mathematics

Article

Keyword-Enhanced Multi-Expert Framework for Hate Speech Detection

Weiyu Zhong [†], Qiaofeng Wu [†], Guojun Lu, Yun Xue and Xiaohui Hu [*]

School of Electronics and Information Engineering, South China Normal University, Foshan 528225, China
[*] Correspondence: huxh@scnu.edu.cn
[†] These authors contributed equally to this work.

Abstract: The proliferation of hate speech on the Internet is harmful to the psychological health of individuals and society. Thus, establishing and supporting the development of hate speech detection and deploying evasion techniques is a vital task. However, existing hate speech detection methods tend to ignore the sentiment features of target sentences and have difficulty identifying some implicit types of hate speech. The performance of hate speech detection can be significantly improved by gathering more sentiment features from various sources. In the use of external sentiment information, the key information of the sentences cannot be ignored. Thus, this paper proposes a keyword-enhanced multiexperts framework. To begin, the multi-expert module of multi-task learning is utilized to share parameters and thereby introduce sentiment information. In addition, the critical features of the sentences are highlighted by contrastive learning. This model focuses on both the key information of the sentence and the external sentiment information. The final experimental results on three public datasets demonstrate the effectiveness of the proposed model.

Keywords: hate speech detection; contrastive learning; multi-task learning

MSC: 18C50

Citation: Zhong, W.; Wu, Q.; Lu, G.; Xue, Y.; Hu, X. Keyword-Enhanced Multi-Expert Framework for Hate Speech Detection. *Mathematics* **2022**, *10*, 4706. https://doi.org/10.3390/math10244706

Academic Editors: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 12 November 2022
Accepted: 8 December 2022
Published: 11 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

With the widespread use of social media and mobile internet platforms, the increasing speed of online speech and the freedom to publish it have led to the malicious prevalence of hate speech. Exposure to such language may cause negative effects on the mental health of victims [1], which may lead to severe social problems. To prevent further negative effects, authorities need to intervene in detecting hate speech online. Thus, the rapid and accurate automatic detection of hate speech has become a popular topic of research in the field of natural language processing. Hate speech detection has gained attention in recent years.

Figure 1 shows an example in which the first sentence contains the hate term *fucking aids* which is an obvious form of offensive hate speech, while the second sentence without obvious hate words or semantics is a positive sentence.

This guy is giving me fucking aids. ➡ Offensive score:0.792

I'm literally doing the same tonight! ➡ Offensive score:-0.625

Figure 1. An example sentence from the Ruddit dataset. The offensive score ranges between -1 (maximally supportive) and 1 (maximally offensive).

An approach to hate speech detection using deep learning has been the focus of most of the research in recent years [2–5]. However, previous research disregarded the sentiment features of target detection sentences and only used pre-trained models or deeper neural networks to obtain semantic features. Wang, C. [6] showed that the semantics of hate speech bear a strong tendency toward negative sentiment. To overcome this problem, recent studies have proposed the use of multi-task learning (MTL), which improves the performance of hate speech detection by using sentiment information [7]. Transfer learning is the process of transferring generalizable knowledge gained from training data to the target task. MTL is a type of transfer learning that involves learning several related tasks simultaneously, allowing these tasks to share information during the learning process, and utilizing the correlation between various tasks to enhance the model's performance and generalization capacity on each task. Kapil, P. [8] proposed a deep MTL framework to exploit useful information from several related classification tasks to perform hate speech detection; this framework uses a hard parameter-sharing approach that is prone to negative transfer. Zhou, X. [9] used multiple feature extraction units to share multi-task parameters so that the model can perform sentiment knowledge sharing. Then, gated networks were used to fuse features for hate speech detection. This model employed a soft parameter sharing method by dividing a single expert into multiple experts, thereby mitigating the negative transfer problem caused by hard parameters.

Although hate speech detection has achieved good performance in recent years, the following problems remain: (1) The latest multi-task framework used in hate speech detection is soft parameter sharing [9], where all experts share all tasks, but the tasks of hate speech detection and sentiment analysis have both positive and negative correlations. Positive correlations are parameter relationships that are beneficial to the fit of the primary task, and conversely, negative correlations are not beneficial. If the negative correlation parameters between tasks are not separated, some noise occurs as part of the tasks, which leads to negative transfer. Moreover, when using multiple experts, the simple gated networks cannot effectively fuse and filter the different information because the experts have abundant information from different tasks. (2) Current work lacks the ability to extract critical information (e.g., keywords) from sentences [5]. It cannot effectively identify different types of hate words, such as profanities, nor identify the association between certain identity terms and offensive statements. Certain identity terms (especially those involving minority groups) appear mainly in texts that are offensive [10], such as the sentence *"This is also the reason that so many of Obama's policies are being overturned/undone, it's just because the Black Guy did them."* has no conspicuous hate words, but rather racial discrimination through the identity term *Black*.

To solve the aforementioned problems, we propose the following approaches. **(1) For the first problem,** we are inspired by the recent progressive layered extraction (PLE) model [11] and gated network research [12]. We divide feature extraction units (e.g., expert modules) into a shared part and task-specific parts. This approach strengthens the independent features of the tasks themselves and better reduces the negative transfer caused by weakly correlated task-sharing parameters. Moreover, we design a feature-filtering gate that can better fuse and filter the information of multiple expert modules. **(2) To solve the second problem,** we propose a solution inspired by a recent contrastive learning model [13]. Our model applies contrastive learning to English hate speech detection by using a swearing dictionary and an identity term dictionary to construct positive and negative examples. This result allows the model to be more sensitive to the critical words so that it can learn the association between various types of hate words or identity term words and offensive statements. In summary, the contributions of our study are as follows:

- To better examine the interaction between hate and sentiment information, we propose an MTL model that is more suitable for hate speech detection, which uses shared experts and task-specific experts to extract features, and finally employs feature-filtering gates to fuse features.
- Given the lack of use of important word information in previous work, we introduce contrastive learning to the pre-trained model to enable our model to better identify keywords in text.
- Experimental results on three baseline datasets demonstrate that our model is effective in hate speech detection.

2. Related Work

Recently, researchers have widely studied automatic hate speech detection. In this section, we review related work on deep-learning-based methods for hate speech detection, especially MTL-based methods, as well as related work on contrastive learning.

Recently, deep-learning-based approaches have achieved considerable success in hate speech detection. Ref. [14] proposed a transformed word embedding model (TWEM), which balances high performance while achieving a simple structure. Ref. [3] proposed a deep neural network structure (combining CNN and GRU) as a feature extractor to learn the semantic features of hate speech. Ref. [4] built a large-scale dataset using hate speech and its reactions and used the pre-trained language model GPT-2 to detect hate speech. Ref. [5] created the first English Reddit comment dataset with fine-grained, real-valued scores and used the pre-trained model HateBERT to detect hate speech. Clearly, deep learning models can extract underlying semantic features of text, which provide the most direct clues to detect hate speech.

Transfer learning can bring more useful information to hate speech detection, and common transfer learning methods include multi-task learning and knowledge distillation [15]. Knowledge distillation aims at knowledge transfer through a wide network (teachers) to a small network (students). Multi-task learning aims at training multiple related tasks and sharing information between tasks at the same time. In recent years, some results have been achieved in the field of hate speech detection using multi-task learning [7]. Ref. [16] proposed a theoretical framework for hate speech type detection that includes fuzzy multi-task learning. Ref. [17] proposed an MTL approach based on the pre-trained model BERT for hate speech detection. Ref. [8] proposed a deep MTL framework to improve the performance of hate speech detection by exploiting useful information from multiple related classification tasks. Ref. [9] proposed a hate speech detection framework based on sentiment knowledge sharing. The preceding studies show that MTL can exploit the relevance between sentiment analysis tasks and hate speech detection tasks, which improves model performance and generalization in hate speech detection.

In addition, some optimization algorithms [18,19] have recently been proposed to obtain better classification results and semantic representations, and contrastive learning is one of them. Contrastive learning aims to learn effective representations by pulling semantically similar sentences together and pushing dissimilar sentences apart [20]. Several recent approaches use contrastive objectives to obtain different views from data augmentation or different copies of the model [21–24]. For example, [24] proposed ConSERT, a Contrastive Framework for Self-Supervised SEntence Representation Transfer, which employs contrastive learning to fine-tune BERT in an unsupervised manner. SimCSE [25] uses the simplest idea of applying only the standard dropout as noise to obtain different outputs of the same sentence, thereby forming positive instances. We propose the use of contrastive learning for hate speech detection, which increases the sensitivity of the model to key information of the sentence and improves the performance of the task.

3. Methodology

In this section, our model keyword-enhanced multi-expert framework for hate speech detection (KMT) is presented. This model exploits critical information of the sentence and external sentiment information to improve hate speech detection.

The general architecture of KMT is shown in Figure 2. The framework consists of four modules: **(1) Textual input module.** The bottom of the figure shows the textual input module, where the pre-trained model BERT or HateBERT is used to encode the input sentences and generate contextually and semantically integrated input vector x; **(2) Multi-task learning module.** The top left of the figure shows the multi-task learning module, where we use the multi-task learning framework to interact sentiment information and hate information, and learn the shared features and task-specific features to assist hate speech detection using sentiment information; **(3) Feature-filtering module.** Gate of the figure is the feature-filtering module, which is used to filter and fuse the features outputted by expert modules to select the important information of sentiment and hate speech; **(4) Contrastive learning module.** The top right of the figure shows the contrastive learning module, which extracts critical information within the sentences to improve the sensitivity of the model to sentence keywords. Finally, the MTL and contrastive learning modules are jointly trained.

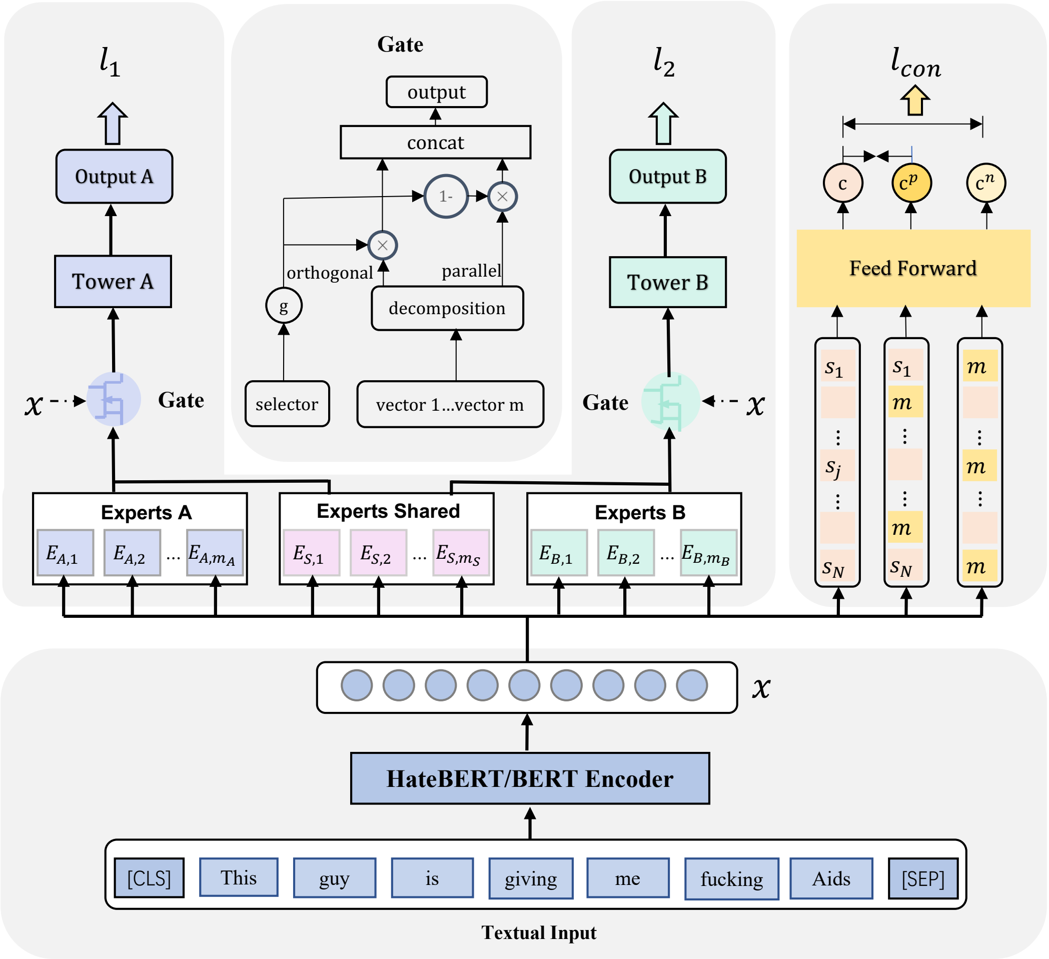

Figure 2. The overall architecture of our proposed Keyword-enhanced Multi-expert Framework for Hate Speech Detection (KMT).

Given the input text $s = \{w_1, w_2, \ldots, w_n\}$, n is the length of the text s. We feed the sequence $[CLS]s[SEP]$ to the BERT or HateBERT encoder in the Textual input module to obtain the input vector x with contextual information. Subsequently, x is taken as input to both multi-task learning module and contrastive learning module. In multi-task learning, the hate information and sentiment information in x are interacted by shared expert and task-specific expert modules, the features are then fused and filtered using a feature-filtering gate, and finally the hate speech detection is performed using the tower containing the classification layer. In the contrastive learning module, positive and negative examples are generated by masking x. Subsequently, the model is enabled to focus on key information in the sentences by bringing x closer to positive examples and away from negative examples. More details of each module are shown as follows.

3.1. Multi-Task Learning Module

Due to the diversity of language, insulting meanings in many sentences are implicit, causing difficulty in determining whether a sentence is offensive or not. For example, the sentence *"These guys are all a bunch of pigs."* does not contain an explicitly hateful word, but the sentence still constitutes hate speech. Although the word *pig* is neutral, most people associate it with foolishness and clumsiness. Thus, likening guys with pigs is demeaning to the former. The secret to effectively judging sentences is grasping emotional common sense. The sentence *"He's a fucking good player."* contains the obvious hate word *fucking*. However, in this case, *fucking* is merely an adverb of level used to indicate excitement; hence, the sentence does not constitute hate speech. From the preceding two examples, we can see that although hate speech tends to contain hate words, achieving better results in detecting it by using only the hate information of the sentence itself is difficult. To introduce external sentiment information, we combine the generic sentiment dataset and then interact the information from the sentiment dataset and the hate dataset using the MTL approach, which improves the performance of hate speech detection.

In MTL frameworks, the problem of overfitting is fundamentally reduced due to extensive use of the shared experts layer structure. However, the effectiveness of the framework may be affected by the seesaw phenomenon and negative migration problem because of the differences between tasks and data distribution [11]. Thus, we use the PLE framework structure [11]. As shown in Figure 2, the model is divided between task-specific tower structures at the top and expert modules at the bottom. The number of Experts in each expert module is the hyperparameter to be tuned. Each expert module comprises numerous sub-networks known as Experts. The shared experts in PLE are responsible for extracting shared features, while the task-specific experts extract task-specific features. Each tower network extracts information from the shared experts and its own task-specific experts. Our expert modules and tower networks consist of feed-forward neural networks. Specifically, when the model performs gradient backpropagation, it changes the parameters in the expert modules. As the output of the task-specific expert modules is only passed to the tower of their own tasks, their parameters are only affected by their own task gradients. By contrast, the shared expert modules have parameters that are affected by the mixed gradients of all tasks because the output is passed to the towers of all tasks.

In the MTL module, features are extracted using the shared experts E_s^T and the task $k's$ specific experts E_k^T. Then, the extracted features are concatenated to form $S^k(x)$ as Equations (1)–(3):

$$E_k^T = \left[E_{(k,1)}^T, E_{(k,2)}^T, \cdots, E_{(k,m_k)}^T\right] \tag{1}$$

$$E_s^T = \left[E_{(s,1)}^T, E_{(s,2)}^T, \cdots, E_{(s,m_s)}^T\right] \tag{2}$$

$$S^k(x) = \left[E_k^T, E_S^T\right]^T \tag{3}$$

where x is the input vector, m_s and m_k are the number of sub-networks in the shared experts E_s^T and task $k's$ specific experts E_k^T, $E_{(k,m_k)}^T$ and $E_{(s,m_s)}^T$ are the sub-networks in task $k's$ specific experts and shared experts, respectively. The features $S^k(x)$ of the shared experts and task $k's$ specific experts are selectively fused through a feature-filtering gate (Gate). The filtered features of task k are formulated as Equation (4):

$$G^k(x) = \text{Gate}\left(x, S^k(x)\right) \tag{4}$$

Lastly, the task k prediction using the tower network is:

$$O^k(x) = f^k\left(G^k(x)\right) \tag{5}$$

where $f^k(\cdot)$ stands for the task $k's$ tower network, which consists of feed-forward neural networks as Equation (5).

3.2. Feature-Filtering Module

The multiple expert setting in MTL enables better interaction of affective and hate information, but because multiple experts have a large amount of information, a structure is needed for selective fusion. Thus, we are inspired by the research on gating modules [12] to design a feature-filtering module that not only better fuses the information between experts but also reduces the noise. As shown in Figure 2, the input vector x is used as a selector to obtain useful information on the selected vector (e.g., the output $S^k(x)$ of the experts) as follows Equations (6)–(9):

$$g^k(x) = W_g^k x \tag{6}$$

$$\text{parallel:} \quad p^k(x) = \frac{S^k(x) \cdot x}{x \cdot x} x \tag{7}$$

$$\text{orthogonal:} \quad o^k(x) = S^k(x) - p^k(x) \tag{8}$$

$$G^k(x) = \text{concat}\Big(g^k(x) o^k(x), \Big(1 - g^k(x)\Big) p^k(x)\Big) \tag{9}$$

where $W_g^k \in R^{(m_k + m_s)d}$ is a parameter matrix, d is the dimension of the input vector, and $g^k(x)$ is the weight vector for task k obtained by a linear transformation. $S^k(x)$ is decomposed into an orthogonal component and a parallel component. The parallel component $p^k(x)$ is a projection of $S^k(x)$ onto x, which contains part of the information of x. On the contrary, $o^k(x)$ is orthogonal to x, and therefore contains new information. Specifically, if x is the hate speech input, $p^k(x)$ is the part of $S^k(x)$ that contains hate speech information, and $o^k(x)$ is the part of $S^k(x)$ that contains sentiment information, then $G^k(x)$ represents the fusion of these two components. $g^k(x)$ is used to regulate the composition of both components to obtain the optimal fusion.

3.3. Contrastive Learning Module

As the pre-trained model lacks the ability to grasp critical word information from sentences, it cannot effectively distinguish between different types of hate words and cannot identify the relationship between certain identity terms and offensive statements. Currently, contrastive learning demonstrates excellent ability in acquiring and distinguishing crucial knowledge by focusing on positive examples and comparing negative examples, which has resulted in considerable advances in many tasks. Our goal is to make our model more sensitive to the essential words within a body of text. To this end, we use a contrastive learning module to focus on the positive examples while pushing the negative ones away, allowing the model to more effectively distinguish between important and minor information. To create a positive example x^p, we mask each non-key token representation in the input vector x as a constant vector $m \in R^d$ where this constant is equal to 1e-6. This method allows the sentence to combine key information and eliminate unimportant words. To obtain the negative example x^n, we simultaneously employ a similar method to mask the key token representation in x as m.

Thereafter, we model x, x^p, and x^n separately using the feed-forward neural networks with the following formulation Equations (10)–(12):

$$c = f(x) \tag{10}$$

$$c^p = f(x^p) \tag{11}$$

$$c^n = f(x^n) \tag{12}$$

where $f(\cdot)$ denotes the feed-forward neural networks. We then compute the cosine similarity of the positive and negative examples as follows Equation (13):

$$\text{sim}\Big(c^1, c^2\Big) = \frac{c_1^T c_2}{\|c_1\| \cdot \|c_2\|} \tag{13}$$

where $\mathrm{sim}(c^1, c^2)$ denotes as $\mathrm{sim}(c, c^p)$ and $\mathrm{sim}(c, c^n)$. We follow the contrast module training objectives developed by [26] as Equation (14):

$$l_{\mathrm{con}} = -\sum_{k=1}^{K}\sum_{i=1}^{N} \log \frac{e^{\frac{\mathrm{sim}\left(c_i, c^p\right)}{\tau}}}{\sum_{j=1}^{N}\left(e^{\frac{\mathrm{sim}\left(c_j, c^p\right)}{\tau}} + e^{\frac{\mathrm{sim}\left(c_j, c^n\right)}{\tau}}\right)} \tag{14}$$

where N is the length of a sentence, K is the batch size, and τ is a temperature hyperparameter that is set to 1 in our model.

3.4. Loss Function

In the training process, we jointly train the objectives of the multi-task learning module and the contrastive learning module. Our training aims to minimize the following total loss functions as Equation (15):

$$\mathrm{loss} = \sum_{i=1}^{n} \lambda_i l_i + \lambda l_{con} \tag{15}$$

where n represents the number of tasks, l_i is the loss function of each task in the MTL module, and λ and λ_i are hyperparameters.

4. Experiments

4.1. Datasets

In our experiments, we employed two sentiment datasets and three public hate speech datasets. Table 1 displays the statistics of the datasets.

Ruddit [5] It is the first English Reddit comment dataset with fine-grained, real-valued scores ranging between -1 (maximum support) and 1 (maximum offense).

OffensEval 2019 (Offen) [27] This dataset was published in the evaluation exercise for SemEval 2019: Task 6. The dataset contains a total of 14,100 tweets. It is divided into a training set with 13,240 tweets and a test set with 860 tweets. There are 4400 tweets marked as offensive in the training and 240 in the test.

AbusEval (Abuse) [28] To obtain this dataset, the researchers added a layer of abusive language annotation to OffensEval 2019. The dataset is the same size as OffensEval 2019, as well as being divided into a training set of 13,240 texts and a test set of 860 texts.

Reddit Sentiment Analysis (RSA) (https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset [November 2022]) This dataset was produced as a result of a university study using PySpark to conduct sentiment analysis across multiple social media networks. The dataset also includes a sentimental label and approximately 37,000 comments. Since this dataset is an auxiliary dataset for training the multi-task learning module, we only use the training set.

Tweet Sentiment Analysis (TSA) (https://www.kaggle.com/datasets/dv1453/twitter-sentiment-analysis-analytics-vidya [November 2022]) This is a tweet sentiment dataset from Kaggle 2018. This dataset contains more positive tweets and less negative tweets. This dataset also uses only the training set.

We used Pearson correlation (Pear) and mean square error (MSE) as evaluation metrics for the Ruddit dataset and Macro F1 (F1) as evaluation metrics for the Offen and Abuse datasets.

Table 1. Statistics of three experimental datasets.

Dataset	Total	Classes
Ruddit	5828	Score 0–1 (2514) Score −1–0 (3442)
Offen	14,100	hate (4640) non-hate (9460)
Abuse	14,100	exp-hate (2129) imp-hate (798) non-hate (11,173)
RSA	37,249	neutral (13,142) negative (8277) positive (15,830)
TSA	31,962	negative (2242) positive (29,720)

4.2. Training Details

We use the five-fold cross-validation approach to evaluate the performance of our model on all three datasets. Referring to [5], we separated the original dataset into five equal parts, using one copy for testing and used the remaining data for training. To prevent the problem of data imbalance in multi-task learning, we use the WeightedRandomSampler approach to sample the data according to the weights. In our experiments, in the MTL module, the number of subnetworks in share expert is 2, and the number of sub-networks in the task-specific expert is also 2. Each expert has one layer of dropout, which is 0.1. The dropout used in the tower network is also 0.1. For the contrastive learning module, the temperature parameter τ is set to 1. The optimizer is Adam, the learning rate is 2×10^{-5}, and the batch size is 16.

4.3. Comparison with Baselines

We compare our model (KMT) with a number of reliable baselines. The following is a brief description of the models:

BERT [29] This pre-trained model is mainly used to capture sentence features for the detection of hate speech.

HateBERT [30] It is a BERT variant that has been specially trained to recognize hate speech in English. The big dataset RAL-E, which contains Reddit comments from communities that have been banned because of their hateful or offensive speech, was used to train HateBERT. In the three popular datasets OffensEval 2019 [27], AbusEval [28], and HatEval [31], HateBERT significantly outperforms the BERT model.

KMT It is our proposed hate speech detection model based on sentence critical information and external sentiment information.

The comparison of the entire performance of KMT is shown in Table 2. From the results in this table, the following conclusions can be drawn:

(1) The performance of HateBERT is much better than that of BERT in the three datasets. In particular, the performance is significantly improved on the Abuse dataset, which indicates that HateBERT can better capture the semantic relationships between words in hate speech and better perform hate speech detection.

(2) Our proposed model KMT obtained good performance on all three datasets. Compared with the current best performing model, the Pearson correlation of KMT increases by 0.006 on the Ruddit dataset, the F1 value of KMT improves greatly by nearly 0.028 on the Abuse dataset. These results illustrate the effectiveness of our method.

Table 2. Comparative results of KMT and existing methods. Superscript * indicates data obtained from the literature. The best results for each model are shown in boldface.

Models	Ruddit (Regression)		Abuse (3 Class)	Offen (2 Class)
	Pear ↑	**MSE ↓**	**F1 ↑**	**F1 ↑**
BERT * [5,30]	0.873 ± 0.005	0.027 ± 0.001	0.727 ± 0.008	0.803 ± 0.006
HateBERT * [5,30]	0.886 ± 0.005	0.025 ± 0.001	0.765 ± 0.006	**0.809 ± 0.008**
KMT (BERT)	0.8764 ± 0.007	0.027 ± 0.0007	0.7882 ± 0.01	0.8028 ± 0.02
KMT (HateBERT)	**0.8921 ± 0.006**	**0.0231 ± 0.001**	**0.7929 ± 0.01**	0.8064 ± 0.01

4.4. Ablation Experiments

We analyze the effect of different modules on the performance of our model. The results are shown in Table 3, where *w/o cl* indicates the ablation experiment for contrastive learning; *w/o s* indicates that the MTL module is removed and the sentiment dataset is not used as input to the model; and *w/o gate* indicates that the feature-filtering gate module is replaced with simple feed-forward neural network and a softmax layer.

According to the results in Table 3, we find that:

(1) When the contrastive learning module is removed, the performance of the model on the two datasets decreases the most, indicating that the swear words and certain identity terms in the sentences are highly correlated with hate speech. The results show that the contrastive learning module can improve the sensitivity of the model to keywords and thus improve the performance of hate detection effectively.

(2) When the MTL module is removed, the performance of the model on the three datasets also decreases, indicating that adding sentiment information can effectively assist the detection of hate speech.

(3) When the feature-filtering module is replaced with the basic gating network, the performance also decreases slightly, indicating that our proposed feature-filtering gates can better achieve the fusion of various expert information and reduce the influence of noise.

(4) KMT outperforms other models, which directly demonstrates the importance and effectiveness of sentence critical information and external sentiment information for hate speech detection.

Table 3. Results of ablation experiments. The best results for each model are shown in boldface.

Models	Ruddit (Regression)		Abuse (3 Class)	Offen (2 Class)
	Pear ↑	**MSE ↓**	**F1 ↑**	**F1 ↑**
KMT *w/o cl*	0.8879 ± 0.005	0.0246 ± 0.0005	0.7827 ± 0.02	0.7995 ± 0.024
KMT *w/o s*	0.8907 ± 0.004	0.0234 ± 0.0008	0.7846 ± 0.02	0.7957 ± 0.019
KMT *w/o gate*	0.8892 ± 0.004	0.0249 ± 0.001	0.7886 ± 0.02	0.8035 ± 0.02
KMT	**0.8921 ± 0.006**	**0.0231 ± 0.001**	**0.7929 ± 0.01**	**0.8064 ± 0.01**

4.5. Effect of Number of Experts

Each expert module in the multi-task module consists of multiple sub-networks called Experts. To investigate the effect of the number of respective Experts (e.g., E_s^T and E_k^T) in the shared expert module and task-specific expert module on the performance, we use 1 to 4 Experts on the Ruddit dataset to evaluate our model. As shown in Figure 3, the model performs best when the shared expert module has two Experts and the task-specific expert module has two Experts, which justifies the number of experts we choose in the experimental setup. In addition, the performance of the model is worse when the number of Experts in the shared expert module is three or four. This result indicates that having a larger number of parameters does not improve the performance of the model because too many parameters may cause the model to be more difficult to train and an extremely large number of Experts may cause redundant information.

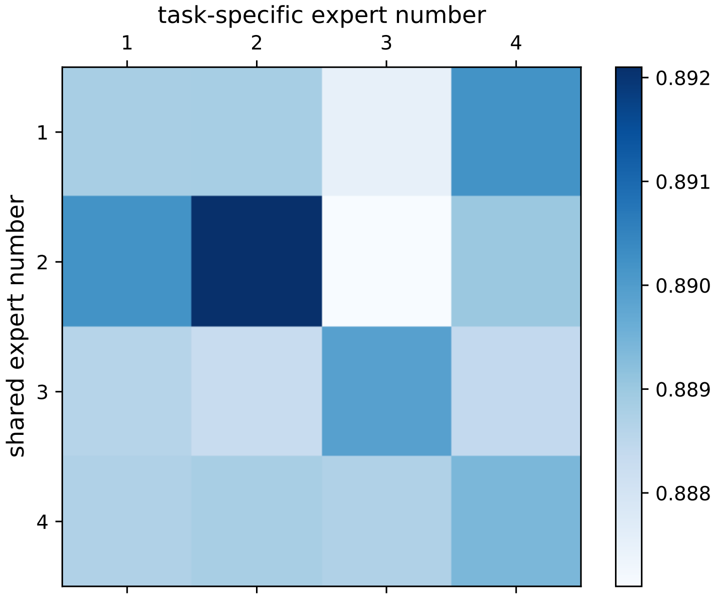

Figure 3. Pear mean value of model with different number of Experts, where the darker color indicates a higher Pear value.

4.6. Effect of Extraction Network Layer Number

Extraction networks are in the multi-tasking module, and each network consists of the expert modules and the feature-filtering module (Gate) in Figure 2, which is mainly used to extract features. To investigate the effect of the number of extraction network layers on performance, we test the effects of one-layer and two-layer extraction networks on our model on the Ruddit dataset. According to experience, the number of training parameters increases with the depth of the network structure. As the results shown in Table 4, the model performs better when the extraction network is one layer. As the depth of the extraction network increases, the model performance decreases because when the model is highly complex, it causes overfitting that the model becomes unstable. Furthermore, we also compare the overall running time of the two models, performed at the 3090 GPU setting, as shown in Figure 4. The results illustrate that the overall performance of the model is improved when the one-layer extraction network is used, besides, the number of parameters is also reduced due to the reduction in the number of network layers, which improves the efficiency of the model.

Table 4. Effect of number of extraction network layers

Models	**Ruddit (Regression)**	
	Pear ↑	**MSE ↓**
1 layer	0.8921 ± 0.006	0.0231 ± 0.001
2 layers	0.8731 ± 0.007	0.0283 ± 0.001

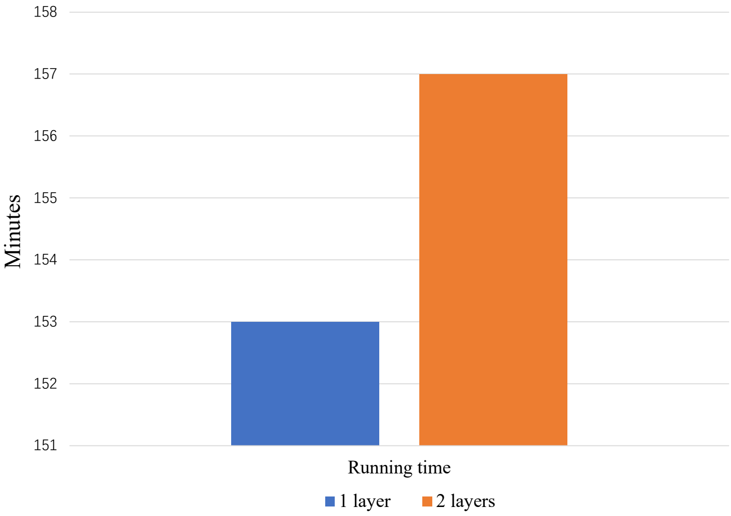

Figure 4. Runtime comparison.

5. Conclusions and Future Work

In this work, we propose a keyword-enhanced multi-expert framework for hate speech detection. This model can leverage both the external sentiment information and critical information of the sentence itself. Moreover, this model mainly uses a shared expert module to share certain parameters of multiple tasks. Through this approach, the model can more effectively share sentiment information and then fuse features by employing a feature-filtering gate to detect hate speech. We use contrastive learning for keyword enhancement, which enables the model to better identify critical information in sentences. Experiments show that our model, keyword-enhanced multi-expert framework, performs better on three datasets. Finally, detailed analysis further demonstrates the effectiveness of our model and the contribution of each module. In future work, we will explore the portability and generalization of the model and conduct portability experiments across datasets. Meanwhile, based on this work, we consider adding image information for multimodal hate detection.

Author Contributions: Conceptualization, W.Z. and G.L.; methodology, W.Z.; formal analysis, W.Z. and Q.W.; writing—original draft preparation, W.Z. and Q.W.; writing—review and editing, Y.X. and X.H.; supervision, Y.X. and X.H.; funding acquisition X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Characteristic Innovation Projects of Guangdong Colleges and Universities (Nos. 2018KTSCX049), and the Science and Technology Plan Project of Guangzhou under Grant Nos. 202102080258 and 201903010013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Munro, E.R. *The Protection of Children Online: A Brief Scoping Review to Identify Vulnerable Groups*; Childhood Wellbeing Research Centre: London, UK, 2011.
2. Jahan, M.S.; Oussalah, M. A systematic review of hate speech automatic detection using natural language processing. *arXiv* **2021**, arXiv:2106.00742.
3. Zhang, Z.; Luo, L. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semant. Web.* **2019**, *10*, 925–945. [CrossRef]
4. Tekiroglu, S.S.; Chung, Y.L.; Guerini, M. Generating counter narratives against online hate speech: Data and strategies. *arXiv* **2020**, arXiv:2004.04216.

5. Hada, R.; Sudhir, S.; Mishra, P.; Yannakoudakis, H.; Mohammad, S.M.; Shutova, E. Ruddit: Norms of offensiveness for English Reddit comments. *arXiv* **2021**, arXiv:2106.05664.
6. Wang, C. Interpreting neural network hate speech classifiers. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October 2018; pp. 86–92.
7. Chiril, P.; Pamungkas, E.W.; Benamara, F.; Moriceau, V.; Patti, V. Emotionally informed hate speech detection: A multi-target perspective. *Cogn. Comput.* **2022**, *14*, 322–352. [CrossRef] [PubMed]
8. Kapil, P.; Ekbal, A. A deep neural network based multi-task learning approach to hate speech detection. *Knowl. Based Syst.* **2020**, *210*, 106458. [CrossRef]
9. Zhou, X.; Yong, Y.; Fan, X.; Ren, G.; Song, Y.; Diao, Y.; Yang, L.; Lin, H. Hate speech detection based on sentiment knowledge sharing. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021; pp. 7158–7166.
10. Sap, M.; Card, D.; Gabriel, S.; Choi, Y.; Smith, N.A. The risk of racial bias in hate speech detection. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1668–1678.
11. Tang, H.; Liu, J.; Zhao, M.; Gong, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, New York, NY, USA, 22 September 2020; pp. 269–278.
12. Lai, T.; Ji, H.; Bui, T.; Tran, Q.H.; Dernoncourt, F.; Chang, W. A context-dependent gated module for incorporating symbolic semantics into event coreference resolution. *arXiv* **2021**, arXiv:2104.01697.
13. Hu, J.; Li, Z.; Chen, Z.; Li, Z.; Wan, X.; Chang, T.H. Graph Enhanced Contrastive Learning for Radiology Findings Summarization. *arXiv* **2022**, arXiv:2204.00203.
14. Kshirsagar, R.; Cukuvac, T.; McKeown, K.; McGregor, S. Predictive embeddings for hate speech detection on twitter. *arXiv* **2018**, arXiv:1809.10644.
15. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vision* **2021**, *129*, 1789–1819. [CrossRef]
16. Liu, H.; Burnap, P.; Alorainy, W.; Williams, M.L. Fuzzy multi-task learning for hate speech type identification. In Proceedings of the The World Wide Web Conference, New York, NY, United States, 13 May 2019; pp. 3006–3012.
17. Ousidhoum, N.; Lin, Z.; Zhang, H.; Song, Y.; Yeung, D.Y. Multilingual and multi-aspect hate speech analysis. *arXiv* **2019**, arXiv:1908.11049.
18. Gou, J.; He, X.; Lu, J.; Ma, H.; Ou, W.; Yuan, Y. A class-specific mean vector-based weighted competitive and collaborative representation method for classification. *Neural Networks.* **2022**, *150*, 12–27. [CrossRef]
19. Gou, J.; Yuan, X.; Du, L.; Xia, S.; Yi, Z. Hierarchical Graph Augmented Deep Collaborative Dictionary Learning for Classification. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25308–25322. [CrossRef]
20. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway Township, NJ, USA, 2006; Volume 2, pp. 1735–1742.
21. Meng, Y.; Xiong, C.; Bajaj, P.; Bennett, P.; Han, J.; Song, X. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23102–23114.
22. Janson, S.; Gogoulou, E.; Ylipää, E.; Cuba Gyllensten, A.; Sahlgren, M. Semantic re-tuning with contrastive tension. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 4 May 2021.
23. Kim, T.; Yoo, K.M.; Lee, S.G. Self-guided contrastive learning for BERT sentence representations. *arXiv* **2021**, arXiv:2106.07345.
24. Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; Xu, W. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv* **2021**, arXiv:2105.11741.
25. Gao, T.; Yao, X.; Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv* **2021**, arXiv:2104.08821.
26. Robinson, J.; Chuang, C.Y.; Sra, S.; Jegelka, S. Contrastive learning with hard negative samples. *arXiv* **2020**, arXiv:2010.04592.
27. 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.
28. Caselli, T.; Basile, V.; Mitrović, J.; Kartoziya, I.; Granitzer, M. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 6193–6202.
29. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
30. Caselli, T.; Basile, V.; Mitrović, J.; Granitzer, M. Hatebert: Retraining bert for abusive language detection in english. *arXiv* **2020**, arXiv:2010.12472.
31. Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Pardo, F.M.R.; 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.