steps. The first involves the data collection and annotation. The second involves the feature selection and model training. The third step involves the model validation. In addition, the bottom part of the list refers to the implementation environments (see Section 4) that were used for data retrieval, processing, model development, and validation training.

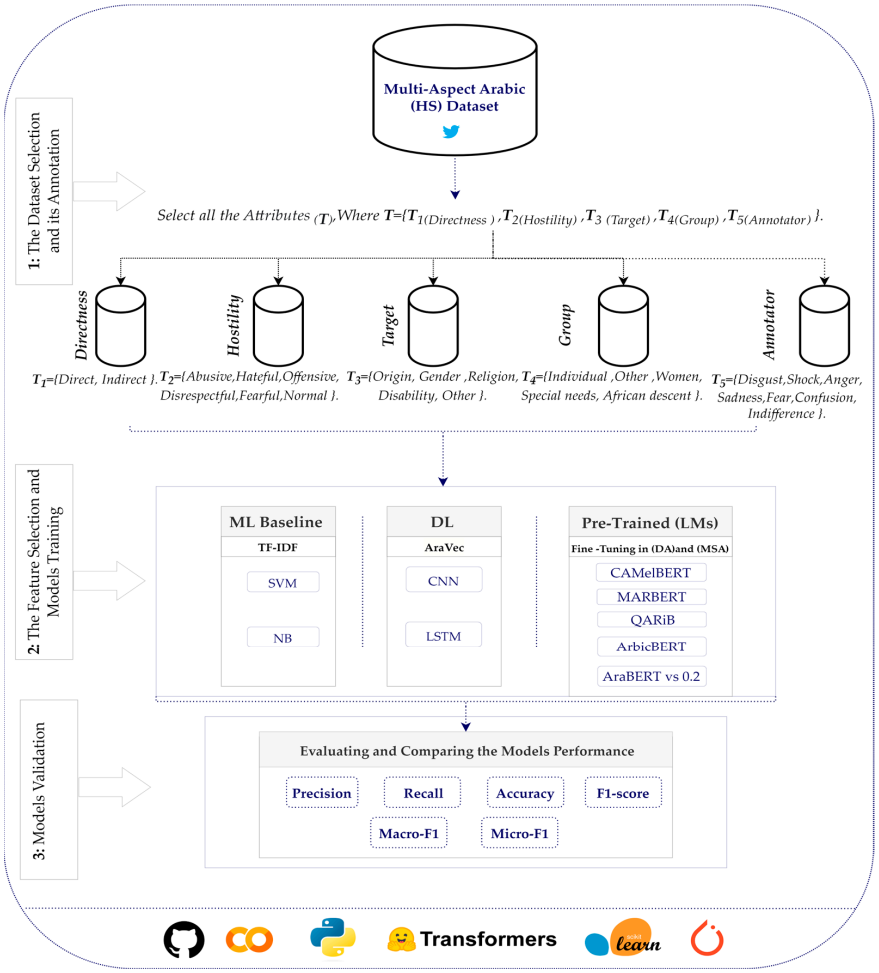


Figure 3. The framework for automatic abusive content detection in the Arabic language.

In the first step, we selected the Arabic dataset that is stored in the GitHub repository [39]. More specifically, we selected all the attributes within the indication labels, where each attribute represented a multi-class classification task, except the directness attribute which used the binary classification task. The number of instances for each attribute and its indication labels are listed in Table 2.

The second step involved the feature selection and model training level, whereby three main approaches were used (**ML**, **DL**, and **pretrained Arabic LMs**). Each approach with selected features and models is presented as follows.

For the **ML** approach, two models were used (SVM and NB) as they were found to be efficient in previous studies [21,40]. For the feature representation, we used term frequency–inverse document frequency (TF-IDF) [41] which performed well when combined with classification models [42,43].

For the **DL** approach, two models were used (the LSTM and the CNN) for hate speech detection studies in the Arabic language [8,22] as they showed the best performance compared to the other models. The LSTM is a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem that was encountered with the standard RNN. CNNs are known for their good performance in image analysis, but recent advancements show that CNNs perform well on text data as they extract important features from input feature vectors that help with any downstream task. For the representation of features, we used word embeddings, which are dense vector representations of words. More specifically, we utilized the Arabic word embedding model called AraVec [23].

For the **pretrained Arabic LMs**, we used the fine-tuning strategy for the text classification task [44]. The pretrained language models in Arabic were mainly based on a stacked BERT [10] model. BERT has two models: (1) the $\text{BERT}_{\text{BASE}}$ (12 encoders with 12 bidirectional self-attention heads) and (2) the $\text{BERT}_{\text{LARGE}}$ (24 encoders with 24 bidirectional self-attention heads). The original BERT was trained on 3.3 B words extracted from English Wikipedia and the book Corpus [45]. The BERT base configuration has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence lengths, and a total of ∼110 M parameters. Several researchers have released and benchmarked pretrained Arabic LMs; some of the main differences among these models are the genre and the amount of Arabic data that they are trained on. For example, AraBERT [15] was trained only on MSA, while AraBERTv0.2 [15] was trained on DA and MSA. MARBERT [11], ArabicBERT [12], and QARiB [14] used DA during training. CAMeLBERT-mix [13], used a combination of all types of Arabic text for training. We summarized the recent pretrained Arabic LMs in Table 3, and we focused on the models that trained on DA in order to fine-tune these models in our selected dataset. In addition, the low number of samples in the selected dataset led to a low resource scenario problem. Thus, we took this issue into account and searched for the recent approaches for the low resource scenario [46]. Therefore, we relied on the data augmentation method using the NLPaug library [47]. NLPaug is a library used for textual augmentation in machine learning experiments. It uses word embedding techniques and various augmenter strategies, including insertion and substitutions, to augment the data on a character level, a word level, and a sentence level. In our experiments, to augment the sample size of tweets, we performed a word-level augmentation using ContextualWordEmbsAug from nlpaug.augmenter.word.

Table 3. Summary of pretrained Arabic LMs.

Pretrained Arabic LMs	Size	# Word	# Token	Language Type
MARBERT [11]	128 GB	100 K	15.6 B	DA
ArabicBERT [12]	95 GB	32 K	8.2 B	DA
CAMeLBERT [13]	16 7 B	30 K	17.3 B	DA/CA/MSA
QARiB [14]	127 GB	64 K	14.0 B	DA
AraBERTv0.2 [15]	77 GB	60 K	8.6 B	DA/MSA

3.3. The Proposed MTL Architecture

Rather than learning each aspect individually, each with its own parameters, we intended to allow the model to share the hidden layer parameters among aspects in order to positively transfer knowledge and leverage information contained in the related aspects. Thus, we treated each aspect as a task in our dataset. MTL is employed in many ML applications to enhance a model's performance and generalizability [48–50]. MTL is a machine learning technique that involves simultaneously training a model on multiple related tasks in order to improve the performance of all tasks. By training on multiple tasks simultaneously, the model can better learn the shared information between the different tasks, leading to more efficient training and generalization. The advantage of MTL is that multiple tasks can be learned simultaneously, which allows for the transfer of information and the generalization of skills from one task to another. This kind of learning can result in a greater understanding of how tasks relate to each other, as well as the improved performance on each task. Since we are dealing with the hierarchical annotation dataset, we considered MTL to be a very useful strategy. Therefore, we adopted the MTL approach to improve the performance of automatic fine-grained abusive content detection in Arabic using a multi-aspect annotated dataset.

The proposed MTL architecture is shown in Figure 4, where the bottom part is the MARBERT language model, which will be shared among tasks. We used MARBERT LM as it showed the best performance compared to other LMs. The upper parts were five modules which were used for each sub-task. We trained four MTL models in the five modules which were used for each sub-task. The models included MTL with MARBERT, MTL with LSTM + MARBERT, MTL with LSTM + CNN + MARBERT, and MTL with BiLSTM + CNN + MARBERT. Unlike previous work [5,29] our paper focuses on improving fine-grained abusive Arabic content detection using MTL with a multi-aspect annotation dataset.

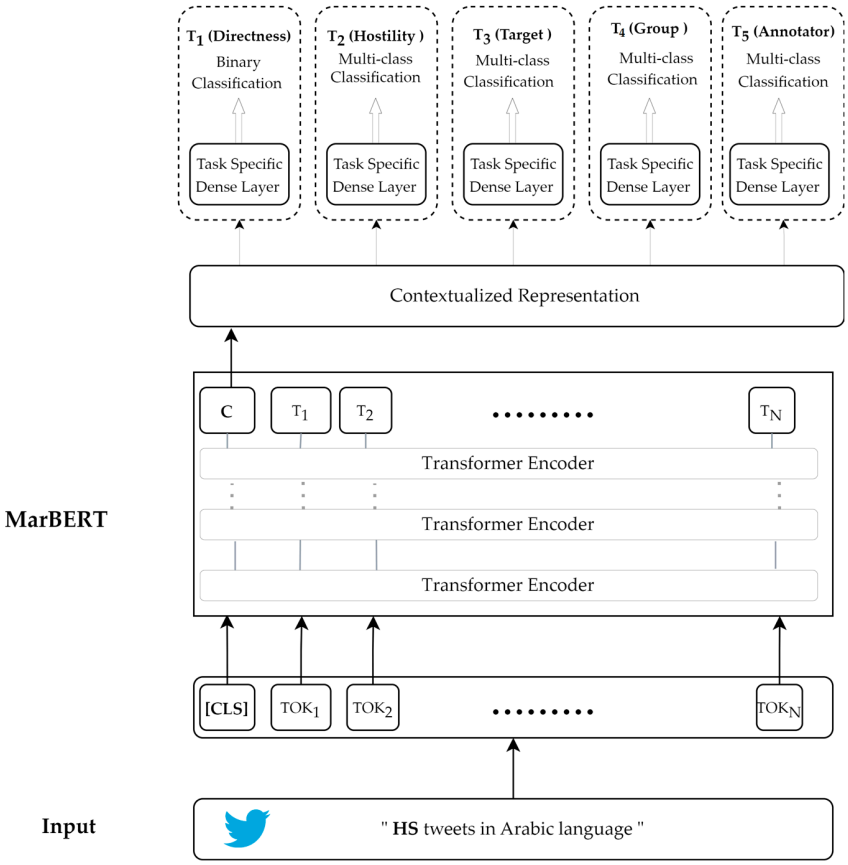


Figure 4. The architecture of the proposed MTL model.

The first model is based on MTL with MARBERT in its base feature extractor without any headers. The final prediction head with only one dense layer was used to map the feature vector into final classification targets. This design inspired the effect of the MARBERT-generalized feature extraction ability and was used in a previous study, but with a cross-corpora MTL model [29].

The second model is based on MTL with MARBERT + LSTM heads; this architecture was used with BERT-based MTL for offensive language detection [18]. The feature extractor model architecture consists of RNNs with LSTM cells followed by a dropout layer to avoid over-fitting and a final classification dense layer.

The third and fourth models are quite similar, apart from the fact that the LSTM was used in the third model and the BiLSTM was used in the fourth model. The third model is based on MTL with MARBERT + LSTM + CNN; this model's architecture is different from the above designs because the CNN network is applied in the architecture consisting of feature extraction, followed by the RNN head with the LSTM, the dropout layer to avoid model overfitting, the 1D convolution layer, batch normalization, ReLU activation, the dropout layer, followed by the final classification dense layer.

Our experiment tested the four different MTL architectures to enhance different types of prediction heads, but the MARBERT-based feature extractor is common for each architecture. The input HS tweet is first fed into the shared MARBERT, and then each sub-task module takes the contextualized embeddings generated by MARBERT and produces a probability distribution for its own target labels. The overall loss L is calculated in Equation (1), where I= {T_1, T_2, T_3, T_4, T_5} and w_i is the loss weight for each task-specific cross-entropy loss L_i.

$$L = \sum_i^I w_i L_i \quad (1)$$

4. Experimental Setup

4.1. Experiment Settings

For the hyperparameter setting, we used different parameter settings with each model. For the ML baseline, the SVM and NB were used with default parameters, as shown in the Scikit-learn library. For the DL models with the LSTM hidden state, the dimension was 64, the dropout was 0.3, the direction was true, the number of units in the fully connected layer was 32, the dropout rate was 0.4, the learning rate was 0.001, the number of epochs was 20, the batch size was 32, and the optimizer was Adam [51]. The hyperparameters used for the CNN were filter sizes(2, 3, 4, 5), the number of filters (32), the number of units in the fully connected layer (30), the dropout rate (0.5), the learning rate (0.001), the number of epochs (20), the batch size (32), the optimizer (Adam), and the loss function (cross entropy loss). The hyperparameter settings used the same values for the pretrained Arabic LMs provided by MARBERT [11], ArabicBERT [12], CAMeLBERT [13], QARiB [14], and AraBERTv0.2 [15]. For instance, we used the same fine-tuning hyperparameter used in MARBERT, where the input sequence length was 256, the batch size was 32, the learning rate was 2×10^{-6}, the number of epochs was 25, and the optimizer was Adam. For the MTL experiments, we used the hyperparameter settings provided by [18].

All the comparison algorithms used the split ratio train:dev:test = 8:1:1, where this split ratio was utilized in the same dataset [9], and the results provided are based on the test set. For the environment setup, all the experiments were run using Google Colab [52] with a GPU accelerator. The models were built using many libraries and tools, all of which were open-source and written in Python. The used tools were Scikit-learn and PyTorch, and all pretrained Arabic LMs were available in the HuggingFace Transformers library [53].

4.2. Performance Measures

The performance metrics used in this study were precision (2), recall (3), accuracy (4), and F1 score (5), where the proportion of correctly categorized positive samples was referred to as true positive (TP). True negative samples are those that can be classified as negative with accuracy (TN). False positive samples are those that are overly positive based

on the sample count (FP). False negatives are those that can be incorrectly categorized as negative (FN). In addition, in order to evaluate the multi-class classification models [54], two performance measures (macro-F1 and micro-F1) were used for the abusive content detection task in the literature.

The micro-average precision and recall score were calculated from the individual classes (TPs, TNs, FPs, and FNs) of the model. The micro-average essentially computes the overall accuracy [54]. The macro-average precision and recall score were calculated as the arithmetic mean of the individual classes' precision and recall scores.

We list here the performance metric equations:

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{2}$$

$$\text{Recall} = \frac{T_P}{T_P + T_N} \tag{3}$$

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{4}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

5. Results and Discussion

In this paper, many experiments were conducted to study the effect of using multiple approaches for automatic abusive content detection in Arabic. The Arabic language has a rich morphological structure and many dialects; thus, it is a challenging task to detect abusive content in Arabic, especially when dealing with daily and informal tweets. Accordingly, we used the recent proposed pretrained LMs in DA (see Table 3) in our proposed framework. On the other hand, the binary classification task was not enough to capture the targeted groups and other related aspects of abusive content in tweets; thus, subsequently tackling this problem using multiclass classification task is useful in order to detect many aspects of abusive content. Therefore, we used the multi-aspect annotated dataset with 3.3 K tweets in Arabic and all the compared algorithms used the same split ratio train:dev:test = 8:1:1, as this split was used in the previous work. Additionally, due to the low resource scenario, we relied on a data augmentation method using the NLPaug library. The results for the proposed framework are presented in Section 5.1 and the results for the MTL models are presented in Section 5.2, where the best results are shown in bold. For the performance evaluation, we selected the macro-F1 metric to compare the achieved results. Macro-F1 has been adopted as the official evaluation measure for the imbalance in class distributions [55]; it is computed as a simple arithmetic mean of per-class F1 scores. Furthermore, the computational complexity measures the amount of computing resources (time and space) consumed by a particular algorithm when it runs. A set of experiments was undertaken using different parameters each time. The specific details of the hyperparameters in each experiment were reported previously in Section 4.1. Simultaneously, we performed the experiments using Google Colab Pro+ with high RAM requirements and a GPU accelerator; however, since we were dealing with a low number of samples, each experiment took 7 to 8 min to run.

5.1. Results of the Proposed Framework

Regardless of all the above challenges for automatic abusive content detection in Arabic, the proposed framework shows enhanced performance for the multi-aspect annotated dataset compared to the previous study [9]. The achieved results for ML, DL, and pretrained LMs are presented in Tables 4–6, respectively.

Table 4. ML baseline performance.

Attribute	ML Baseline Models	Performance Metrics			
		Macro-Avg			Micro-F1
		Prec.	Recall	F1	Acc.
Directness	SVM	0.58	0.57	**0.56**	0.59
	LR *	-	-	0.53	0.56
	NB	0.53	0.53	0.51	0.51
Hostility	SVM	0.34	0.25	0.25	0.43
	LR *	-	-	0.25	0.48
	NB	0.26	0.36	**0.30**	0.39
Target	SVM	0.43	0.40	0.41	0.59
	LR *	-	-	**0.47**	0.53
	NB	0.38	0.39	0.38	0.46
Group	SVM	0.70	0.72	**0.70**	0.68
	LR *	-	-	0.40	0.62
	NB	0.50	0.43	0.45	0.52
Annotator	SVM	0.12	0.15	0.13	0.39
	LR *	-	-	0.14	0.46
	NB	0.20	0.22	**0.21**	0.31

* Result obtained by Ousidhoum et al. [9].

Table 5. DL performance results.

Attribute	DL Models	Performance Metrics			
		Macro-Avg			Micro-F1
		Prec.	Recall	F1	Acc.
Directness	CNN	0.62	0.56	0.53	0.61
	LSTM	0.62	0.57	0.55	0.62
	BiLSTM *	-	-	**0.84**	0.72
Hostility	CNN	0.22	0.22	0.21	0.31
	LSTM	0.21	0.24	0.22	0.35
	BiLSTM *	-	-	**0.31**	0.47
Target	CNN	0.38	0.34	0.34	0.55
	LSTM	0.39	0.40	0.40	0.55
	BiLSTM *	-	-	**0.63**	0.50
Group	CNN	0.69	0.49	**0.52**	0.60
	LSTM	0.45	0.46	0.45	0.61
	BiLSTM *	-	-	0.04	0.58
Annotator	CNN	0.14	0.16	**0.13**	0.42
	LSTM	0.10	0.15	0.09	0.41
	BiLSTM *	-	-	0.12	0.48

* Result obtained by Ousidhoum et al. [9].

Table 6. Pretrained Arabic LM performance results.

Attribute	Pretrained Arabic LMs	Performance Metrics							
		Before NLPaug				After NLPaug			
		Macro-Avg			Micro F1	Macro-Avg			Micro-F1
		Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
Directness	MARBERT	0.63	0.60	0.60	0.64	0.68	0.65	**0.65**	0.67
	CAMeLBERT	0.64	0.63	0.63	0.65	0.64	0.62	0.62	0.64
	QARiB	0.70	0.61	0.58	0.66	0.65	0.64	0.64	0.65
	ArabicBERT	0.64	0.59	0.57	0.64	0.63	0.61	0.60	0.63
	AraBERTv0.2	0.62	0.60	0.59	0.63	0.65	0.64	0.64	0.66
Hostility	MARBERT	0.28	0.29	0.26	0.41	0.45	0.46	**0.44**	0.46
	CAMeLBERT	0.30	0.31	0.26	0.43	0.44	0.44	0.43	0.45
	QARiB	0.30	0.29	0.25	0.41	0.42	0.42	0.40	0.43
	ArabicBERT	0.23	0.28	0.25	0.40	0.43	0.43	0.40	0.45
	AraBERTv0.2	0.24	0.27	0.24	0.38	0.42	0.43	0.41	0.44
Target	MARBERT	0.57	0.61	0.57	0.63	0.79	0.80	0.79	0.79
	CAMeLBERT	0.58	0.61	0.58	0.59	0.81	0.82	**0.82**	0.81
	QARiB	0.58	0.65	0.60	0.63	0.78	0.79	0.78	0.78
	ArabicBERT	0.58	0.51	0.53	0.58	0.78	0.78	0.77	0.77
	AraBERTv0.2	0.60	0.61	0.59	0.61	0.80	0.81	0.80	0.80
Group	MARBERT	0.57	0.60	0.58	0.77	0.89	0.89	**0.89**	0.88
	CAMeLBERT	0.76	0.78	0.76	0.75	0.84	0.84	0.84	0.84
	QARiB	0.75	0.78	0.75	0.77	0.81	0.81	0.80	0.81
	ArabicBERT	0.71	0.63	0.66	0.72	0.80	0.80	0.79	0.80
	AraBERTv0.2	0.72	0.70	0.71	0.70	0.81	0.81	0.80	0.83
Annotator	MARBERT	0.17	0.17	0.16	0.36	0.55	0.54	0.54	0.55
	CAMeLBERT	0.19	0.17	0.16	0.35	0.56	0.57	**0.56**	0.57
	QARiB	0.14	0.13	0.13	0.26	0.15	0.15	0.28	0.55
	ArabicBERT	0.14	0.16	0.14	0.26	0.46	0.50	0.46	0.51
	AraBERTv0.2	0.18	0.17	0.16	0.28	0.38	0.41	0.38	0.43

As outlined in Table 4, the ML model's performance using TF-IDF improved the macro-F1 results in four out of five attributes compared to the obtained result [9]. Table 5 shows that the DL models do not show better performance in three out of five attributes; this is due to the fact that DL models require a higher amount of data, and the imbalanced dataset also affects the F1 macro scores. Therefore, this led to a limitation in our study which can be addressed in future studies by using different methods to enhance the performance. Nevertheless, pretrained Arabic LMs provide better performance compared to the other models, especially when we increase the sample size of tweets by using NLPaug library. The effect of before and after using NLPaug with pretrained LMs is presented in Table 6. The obtained results demonstrated that the use of pretrained Arabic LMs with NLPaug achieved the highest performance among all the previous obtained results. This is because that the selected models MARBERT, CAMeLBERT, QARiB, ArabicBERT, and AraBERTv0.2 were pretrained on DA datasets. In addition, we found that MARBERT achieved better performance compared to the others in three out of five attributes (the best results are shown in bold).

5.2. Results of the MTL Models

To boost the performance of automatic fine-grained abusive content detection in Arabic, the MTL approach with the MARBERT language model was applied on a multi-aspect annotation dataset. A benefit of MTL is that multiple tasks can be learned simultaneously, which allows for the transfer of information and the generalization of skills from one task to another. We performed four experiments (MTL with MARBERT, MTL with LSTM + MARBERT, MTL with LSTM + CNN + MARBERT, and MTL with BiLSTM + CNN + MARBERT). The results achieved for the four experiments are presented in Table 7, and we compare it with single task single language (STSL), which were used in previous work [9] by utilizing a BiLSTM model. Notably, using MTL with MARBERT in different architectures shows better results in four out of the five attributes compared to the previous study [9]. The macro-F1 performance results in the four attributes were 0.71 for the target attribute, 0.91 for the group attribute, 0.34 for the hostility attribute, and 0.23 for the annotator attribute. The precision and recall scores for each experiment are shown in Figure 5. However, the weak performance in the hostility and annotator attributes was due to the highly imbalanced and low number of samples in each label of those attributes in the dataset. Therefore, some of the limitations and directions for future are addressed in this paper. Overall, it is important to highlight that dealing with a multi-aspect annotation dataset is a very challenging task, especially given some of limitations that have been discussed in this study. The study's findings have implications for social media platforms to adopt more effective methods for detecting and addressing abusive content in Arabic especially by using pretrained Arabic LMs. Moreover, we believe that the abusive content detection in tweets is more reliable using the multi-aspect annotation dataset.

Table 7. MTL performance in comparison to previous work.

Models	Attribute									
	Directness		Hostility		Target		Group		Annotator	
	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)
STSL (BiLSTM) *	0.72	**0.84**	0.47	0.31	0.50	0.63	0.58	0.04	0.48	0.12
MARBERT	0.64	0.60	0.41	0.26	0.63	0.57	0.77	0.58	0.36	0.16
MTL (MARBERT)	0.73	0.72	0.47	**0.34**	0.67	0.51	0.88	0.80	0.45	**0.23**
MTL (MARBERT + LSTM)	0.76	0.76	0.46	**0.34**	0.75	**0.71**	0.88	0.87	0.39	0.22
MTL (MARBERT + LSTM + CNN)	0.70	0.70	0.44	0.29	0.68	0.51	0.87	0.66	0.47	019
MTL (MARBERT + BiLSTM + CNN)	0.75	0.75	0.45	0.32	0.60	0.59	0.89	**0.91**	0.41	**0.23**

* Result obtained by Ousidhoum et al. [9].

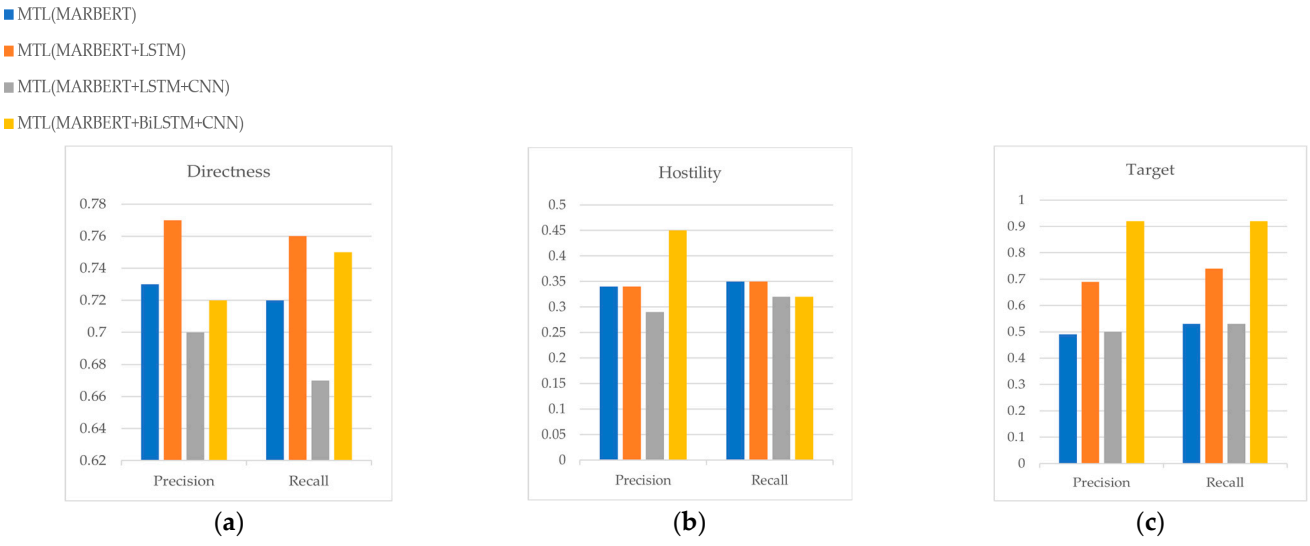
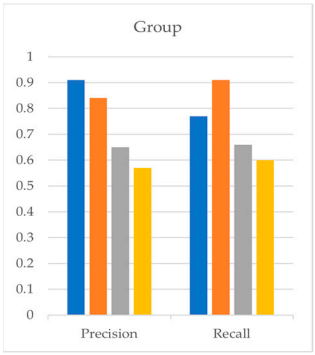
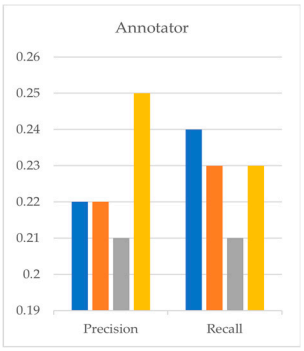


Figure 5. *Cont.*



(**d**)



(**e**)

Figure 5. The precision and recall results of all MTL architectures. The performance of the models based on: (**a**) the directness attribute; (**b**) the hostility attribute; (**c**) the target attribute; (**d**) the group attribute; and (**e**) the annotator attribute.

6. Conclusions

In this work, we present a framework for automatic abusive content detection in Arabic using the multi-aspect annotated dataset. Three main approaches were applied (ML, DL, and pretrained Arabic LMs) and we then comprehensively evaluated the performance for each approach and compared the obtained results with the previous study. TF-IDF feature representation was used with ML baseline models (SVM and NB). For the DL approach, a word embedding model called AraVec was used with two DL models (the CNN and the LSTM). A fine-tuning strategy was applied in five pretrained Arabic LMs in DA, MSA, and mixed language types. The pretrained Arabic LMs used in this study were MARBERT, ArabicBERT, CAMeLBERT, QARiB, and AraBERTv0.2. Additionally, due to the low resource scenario of the selected dataset, a data augmentation technique was applied using the NLPaug library with pretrained Arabic LMs. The MARBERT and CAMeLBERT show better performance among the other pretrained Arabic LMs. The obtained results outperformed four out of five attributes achieving macro-F1 results of 0.84 for the group attribute, 0.82 for the target attribute, 0.56 for the annotator attribute, and 0.44 for the hostility attribute.

Moreover, we investigated the impact of using MTL with a multi-aspect annotated dataset. Therefore, we developed a MTL based on MARBERT, and four different architectures were examined. We then compared the archived performance with the previous study. The results demonstrate that MTL enhanced the performance compared to the existing DL model proposed in the literature. The obtained result outperformed four out of five attributes by achieving macro-F1 scores of 0.71 for the target attribute, 0.91 for the group attribute, 0.34 for the hostility attribute, and 0.23 for the annotator attribute.

Finally, our experiments attempted to address the two issues in automatic abusive content detection in Arabic using multiple approaches. However, further investigation is necessary to address one issue in the used dataset: the highly imbalanced dataset. Therefore, applying some algorithms for an imbalanced dataset may enhance the performance, such as the synthetic minority oversampling technique (SMOTE). Nevertheless, increasing the size of the dataset may enhance the detection performance, especially with DL approaches and pretrained LMs, due to the fact that those approaches are data-hungry and require much more data. This study was limited to focusing on the Arabic language only, but it could also use the multi-aspect dataset with multilingual information by applying similar methods or, for example, the BERT multilingual base model.

Author Contributions: Conceptualization, B.A., A.J. and A.A.; methodology, B.A., A.J. and A.A.; software, B.A.; validation, B.A., A.J. and A.A.; investigation B.A., A.J. and A.A.; writing—original draft, B.A.; writing—review and editing A.J. and A.A.; supervision, A.J. and A.A.; funding acquisition, none. All authors have read and approved the submission of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available in the GitHub repository https://github.com/HKUST-KnowComp/MLMA_hate_speech accessed on 12 November 2021.

Acknowledgments: The authors express their gratitude towards the anonymous reviewers, whose comments greatly contributed to improving this paper.

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

Abbreviations

DA	Dialectal Arabic
NLP	Natural language processing
ML	Machine learning
DL	Deep learning
LMs	Language models
MTL	Muti-task learning
BERT	Bidirectional encoder representations from transformers
MSA	Modern standard Arabic
CA	Classical Arabic
NN	Neural network
SVM	Support vector machine
LR	Logistic regression
NB	Naive Bayes
DNN	Deep neural network
LSTM	Long short-term memory
BiLSTM	Bidirectional LSTM
GRU	Gated recurrent units
CNN	Convolutional neural network
OSACT	Open-source Arabic corpora and processing tools

References

1. Salem, F. *Social Media and the Internet of Things (The Arab Social Media Report 2017)*; MBR School of Government: Dubai, United Arab Emirates, 2017.
2. Abdelali, A.; Mubarak, H.; Samih, Y.; Hassan, S.; Darwish, K. Arabic Dialect Identification in the Wild. *arXiv* **2020**, arXiv:2005.06557.
3. Fraiwan, M. Identification of Markers and Artificial Intelligence-Based Classification of Radical Twitter Data. *Appl. Comput. Informatics* **2022**. [CrossRef]
4. MacAvaney, S.; Yao, H.R.; Yang, E.; Russell, K.; Goharian, N.; Frieder, O. Hate Speech Detection: Challenges and Solutions. *PLoS ONE* **2019**, *14*, e0221152. [CrossRef]
5. AlKhamissi, B.; Diab, M. Meta AI at Arabic Hate Speech 2022: MultiTask Learning with Self-Correction for Hate Speech Classification. *arXiv* **2022**, arXiv:2205.07960.
6. Sanguinetti, M.; Poletto, F.; Bosco, C.; Patti, V.; Stranisci, M. An Italian Twitter Corpus of Hate Speech against Immigrants. In Proceedings of the LREC 2018—11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018; pp. 2798–2805.
7. Assimakopoulos, S.; Muskat, R.V.; Van Der Plas, L.; Gatt, A. Annotating for Hate Speech: The MaNeCo Corpus and Some Input from Critical Discourse Analysis. *arXiv* **2020**, arXiv:2008.06222.
8. Alshalan, R.; Al-Khalifa, H. A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. *Appl. Sci.* **2020**, *10*, 8614. [CrossRef]

9. Ousidhoum, N.; Lin, Z.; Zhang, H.; Song, Y.; Yeung, D.Y. Multilingual and Multi-Aspect Hate Speech Analysis. In Proceedings of the EMNLP-IJCNLP 2019—2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 4675–4684.
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 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.
11. Abdul-Mageed, M.; Elmadany, A.R.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 5–6 August 2021; Association for Computational Linguistics: Cedarville, OH, USA, 2021; pp. 7088–7105. [CrossRef]
12. Safaya, A.; Abdullatif, M.; Yuret, D. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. In Proceedings of the 14th International Workshops on Semantic Evaluation, SemEval 2020—Co-Located 28th International Conference on Computational Linguistics, Virtual, 8–12 December 2020; pp. 2054–2059.
13. Inoue, G.; Alhafni, B.; Baimukan, N.; Bouamor, H.; Habash, N. The Interplay of Variant, Size, and Task Type in Arabic Pre-Trained Language Models. In Proceedings of the WANLP 2021—6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop, Kyiv, Ukraine, 19 April 2021; pp. 92–104.
14. Abdelali, A.; Hassan, S.; Mubarak, H.; Darwish, K.; Samih, Y. Pre-Training BERT on Arabic Tweets: Practical Considerations. *arXiv* **2021**, arXiv:2102.10684.
15. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-Based Model for Arabic Language Understanding. *arXiv* **2020**. [CrossRef]
16. Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; Kumar, R. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). *arXiv* **2019**, arXiv:1903.08983.
17. Liu, P.; Li, W.; Zou, L. NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection Using Bidirectional Transformers. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 87–91.
18. Dai, W.; Yu, T.; Liu, Z.; Fung, P. Kungfupanda at SemEval-2020 Task 12: BERT-Based Multi-Task Learning for Offensive Language Detection. In Proceedings of the 14th International Workshops on Semantic Evaluation, SemEval 2020—Co-Located 28th International Conference on Computational Linguistics, Virtual, 8–13 December 2020; pp. 2060–2066.
19. Albadi, N.; Kurdi, M.; Mishra, S. Are They Our Brothers? Analysis and Detection of Religious Hate Speech in the Arabic Twittersphere. In Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining—ASONAM 2018, Barcelona, Spain, 28–31 August 2018; pp. 69–76.
20. Duwairi, R.; Hayajneh, A.; Quwaider, M. A Deep Learning Framework for Automatic Detection of Hate Speech Embedded in Arabic Tweets. *Arab. J. Sci. Eng.* **2021**, *46*, 4001–4014. [CrossRef]
21. Mulki, H.; Haddad, H.; Bechikh Ali, C.; Alshabani, H. *L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 111–118. Available online: https://aclanthology.org/W19-3512 (accessed on 28 September 2022).
22. Al-Hassan, A.; Al-Dossari, H. Detection of Hate Speech in Arabic Tweets Using Deep Learning. Multimedia Systems; Springer: Berlin/Heidelberg, Germany, 2021.
23. Soliman, A.B.; Eissa, K.; El-Beltagy, S.R. AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP. *Procedia Comput. Sci.* **2017**, *117*, 256–265. [CrossRef]
24. Mubarak, H.; Darwish, K.; Magdy, W.; Al-Khalifa, H. *Overview of OSACT4 Arabic Offensive Language Detection Shared Task*; European Language Resource Association: Marseille, France, 2020; pp. 11–16. Available online: https://aclanthology.org/2020.osact-1.7 (accessed on 25 September 2021).
25. Hassan, S.; Samih, Y.; Mubarak, H.; Abdelali, A.; Rashed, A.; Chowdhury, S.A. ALT Submission for OSACT Shared Task on Offensive Language Detection. In Proceedings of the OSACT 2020, Marseille, France, 12 May 2020.
26. Husain, F. *OSACT4 Shared Task on Offensive Language Detection: Intensive Preprocessing-Based Approach*; European Language Resource Association: Marseille, France, 2020; Available online: https://aclanthology.org/2020.osact-1.8 (accessed on 28 September 2022).
27. Djandji, M.; Baly, F.; Antoun, W.; Hajj, H. Multi-Task Learning Using AraBert for Offensive Language Detection. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020; pp. 97–101.
28. Farha, I.A.; Magdy, W. Multitask Learning for Arabic Offensive Language and Hate-Speech Detection. In Proceedings of the OSACT 2020, Marseille, France, 12 May 2020.
29. Aldjanabi, W.; Dahou, A.; Al-Qaness, M.A.A.; Elaziz, M.A.; Helmi, A.M.; Damaševičius, R. Arabic Offensive and Hate Speech Detection Using a Cross-Corpora Multi-Task Learning Model. *Informatics* **2021**, *8*, 69. [CrossRef]
30. 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, Tokyo, Japan, 7–11 August 2017; pp. 1005–1008. [CrossRef]
31. 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]

32. 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]
33. Al-Khalifa, H.; Mubarak, H.; Al-Thubaity, A.; Magdy, W.; Darwish, K. UPV at the Arabic Hate Speech 2022 Shared Task: Offensive Language and Hate Speech Detection Using Transformers and Ensemble Models. In Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, Marseille, France, 20 June 2022; pp. 20–25.
34. Haddad, H.; Mulki, H.; Oueslati, A. T-HSAB: A Tunisian Hate Speech and Abusive Dataset. *Commun. Comput. Inf. Sci.* **2019**, *1108*, 251–263. [CrossRef]
35. Amazon Mechanical Turk. Available online: https://www.mturk.com/ (accessed on 1 February 2023).
36. Artstein, R.; Poesio, M. Inter-Coder Agreement for Computational Linguistics. *Comput. Linguist.* **2008**, *34*, 555–596. [CrossRef]
37. Voyant Tools. Available online: https://voyant-tools.org/ (accessed on 1 February 2023).
38. Kowsari, K.; Meimandi, K.J.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. *Informatics* **2019**, *10*, 150. [CrossRef]
39. HKUST-KnowComp/MLMA_hate_speech: Dataset and Code of Our EMNLP 2019 Paper "Multilingual and Multi-Aspect Hate Speech Analysis". Available online: https://github.com/HKUST-KnowComp/MLMA_hate_speech (accessed on 1 October 2022).
40. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. *arXiv* **2017**, arXiv:1703.04009. [CrossRef]
41. Robertson, S. Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [CrossRef]
42. Mehdad, Y.; Tetreault, J. Do Characters Abuse More Than Words? 2016. Available online: https://aclanthology.org/W16-3638 (accessed on 5 October 2020).
43. Schmidt, A.; Wiegand, M. A Survey on Hate Speech Detection Using Natural Language Processing. In Proceedings of the SocialNLP 2017—5th International Workshop on Natural Language Processing for Social Media, Proceedings of the Workshop AFNLP SIG SocialNLP, Valencia, Spain, 3 April 2017; pp. 1–10.
44. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to Fine-Tune BERT for Text Classification? In *Chinese Computational Linguistics*; Springer: Cham, Switzerland, 2019; Volume 11856, pp. 194–206. [CrossRef]
45. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *Proc. IEEE Int. Conf. Comput. Vis.* **2015**, *2015*, 19–27. [CrossRef]
46. Hedderich, M.A.; Lange, L.; Adel, H.; Strötgen, J.; Klakow, D. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Virtual, 6–11 June 2021; pp. 2545–2568. [CrossRef]
47. Nlpaug.Augmenter.Word.Context_word_embs—Nlpaug 1.1.11 Documentation. Available online: https://nlpaug.readthedocs.io/en/latest/augmenter/word/context_word_embs.html (accessed on 12 April 2022).
48. Kang, Z.; Grauman, K.; Fei, S. Learning with Whom to Share in Multi-Task Feature Learning. 2011. Available online: https://dl.acm.org/doi/10.5555/3104482.3104548 (accessed on 21 September 2022).
49. Long, M.; Cao, Z.; Wang, J.; Yu, P.S. Learning Multiple Tasks with Multilinear Relationship Networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 2017, pp. 1595–1604.
50. Dankers, V.; Rei, M.; Lewis, M.; Shutova, E. Modelling the Interplay of Metaphor and Emotion through Multitask Learning. In Proceedings of the EMNLP-IJCNLP 2019—2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2218–2229.
51. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
52. Welcome to Colaboratory—Colaboratory. Available online: https://colab.research.google.com/notebooks/intro.ipynb#recent=true (accessed on 3 February 2022).
53. Transformers. Available online: https://huggingface.co/docs/transformers/index (accessed on 20 January 2022).
54. Grandini, M.; Bagli, E.; Visani, G. Metrics for Multi-Class Classification: An Overview. *arXiv* **2020**, arXiv:2008.05756.
55. Mubarak, H.; Rashed, A.; Darwish, K.; Samih, Y.; Abdelali, A. Arabic Offensive Language on Twitter: Analysis and Experiments. In Proceedings of the WANLP 2021—6th Arabic Natural Language Processing Workshop, Kyiv, Ukraine, 19 April 2021; pp. 126–135.