algorithms

MDPI

Article

A Multitask Learning Framework for Abuse Detection and Emotion Classification

Yucheng Huang [1], Rui Song [2], Fausto Giunchiglia [3] and Hao Xu [1,2,*]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China; huangyc19@mails.jlu.edu.cn
[2] School of Artificial Intelligence, Jilin University, Changchun 130012, China; songrui20@mails.jlu.edu.cn
[3] Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy; fausto@disi.unitn.it
* Correspondence: xuhao@jlu.edu.cn

Abstract: The rapid development of online social media makes abuse detection a hot topic in the field of emotional computing. However, most natural language processing (NLP) methods only focus on linguistic features of posts and ignore the influence of users' emotions. To tackle the problem, we propose a **m**ultitask **f**ramework combining **a**buse detection and **e**motion classification (MFAE) to expand the representation capability of the algorithm on the basis of the existing pretrained language model. Specifically, we use bidirectional encoder representation from transformers (BERT) as the encoder to generate sentence representation. Then, we used two different decoders for emotion classification and abuse detection, respectively. To further strengthen the influence of the emotion classification task on abuse detection, we propose a cross-attention (CA) component in the decoder, which further improves the learning effect of our multitask learning framework. Experimental results on five public datasets show that our method is superior to other state-of-the-art methods.

Keywords: abuse detection; multitasking learning; emotion prediction; BERT

Citation: Huang, Y.; Song, R.; Giunchiglia, F.; Xu, H. A Multitask Learning Framework for Abuse Detection and Emotion Classification. *Algorithms* **2022**, *15*, 116. https://doi.org/10.3390/a15040116

Academic Editor: Frank Werner

Received: 1 March 2022
Accepted: 26 March 2022
Published: 28 March 2022

1. Introduction

1.1. Background

While the rapid growth of social media has brought people closer together, the amount of abuse language has grown at an alarming rate along with the number of users. The term abuse refers to all forms of expression that denigrate or offend an individual or group of people, including racism, sexism, personal attacks, harassment, cyberbullying, etc. [1], and abusive language also could be used on specific individuals or groups [2]. Abusive behavior online can lead to severe psychological consequences for its victims and hampers free speech. Automatic abuse detection can mine offensive language in a large number of online social comments, which is critical for avoiding psychological impact on victims, and thereby preventing hate crimes [3]. Therefore, how to detect abuse automatically becomes an important problem in the emotional computing field.

In general, abuse detection can be viewed as an online social text classification task. With the continuous development of natural language processing (NLP) technology, abuse detection can be divided into three different stages. Some of the earliest studies used logistic regression models for automatic abuse detection by using a variety of different features, such as character-level and word-level n-gram features, syntactic features, linguistic features, and comment-embedding features [4–7]. They represent traditional machine learning methods where the abuse detection results depend on the characteristics of a manual design. The second stage is a deep learning method represented by convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [8]. Deep learning methods do not rely on manual design features and can automatically capture the context features of abused language through neural networks. Thirdly, with the emergence of large-scale pretrained

language models, such as BERT [9] and HateBERT [10], the universality and transferability of knowledge in text have been further explored.

However, due to the variety of linguistic traits, the automatic detection of abuse is still very challenging [11]. For example, [12] divides abuse into explicitness and directness. Abuse of the former type manifests itself in a direct way, perhaps in the form of certain abusive words, but the latter type may cover metaphors or analogies which may lead to some possible confusion and errors. Moreover, most of the existing methods focus on how to model the features of linguistic properties, without fully taking into account the influence of users' emotions and psychological states on their expressions. Psychological research has proved that abusive language and HateBERT are also inseparable from the speaker's emotional and psychological state [13]. In addition, some studies in other fields of affective computing have also confirmed the relationship between users' emotions and their psychological traits, such as personality detection [14]. Therefore, the main purpose of this paper is to answer the following question: Can effective emotion classification facilitates abuse detection? To this end, we propose a multitask learning (level) method combining emotion classification and abuse detection to verify the learning effect of abuse detection by taking user emotion classification as an auxiliary task.

1.2. Motivations and Contributions

Our main motivation is to build a multitask framework that integrates emotion classification and abuse detection to improve the results of abuse detection through the auxiliary task. Different from existing multitask frameworks for emotion classification and abuse detection [1], our approach starts with an automatic emotion label generation because many datasets for abuse detection do not provide user emotion labels and manual labeling is too time-consuming and costly. Therefore, we use the most advanced pretrained model for sentiment analysis [15] to derive the emotion probability distribution for the text to be detected to generate the predictive data needed for MTL. Then, we use BERT as the encoder of the model to get the dynamic encoding of the text. Multihead self-attention ensures that the model can better capture context information. For two different tasks of abuse detection emotion classification, considering the different emphases of the two tasks, we use TextCNN [16] and BiLSTM [17] as decoders, respectively. In order to make better use of the benefits of emotion classification, we propose a cross-attention (CA) interaction mechanism similar to self-attention, further improving the effect of abuse detection.

Overall, our main contributions are as follows:

- We propose a multitask framework combining emotion classification and abuse detection to construct auxiliary tasks of emotion classification. We use a pretrained sentiment analysis model to derive emotion labels, thus avoiding a lot of manual labeling.
- We propose a decoding structure containing cross-attention to further enhance the positive effect of the auxiliary task on the primary task through the cross-attentional mechanism.
- Our label utilization approach is also easy to incorporate into other frameworks and take advantage of multitasking to improve the original model performance.
- We conduct a large number of experiments and show the superiority of our method compared with several most advanced algorithms.

The rest of the paper is organized as follows. Section 2 introduces some of the most relevant work to this paper, including multitasking learning and abuse detection. Section 3 introduces our proposed framework in detail. Section 4 illustrates the experimental process and results, the performance and key parameters of the model are further discussed. Finally, in Section 5, we draw conclusions and present future work.

2. Related Work

This section describes two aspects of multitask learning and abuse detection.

2.1. Multitask Learning

By sharing representations between related tasks, we can make our model better generalize the original task. This approach is called multitasking learning (MTL). In a broad sense, as long as we introduce more than one loss function in the training, it can be considered as multitask learning. In the context of deep learning, MTL has two different categories: hard or soft parameter sharing. The former shares all hidden layer parameters [18], while the latter has its own hidden layer parameters in each task, and a regularization term is optimized to determine the similarity among different tasks [19]. At present, MLT has been regarded as a universal method and has been applied to various fields such as natural language processing [20] and computer vision [21]. However, there are still challenges to domain-specific tasks, especially in the field of abuse detection.

2.2. Abuse Detection

The development of abuse detection can be divided into three stages: manual feature engineering, deep learning methods, and pretrained models.

The earliest abuse detection works used rules to train a classifier [22]. This work creates rules manually on the text to generate feature vectors for learning. Next, many methods based on manual features have been proposed. There are two main sources of features: text and user attributes. The former attempts to use lexicon-based features [23], bag-of-words (BOW) or N-gram features [24] to extract features from users' posts on the network. Ref. [25] also shows that the dense comment representations generated by paragraph2vec are superior to the bag-of-words feature. The latter tries to infer the likelihood of abuse based on the user's age [26], time of publication [27], and so on.

With the development of deep learning, some works have used deep neural architecture to achieve impressive results on various abuse detection datasets. Ref. [28] reports different deep learning methods on a Twitter dataset, including CNN and LSTM with random embeddings and GloVe. Ref. [8] proposes a transformed word embedding model by pretrained word embeddings and max/mean pooling from simple, fully connected transformations of these embeddings. This approach can easily be extended to some unseen or rare tokens in the training dataset by projecting the pretrained embedding into a space that the encoder can understand. Some studies also try to extract better features by combining the structure of CNN and RNN [29]. Because abuse detection often contains some cryptic expressions, character-level methods also show effectiveness [30]. More recently, some studies have also expanded the application scenarios of abuse detection by studying cross-language and cross-domain aspects [31–33].

Some works have also taken advantage of pretrained models to greatly improve abuse detection by pretraining large datasets of abuse. Ref. [10] collects a large dataset banned for being offensive, offensive or hateful comments on Reddit, and generates a pretrained model called HateBERT. HateBERT outperforms the corresponding general BERT model. In addition, [34] also enhanced the results through the use of German BERT, pretrained from German Wikipedia and German Twitter corpora.

More recently, some works have begun to explore the role of emotion in abuse language detection [1,3]. However, they rely on additional annotation of data and cannot be extended effectively.

3. Proposed Method

In this section, we introduce the proposed method in detail. Our model can be divided into three main parts as shown in Figure 1: emotional label generation, encoder and decoder.

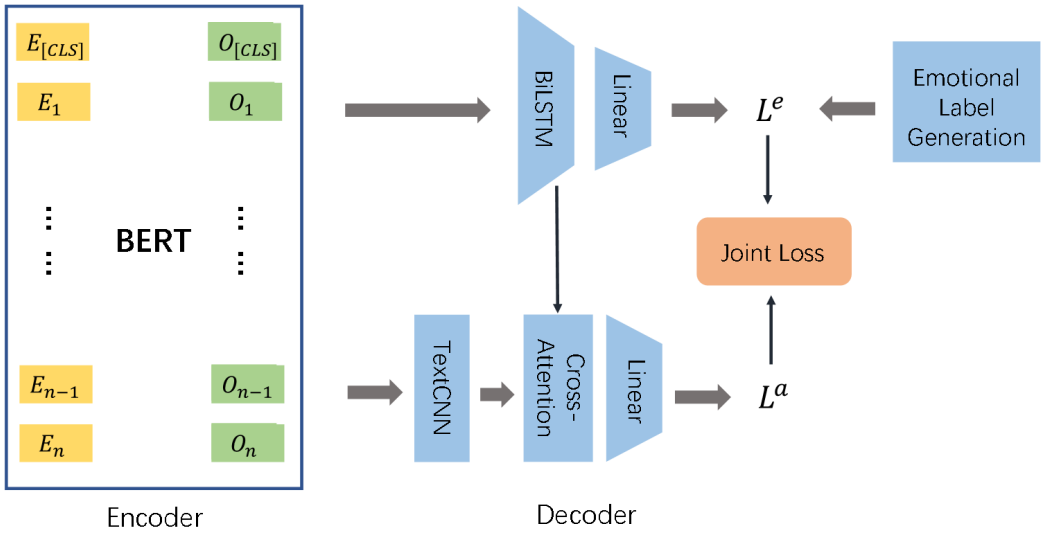

Figure 1. Detailed model structure, which consists of emotional label generation, encoder and decoder. The emotional label generation is a pretrained sentiment analysis model to derive emotion labels. The encoder module encodes the input text and the encoder is initialized with BERT. The decoder module uses TextCNN and BiLSTM for two tasks of abuse detection and emotion classification, and contains a cross-attention interaction mechanism to further improve the effect of abuse detection.

3.1. Notations

First, for the sake of illustration, we give some symbolic definitions. Given a text corpus T and any text t_i in it, the main purpose of MTL is to learn a mapping function $\chi : t_i \rightarrow \{y_i^a, y_i^e\}$, where $y_i^a \in Y^a$ and $y_i^e \in Y^e$ represent the label space of abuse detection and sentiment classification, respectively. It is important to note that in our model, $y_i^e = (p_i^e, 1 - p_i^e)$ is a mutually exclusive probability pair, used to represent the probability of positive and negative emotions. So both of the label space sizes of the two different tasks are 2. For the rest of this article, we will use upper-case letters for sets or tensors and lower case letters for individual samples or vectors.

3.2. Emotional Label Generation

Before building the model, we need to prepare labels for the emotion classification task. However, most abuse detection datasets have no corresponding emotional labels, and manual labeling, while effective, is costly in time and labor. Therefore, we explored automatic labeling of emotion classification.

Some previous works have focused on resource creation or sentiment categorization for specific tasks and domains [35–37]. They constructed dictionaries associated with different psychological traits, judging emotional content by specific words. However, the word-dependent approaches often face the out-of-vocabulary (OOV) problem, and the overly fine-grained emotional dimension of words often hinders the accurate judgment on the emotional polarity of the whole sentence. Therefore, inspired by some transfer learning methods [38], we adopted the pretrained model SKEP (https://github.com/baidu/Senta) (accessed on 23 March 2022) [15] as the derivation method of emotion dichotomy labels. SKEP uses the pretrained model with enhanced emotional knowledge to comprehensively surpass state-of-the-art methods in 14 typical emotional analysis tasks.

However, the abuse of language detection datasets tends to be associated with strong negative emotions, so the probability of negative emotions remained high across all datasets. This resulted in a serious sample disequilibrium, which affected the generalization ability of the model. In order to reduce the influence of the sample distribution, we used the probability of the last SKEP layer output as the soft label of emotion classification rather than 0 or 1, so that the objective of our auxiliary task was to approximate this probability.

3.3. Encoder

To capture complex semantic information within a sentence, we used a fine-tuned BERT [9] as the encoder to get a vectorized representation of each word. In order to accelerate the convergence speed of the model and obtain a more stable representation, we fixed the gradient of the first 8 BERT layers and only updated the parameters of the last 4 layers as advised by [39]. We then represented the sentence $X \in \mathcal{R}^{m*n*768}$ after BERT with a tensor and fed it to the decoder, where m denotes the sample size and n denotes the maximal sentence length. To prevent sign abuse, we omitted the subscript i, which stands for the ith sentence.

3.4. Decoder

For different tasks, we used BiLSTM and TextCNN as two main components of different decoders because of their different focus. In addition, in order to better facilitate abuse detection by emotion classification, we propose a cross-attention enhancement component.

Decoder for Emotion Classification. A two-layered bidirectional long short-term memory (LSTM) network was applied to BERT's output to obtain word contextualized representations. For a LSTM output $H^e = \{h_0, h_1, \ldots, h_n\}$, the BiLSTM further represents each token as:

$$h_t^e = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{1}$$

where $\overrightarrow{h_t}; \overleftarrow{h_t} \in \mathbf{R}^d$ and d denotes the hidden size of the BiLSTM. We used a maximum pooling to get a representation of the entire sentence:

$$H_{max}^e = max(h_1; h_2; \ldots, h_n) \tag{2}$$

Then, the probability distribution representation of the emotional label was generated by a full connection layer with a softmax activation function:

$$P^e = softmax(H_{max}^e W^e + b^e) \tag{3}$$

where $P^e \in \mathbf{R}^{m*2}$.

Decoder for Abuse Detection. TextCNN [16] was used to obtain finer local context features, and a representation similar to n-gram is obtained by adjusting the size of the convolution kernel. We then employed dynamic maximum pooling to capture sentence level representations of varying granularity. The output of textCNN is represented as:

$$H_k^a = TextCNN_k(X) \tag{4}$$

For different convolution kernels k, we obtained different sentence-level representations $\{H_0^a, \ldots, H_k^a\}$. In the actual operation, we chose three different convolution kernels of size $\{2, 4, 6\}$ for StormW and $\{2, 3, 4\}$ for the others. The final expression of the sentence was the concatenation of the output of the three kernels:

$$H^a = [H_2^a; H_3^a; H_4^a] \tag{5}$$

where $H^a \in \mathbf{R}^{d'*3}$ and d' is the number of output channels of the CNN. As with the emotion classification decoder, we used a full connection layer with softmax to obtain the probability distribution:

$$P^a = softmax(H^a W^a + b^a) \tag{6}$$

Cross Attention (CA). Although the input of BiLSTM and TextCNN X are derived from BERT, the results should be more focused on different subtasks after the decoder. CA is designed to capture the actual impact of the emotion classification subtask on the main task. Like self-attention [40,41], cross attention can essentially be described as a mapping

from a query to a set of key–value pairs, except that the key is derived from the BiLSTM's output H^e:

$$\begin{aligned} K &= XW^K \\ Q &= H^e W^K \\ V &= XW^V \end{aligned} \tag{7}$$

On this basis, CA was calculated by:

$$CA(K, Q, V) = softmax(\frac{QK^T}{\sqrt{d_{ca}}}V) \tag{8}$$

To ensure consistency of dimensions, we set $d_{ca} = d = 768$, $K, Q, V \in \mathbf{R}^{768\times768}$. The output of the CA was fed to the normalization layer:

$$H^{ca} = norm(CA(K, Q, V)) \tag{9}$$

Finally, we replaced X in (4) with the emotional classification augmented results H^{ca}. The overall flow of cross attention is shown in Figure 2. CA can effectively control the effect of the results of affective classification on abuse detection, and we further explore this in the ablation study Section 4.5.2.

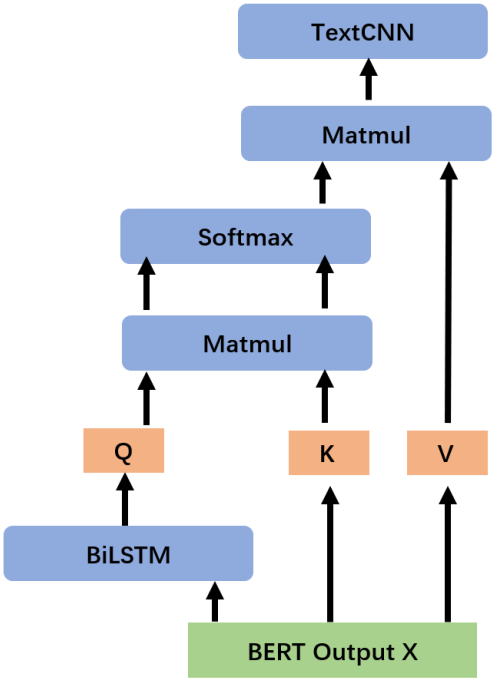

Figure 2. Cross attention (CA) process. The inputs of CA are derived from the BERT encoder, and CA can essentially be described as a mapping from a query to a set of key–value pairs, where the key is derived from BiLSTM's output. Please see Section 3.4 for calculation details.

3.5. Joint Loss

For the two different tasks, binary cross entropy (BCE) was used as the loss function:

$$Loss = -\frac{1}{M}\sum_{i=1}^{M} y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i)) \tag{10}$$

where M is the size of training set, y denotes the ground truth, and $p(y)$ denotes the predicted label. The total loss function of the model was the combined loss of the two:

$$\mathbf{L} = \mathbf{L}^a + \lambda \mathbf{L}^e \tag{11}$$

4. Experiment

In this section, we describe the experimental details, including the datasets required for the experiment, the comparison algorithms, the experimental results and further analysis.

4.1. Datasets

The different datasets are described and the statistics of all datasets are shown in Table 1. To be fair, we treated all the datasets uniformly, removing the URL field containing "http" and removing the "#" from the tags on datasets from Twitter. After that, we converted all the characters to lowercase and treated all the samples to a uniform length of 50. For samples greater than 50, we truncated them; for those less than 50, we added "[PAD]" at the end.

Table 1. Characterization of the datasets. The table below details the distribution of the number and composition of abuse language across the various datasets.

Datasets	Size	Composition
HatEval	12,000	Hateful (42.08%), Non-hateful (57.92%)
Davids	24,783	Hate (5.77%), offensive (77.43%), Neither (16.80%)
OffEval	14,100	Offensive (33%), Not-offensive (67%)
FNUC	1528	Hateful (28.50%), Non-hateful (71.50%)
StormW	10,944	Hate (10.93%), Nohate (89.07%)

- HatEval [42], a Twitter-based hate speech dataset released in the SemEval-2019 (https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/) (accessed on 27 March 2022) mission. Its English portion provides a sentence-level commentary on hate speech against immigrants and women. Only hateful messages targeting migrants and women belong to the positive class, leaving any other message (including offensive or abusive messages against other targets) to the negative class. The training set is composed of 9000 messages and the test contains 3000 messages.
- Davids [43], a Twitter-based online abusive language detection dataset, which includes three classes, Hate, Offensive or Neither based on the hate speech lexicon from Hatebase.org. Complying with [44], we took Hate and Offensive as the positive class, Neither as the negative class.
- OffEval [45]; it contains three subtasks A, B, and C, and we focused on subtask A, which is about offensive language identification. It contains 13,240 tweets, and each tweet is classified as to whether it is offensive (33%) or not (67%). It has 13,240 pieces of training data, and 840 pieces are dedicated to testing.
- FNUC [46], a lightweight hate speech detection dataset collected from complete Fox News discussion threads, and annotated with the general level categories Hateful or Non-hateful. It contains 1528 records.
- StormW [47], a Stormfront-based hate language detection dataset with general-level labels Hate and NoHate. Stormfront is a supremacist forum that promotes white nationalism and antisemitism.

4.2. Baselines

We report the baseline algorithm against which the proposed model is compared.

- Hybrid CNN [48], a hybrid CNN (word-level and character-level) model was applied to abusive tweet detection. In addition, we also implemented a word-level CNN and character-level CNN for comparison.

- Multifeatures with RNN [49], a hybrid character-based and word-based recursive neural network (RNN) model was used to detect abusive language. In addition, BiLSTM and BiLSTM Attention, the baseline method of text classification in this paper, were also used for further comparison.
- Character-based transformer [50], a character-level transformer-based classifier for harassment detection on Twitter posts.
- BERT [9], a fine-tuned BERT model, consistent with the encoder used in our model. We used the maximum pooling of the output of the last layer as the document characteristic and the output features were used to train the text classifier.
- HateBERT [10], a re-trained BERT model for abusive language detection in English. The model was trained on RAL-E, a large-scale dataset of Reddit comments in English from communities banned for being offensive, abusive, or hateful.
- MTL [1], a multitask framework for emotion detection and abuse detection based on BiLSTM, which proposes three different BiLSTM-based decoder structures. *Hard* consists of an encoder shared and updated by two tasks, followed by task-specific branches. *Double Encoder Model* has a task-specific two-layered BiLSTM encoder for each task. *Gated Double Encoder Model* uses gate to control information flow between different task encoder. The three variants are written as MTL_{Hard}, $\text{MTL}_{DEncoder}$ and $\text{MTL}_{GatedDEncoder}$, respectively. To be fair, we extended MTL with our label generation approach to verify the effectiveness of our proposed multitasking approach.

4.3. Details

We chose the PyTorch version of BERT (https://huggingface.co/bert-base-uncased) (accessed on 27 March 2022) and HateBERT (https://osf.io/ryvs8/) (accessed on 27 March 2022) to implement the encoder. For all word-level baselines, we adopted Glove300 as word features. We used the data split standard provided in the original papers, and if the dataset split standard was not mentioned in the original paper, we used a 10-fold cross-validation. We used Adam as the optimizer, set the epoch to 20 and the maximum document length to 50. When the performance did not improve after 5 epochs, we stopped the model early. Although different batches and learning rates may slightly affect the results of the model, to be fair, we set the batch size to 64, except for the HatEval dataset that was set to 256, because a small batch greatly reduced the convergence rate of the model over HatEval. In addition, we used gradient clipping [51] and set the maximum norm of the gradient to 1 to prevent gradient disappearance/gradient explosion due to the particularity of the dataset, leading to the model being unable to update. We obtained the optimal λ for different datasets. Detailed parameter settings are shown in Table 2.

Table 2. Parameter settings on different datasets.

Datasets	HatEval	Davids	OffEval	FNUC	StormW
Epoch	20	20	20	20	20
Mmax length	50	50	50	50	50
Learning rate	2×10^{-5}	2×10^{-5}	2×10^{-5}	2×10^{-5}	2×10^{-5}
Batch size	256	64	64	64	64
λ	0.9	0.6	0.9	0.5	0.8

4.4. Results

We tested the performance of the proposed framework from the two aspects of test accuracy and weighted F1 value. The specific results are shown in Tables 3 and 4.

Accuracy and weighted F1. First, we note that compared to character-level methods, word-level methods could achieve better results regardless of accuracy or weighted F1 value. We attribute its effectiveness to the semantic information contained in the pretrained static word vectors (GloVe), whereas in the character-level approaches, we needed to randomly assign the character vector. This inspired us to seek more effective word/sentence feature representation, so we used a pretrained BERT as our encoder.

Second, we found that the MTL architectures with emotion classification tasks had obvious advantages over the RNN-based approaches. Although hyper-RNN can learn more about abuse detection through character-level features than ordinary word-level RNN methods, it was obvious that the emotion classification auxiliary task helped more. This inspired us to use additional multitasking frameworks to facilitate the abuse detection results. At the same time, it also verified the validity of our proposed emotion classification label generation method.

Thirdly, our method achieved optimal results in all datasets regardless of accuracy or weighted F1 values, especially compared with BERT. This showed that the proposed multitask learning method can be further be expanded on the basis of BERT. In addition, due to the small scale of FNUC, BERT was easy to overfit, so its effect was even inferior to CNN, RNN and other methods. However, the addition of the emotion classification task can effectively improve the overfitting problem of pretrained models on small datasets, which further explains the necessity to introduce multitasking.

Macro F1. We also compared our model with other multitask abuse detection models and HateBERT, a pretrained model for abuse detection. Since macro F1 was used as the evaluation in [10], we also used macro F1 to reevaluate our model. First, for the OffEval dataset, our multitasking approach was better because the BERT encoder was more advantageous. Compared with HateBERT, our method was also competitive, suggesting that sentiment analysis can indeed promote the results of abuse detection, even comparable to the domain-pretraining model. Therefore, we also explored the ability of our method to combine with HateBERT in subsequent analysis. It should be noted that HateBERT achieved a significant improvement on HatEval due to the usage of gradient clipping. We tried to remove the gradient clipping in the actual experiment and achieved a result of 53%, which was similar to the result of [10].

Table 3. Weighted F1 score (%) for different models on five datasets. The optimal results are indicated in bold. The table lists the character-level methods, word-level methods, MTL, BERT, and our models, respectively, and the results show that our model (MFAE) achieves the best performance.

Models/Datasets	HatEval	Davids	OffEval	FNUC	StormW	Average
Word-level BiSLTM	60.59	92.42	76.77	66.91	89.18	77.174
Word-level BiSLTM Attention	61.90	92.99	79.73	66.19	88.94	77.95
Character-level BiLSTM	52.07	93.82	75.59	66.02	88.39	75.178
Hyper-RNN [49]	57.85	95.55	80.04	70.78	90.13	78.87
Word-level CNN	60.31	91.54	77.27	71.31	89.09	77.904
Character-level CNN	53.40	93.56	73.73	66.58	85.88	74.63
Hyper CNN [48]	56.36	93.72	76.44	69.88	88.95	77.07
Character-level Transformer [50]	63.28	92.33	79.09	70.40	89.68	78.956
MTL [1]						
MTL_{Hard}	62.59	92.93	80.12	71.13	89.15	79.184
$MTL_{DEncoder}$	62.62	92.57	76.74	67.62	89.71	77.852
$MTL_{GatedDEncoder}$	60.17	92.65	79.79	70.49	89.18	78.456
BERT [9]	62.08	95.88	83.59	69.36	90.71	80.324
MFAE	**64.65**	**96.49**	**84.43**	**73.08**	**91.60**	**82.05**

Table 4. Accuracy score (%) for different models on five datasets. The optimal results are indicated in bold.

Models/Datasets	HatEval	Davids	OffEval	FNUC	StormW	Average
Word-level BiSLTM	60.81	89.57	78.26	71.71	90.28	78.126
Word-level BiSLTM Attention	59.73	91.22	80.12	72.37	90.50	78.788
Character-level BiLSTM	52.05	93.77	78.37	72.36	90.37	77.384
Hyper-RNN	58.21	95.54	80.47	73.03	90.96	79.642
Word-level CNN	60.20	91.59	79.53	73.68	90.69	79.138
Character-level CNN	53.73	93.24	77.79	73.02	89.95	77.546
Hyper CNN	57.27	93.64	78.83	72.37	90.73	78.568
Character-level Transformer [50]	63.43	92.41	80.47	73.03	90.50	79.968
MTL [1]						
MTL_{Hard}	62.53	93.12	81.27	74.34	90.73	80.398
$MTL_{DEncoder}$	62.39	92.69	79.53	71.71	90.78	79.42
$MTL_{GatedDEncoder}$	59.87	92.59	80.81	71.70	90.23	79.04
BERT [9]	61.12	95.84	84.30	73.03	91.14	81.086
MFAE	**63.55**	**96.25**	**84.89**	**75.10**	**91.79**	**82.316**

4.5. Analysis

We further analyzed the model, including testing the ability of our method to combine with HateBERT, conducting ablation studies and a discussion of selection for key parameters λ. Then, a case study was conducted to explore the impact of emotion classification on abuse detection.

4.5.1. Combine with HateBERT

By replacing the encoder with HateBERT, we explored the ability of our method to combine with HateBERT. We guaranteed that all parameters were set in accordance with Table 2 and we used macro F1 as the metric. The overall results are shown in Figure 3. As we can see, multitask learning improved the performance on all datasets, which means that our multitask learning framework is easy to expand and effective. However, for OffEval, StormW, and Davids, the improvement was not large. Although we did not conduct further parameter adjustment according to HateBERT, it can also be seen that compared with BERT, HateBERT's ability to combine with the task of emotion classification was not strong. This may be due to the fact that HateBERT had been pretrained by data from abuse detection domain, which made it less sensitive to external multitask adjustment. That is why we chose BERT as our base encoder instead of HateBERT, although HateBERT is theoretically more expressive. We will also explore HateBERT's performance further in future work.

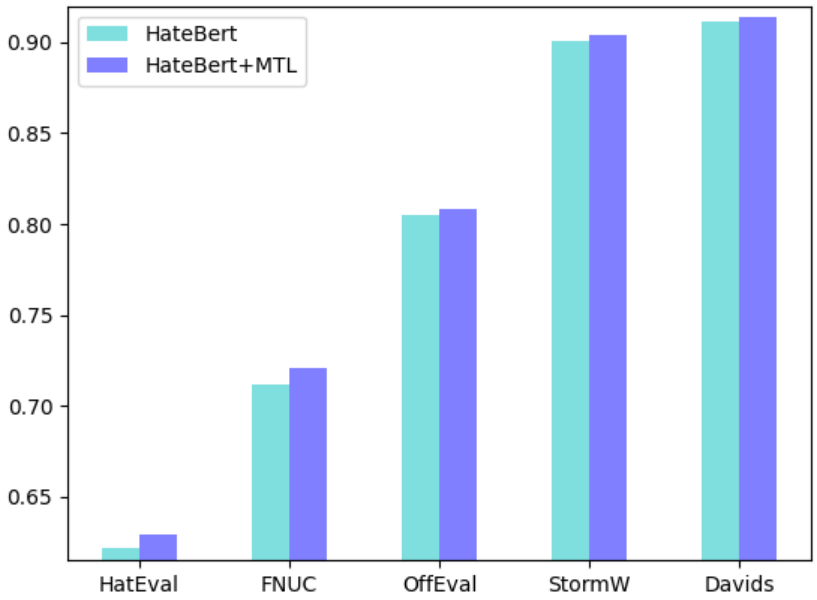

Figure 3. Our framework combined with HateBERT. The results show that multitask learning improves performance on all datasets and our framework is easy to expand.

4.5.2. Ablation Study

To demonstrate the role of the core components in the model, we performed ablation experiments. Specifically, we further propose variations of four multitask models.

- Without cross attention (WO CA): we removed the CA component in the decoder, and kept the rest consistent with the original model, including parameters.
- Without decoder (WO Decoder): we removed all components from the decoder and used two different maximum pooling and linear transformations as outputs for the two different tasks.
- Without BiLSTM (WO BiLSTM): we removed the BiLSTM component in the decoder.
- Without TextCNN (WO TextCNN): we remove the TextCNN component in the decoder.

The specific results comparison is shown in Figures 4 and 5. In most cases, removing any component has a negative impact on model performance, except for the OffEval dataset, which illustrates the validity of our decoder and CA component. Specific emotion classification tasks can positively influence abuse detection results, which is the essence of our proposed approach. However, it can be seen that after the removal of decoder (WO decoder), although the performance of the model is still improved compared with BERT, the degree of improvement is no longer obvious. This may be because our emotion classification datasets are derived from the existing pretrained model, which contains some noise and reduces the effect of the multitask learning. In the future, we will continue to explore how to reduce the noise of false tags or use better derivation methods.

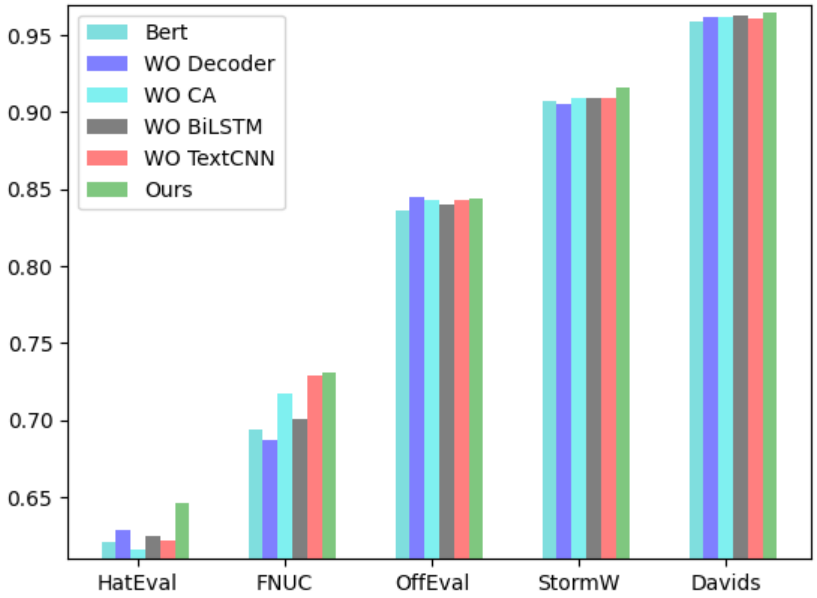

Figure 4. Weighted F1 results for the ablation study. We removed different components from model separately, such as decode, CA, BiLSTM, TextCNN. The results show that removing any component has a negative impact on model performance.

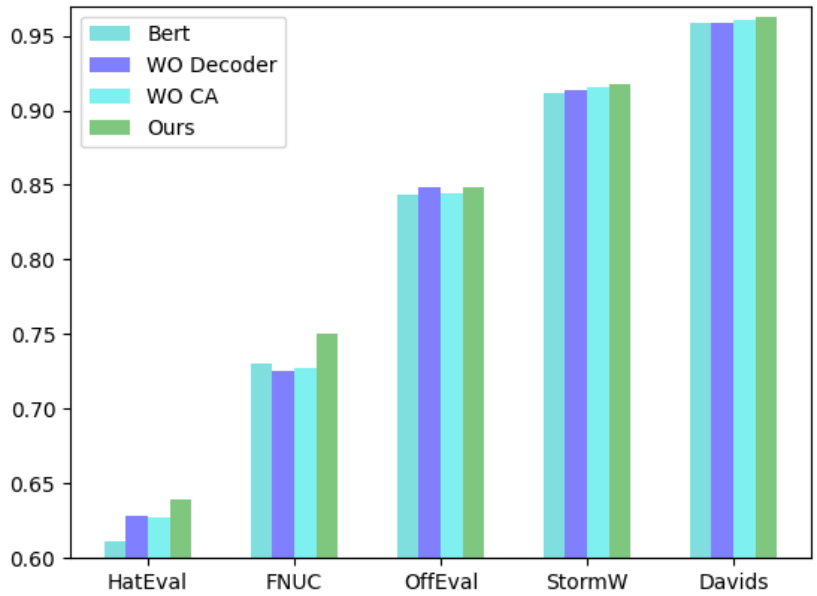

Figure 5. Accuracy results for the ablation study.

4.5.3. Key Parameter Selection

We tested the model performance under different λ values to explore the effect of the auxiliary task of emotion classification on abuse detection. As is shown in Figure 6, we searched for the optimal value of the parameter λ in $[0.1, 0.2 \ldots, 1]$ and drew line charts with the change of weighted F1 value. We found that for all datasets, the model performance showed an overall upward trend with the increase of λ, which highlighted the importance of the task of auxiliary emotion classification because the larger the λ, the more significant the impact on the final prediction results. This inspired us to give it more attention (usually greater than 0.5), when building auxiliary tasks to facilitate abuse detection.

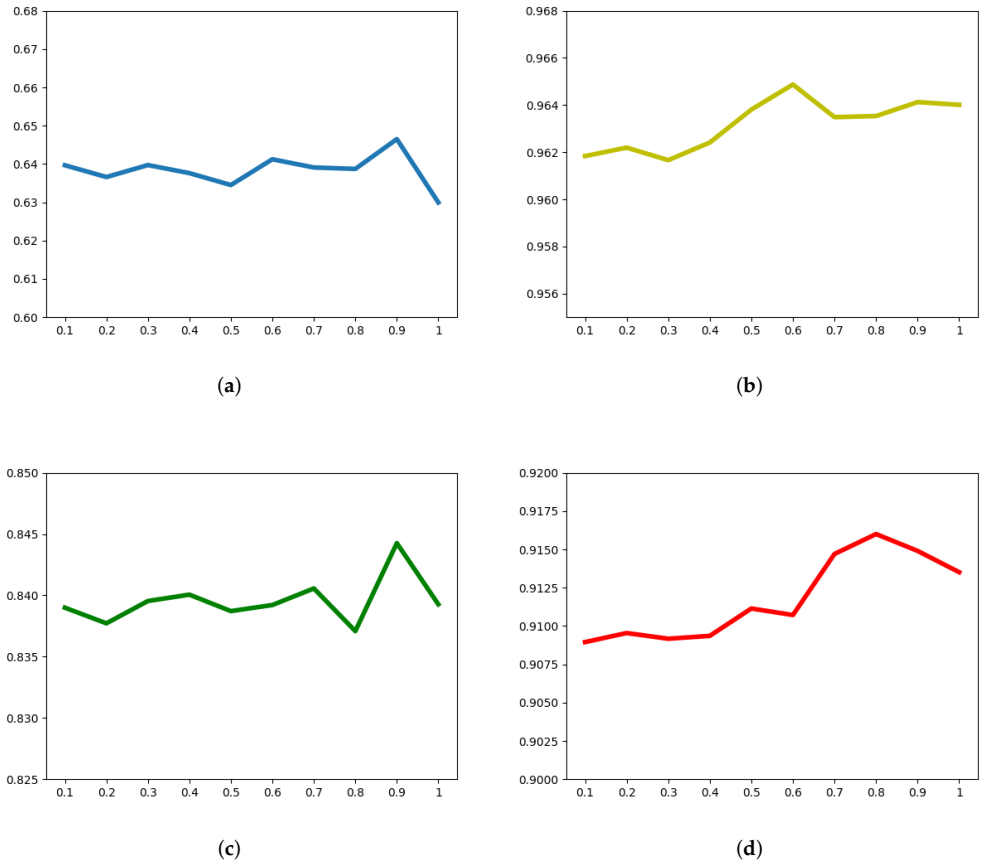

Figure 6. Weighted F1 values with different λ on four datasets. (**a**) HatEval, (**b**) Davids, (**c**) OffEval, (**d**) StormW.

4.5.4. Case Study

We selected some samples incorrectly identified by BERT from the Hatval dataset for case studies as shown in Table 5. Some words with negative emotional tendencies are prevalent in the samples, such as b**ch (samples 1, 3, 4 and 5), stupid (sample 2), etc. However, in some cases, it is not offensive or aggressive, but rather a colloquial statement (samples 1 and 2), which leads to a neutral emotion. Moreover, some abuse language tends to show strong negative emotions (samples 3 and 4), so its probability of being negative is much higher than that of being positive. Therefore, the introduction of the emotion classification task can establish a positive relationship to abuse detection. However, we found from sample 5 that emotion labeling is sometimes unreliable because abuse language can be deceptive. The appearance of words such as "kind" and "love" skews the results of emotion classification towards the positive aspect, which can be misleading about the results of abuse detection. Therefore, affective classification is best used as an auxiliary task to provide some positive guidance for abuse detection.

Table 5. Case study. Some test samples which contain abuse language leading to a neutral emotion (samples 1 and 2). Most abuse language tends to show strong negative emotions (samples 3 and 4). In rare cases, the emotion labeling is unreliable (sample 5).

No.	Test Sample	Abuse or Not	Positive Probability	Negative Probability
1	@Hoe_YouAshy friends? that is my BLOOD brother b**ch… he in yo city !! i'm hours away. ill be through there tomo…	No	56.5%	43.5%
2	@Patriot_gay The magic wand? Choice! O chose NOT TO help us, POTUS TRUMP chose TO help us. ITS THE ECONOMY STUPID! #MAGA #NoDACA #howdoyoulikeusnow	No	55.5%	44.5%
3	I'm tired of these h**s saying us ni**as ai not nah b**ch you ai not sh*t either tf	Yes	5.6%	94.4%
4	MOTHERF**KING B**CH D**K CUNTBAG LARD EATING *SS SU… —i'm gonna make this my bio tell me who this is i wa…	Yes	19.0%	81.0%
5	me: be kind to everyone x spread love and positivity x me after a b**ch crosses my path: rot in hell you dumb wh**e	Yes	57.7%	42.3%

5. Conclusions and Future Work

In this paper, we proposed a new multitask framework for emotion classification and abuse detection. We derived the emotion labels from the existing pretrained emotion analysis model and proposed a decoder component based on cross attention, which effectively utilizeds the pseudo-label information containing noise. The decoder component used TextCNN and BiLSTM for two different tasks of abuse detection and emotion classification, and we performed ablation experiments, showing that each subcomponent in the decode module was indispensable. We verified the validity of the framework on five public datasets. Results on five datasets showed that our model (MFAE) outperformed other methods. Our method does not need a lot of auxiliary task-marking data, so it has good scalability.

In future work, we will use the pseudo-label information containing noise in a more reasonable way and expand the simple emotion dichotomy task to multiple classification. In addition, we will also continue to study how to combine with additional pretrained models.

Author Contributions: Y.H.: conceptualization, methodology, software, data curation, writing—original draft preparation, writing—reviewing and editing; R.S.: visualization, investigation; F.G.: software, validation; H.X.: supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data underlying this article are available in the article.

Acknowledgments: This work was supported by National Natural Science Foundation of China (NSFC), "From Learning Outcome to Proactive Learning: Towards a Humancentered AI Based Approach to Intervention on Learning Motivation" (No. 62077027).

Conflicts of Interest: The authors certify that there is no conflict of interest in the subject matter discussed in this manuscript.

References

1. Rajamanickam, S.; Mishra, P.; Yannakoudakis, H.; Shutova, E. Joint Modelling of Emotion and Abusive Language Detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Washington, DC, USA, 5–10 July 2020; pp. 4270–4279.
2. Chandra, M.; Pathak, A.; Dutta, E.; Jain, P.; Gupta, M.; Shrivastava, M.; Kumaraguru, P. AbuseAnalyzer: Abuse Detection, Severity and Target Prediction for Gab Posts. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6277–6283.
3. Vrysis, L.; Vryzas, N.; Kotsakis, R.; Saridou, T.; Matsiola, M.; Veglis, A.; Arcila-Calderón, C.; Dimoulas, C. A Web Interface for Analyzing Hate Speech. *Future Internet* **2021**, *13*, 80. [CrossRef]
4. Chen, Y.; Zhou, Y.; Zhu, S.; Xu, H. Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. In Proceedings of the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, Amsterdam, The Netherlands, 3–5 September 2012; pp. 71–80.
5. Hee, V.C.; Lefever, E.; Verhoeven, B.; Mennes, J.; Desmet, B.; Pauw, D.G.; Daelemans, W.; Hoste, V. Detection and fine-grained classification of cyberbullying events. In Proceedings of the International Conference Recent Advances in Natural Language Processing, Hissar, Bulgaria, 5–11 September 2015; pp. 672–680.
6. Waseem, Z.; Hovy, D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), San Diego, CA, USA, 12–17 June 2016; pp. 88–93.
7. Wulczyn, E.; Thain, N.; Dixon, L. Ex Machina: Personal Attacks Seen at Scale. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1391–1399.
8. Kshirsagar, R.; Cukuvac, T.; McKeown, K.; McGregor, S. Predictive Embeddings for Hate Speech Detection on Twitter. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October–1 November 2018; pp. 26–32.
9. 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, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
10. Caselli, T.; Basile, V.; Mitrović, J.; Granitzer, M. HateBERT: Retraining BERT for Abusive Language Detection in English. In Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), Bangkok, Thailand, 6 August 2021; pp. 17–25.
11. Zheng, L.; Jiang, P.; Qiao, L.; Xi, L. Challenges and frontiers of manufacturing systems. *Jixie Gongcheng Xuebao/J. Mech. Eng.* **2010**, *46*, 124–136. [CrossRef]
12. Waseem, Z.; Davidson, T.; Warmsley, D.; Weber, I. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Language Online*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 78–84.
13. Patrick, G.T.W. The Psychology of Profanity. *Philos. Rev.* **1901**, *10*, 113. [CrossRef]
14. Ren, Z.; Shen, Q.; Diao, X.; Xu, H. A sentiment-aware deep learning approach for personality detection from text. *Inf. Process. Manag.* **2021**, *58*, 102532. [CrossRef]
15. Hao, T.; Can, G.; Xinyan, X.; Hao, L.; Bolei, H.; Hua, W.; Haifeng, W.; Feng, W. SKEP: Sentiment Knowledge Enhanced Pre-training for Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Washington, DC, USA, 5–10 July 2020; pp. 4067–4076.
16. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
18. Caruana, R. Multitask Learning: A Knowledge-Based Source of Inductive Bias. *ICML* **1993**, *28*, 41–48.
19. Duong, L.; Cohn, T.; Bird, S.; Cook, P. Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 25–31 July 2015; pp. 845–850.
20. Subramanian, S.; Trischler, A.; Bengio, Y.; Pal, J.C. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
21. Liu, S.; Johns, E.; Davison, J.A. 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.

22. Spertus, E. Smokey: Automatic recognition of hostile messages. In Proceedings of the Conference on Innovative Applications of Artificial Intelligence (IAAI), San Francisco, CA, USA, 27–31 July 1997; pp. 1058–1065.
23. Gitari, D.N.; Zuping, Z.; Damien, H.; Long, J. A Lexicon-based Approach for Hate Speech Detection. *Int. J. Multimed. Ubiquitous Eng.* **2015**, *10*, 215–230. [CrossRef]
24. Sood, O.S.; Antin, J.; Churchill, F.E. Using Crowdsourcing to Improve Profanity Detection. In Proceedings of the AAAI Spring Symposium: Wisdom of the Crowd, Palo Alto, CA, USA, 26–28 March 2012; pp. 69–74.
25. Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.; Bhamidipati, N. Hate Speech Detection with Comment Embeddings. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 29–30.
26. Dadvar, M.; Trieschnigg, D.; Ordelman, R.; Jong, d.F. Improving cyberbullying detection with user context. In Proceedings of the ECIR'13—35th European conference on Advances in Information Retrieval, Moscow, Russia, 24–27 March 2013; pp. 693–696.
27. Galán-García, P.; Puerta, G.d.l.J.; Gómez, L.C.; Santos, I.; Bringas, G.P. Supervised Machine Learning for the Detection of Troll Profiles in Twitter Social Network: Application to a Real Case of Cyberbullying. *Log. J. IGPL* **2016**, *24*, 42–53.
28. Badjatiya, P.; Gupta, S.; Gupta, M.; Varma, V. Deep Learning for Hate Speech Detection in Tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 759–760.
29. Wang, C. Interpreting Neural Network Hate Speech Classifiers. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October–1 November 2018.
30. Mishra, P.; Yannakoudakis, H.; Shutova, E. Neural Character-based Composition Models for Abuse Detection. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October–1 November 2018; pp. 86–92.
31. Fortuna, P.; Soler-Company, J.; Wanner, L. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Inf. Process. Manag.* **2021**, *58*, 102524. [CrossRef]
32. Pamungkas, W.E.; Basile, V.; Patti, V. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Inf. Process. Manag.* **2021**, *58*, 102524. [CrossRef]
33. Pamungkas, W.E.; Basile, V.; Patti, V. Misogyny Detection in Twitter: A Multilingual and Cross-Domain Study. *Inf. Process. Manag.* **2020**, *57*, 102360. [CrossRef]
34. Paraschiv, A.; Cercel, D.C. UPB at GermEval-2019 Task 2—BERT-Based Offensive Language Classification of German Tweets. In Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), Erlangen, Germany, 9–11 October 2019.
35. Pennebaker, J.W.; Francis, L.E.; Booth, R.J. *Linguistic Inquiry and Word Count (LIWC): LIWC2001*; Lawrence Erlbaum Associates: Mahwah, NJ, USA, 2001.
36. Strapparava, C.; Valitutti, A. WordNet Affect: An Affective Extension of WordNet. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal, 26–28 May 2004; European Language Resources Association (ELRA): Lisbon, Portugal, 2004.
37. Cambria, E.; Poria, S.; Hazarika, D.; Kwok, K. SenticNet 5: Discovering Conceptual Primitives for Sentiment Analysis by Means of Context Embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1795–1802.
38. Cevher, D.; Zepf, S.; Klinger, R. Towards Multimodal Emotion Recognition in German Speech Events in Cars using Transfer Learning. In Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, 9–11 October 2019.
39. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to Fine-Tune BERT for Text Classification? In Proceedings of the China National Conference on Chinese Computational Linguistics, Changsha, China, 19–21 October 2019; pp. 194–206.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, N.A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
41. Wang, K.; Lu, D.; Han, C.S.; Long, S.; Poon, J. Detect All Abuse! Toward Universal Abusive Language Detection Models. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6366–6376.
42. Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Pardo, M.R.F.; Rosso, P.; Sanguinetti, M. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 54–63.
43. Davidson, T.; Warmsley, D.; Macy, W.M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017, Montréal, QB, Canada, 15–18 May 2017.
44. Bose, T.; Illina, I.; Fohr, D. Generalisability of Topic Models in Cross-corpora Abusive Language Detection. In Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, Mexico City, Mexico, 6 June 2021; pp. 51–56.
45. Puiu, A.B.; Brabete, A.O. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 75–86.
46. Gao, L.; Huang, R. Detecting Online Hate Speech Using Context Aware Models. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, 2–8 September 2017; pp. 260–266.
47. Gibert, D.O.; Pérez, N.; Pablos, G.A.; Cuadros, M. Hate Speech Dataset from a White Supremacy Forum. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October–1 November 2018; pp. 11–20.

48. Park, H.J.; Fung, P. One-step and Two-step Classification for Abusive Language Detection on Twitter. In Proceedings of the Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 41–45.
49. Mehdad, Y.; Tetreault, R.J. Do Characters Abuse More Than Words? In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Los Angeles, CA, USA, 13–15 September 2016; pp. 299–303.
50. Bugueño, M.; Mendoza, M. Learning to Detect Online Harassment on Twitter with the Transformer. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Würzburg, Germany, 16–20 September 2019; pp. 298–306.
51. Chen, X.; Wu, S.Z.; Hong, M. Understanding Gradient Clipping in Private SGD: A Geometric Perspective. In Proceedings of the NIPS 2020, Virtual, 6–12 December 2020.