applied system innovation

MDPI

Article

Offensive-Language Detection on Multi-Semantic Fusion Based on Data Augmentation

Junjie Liu [1,†], Yong Yang [1,†], Xiaochao Fan [1,†], Ge Ren [1,*], Liang Yang [2] and Qian Ning [3,4,*]

[1] School of Computer Science and Technology, Xinjiang Normal University, Urumqi 830000, China; junjiemail2021@163.com (J.L.); yangyong1900@163.com (Y.Y.); fxc_edu@163.com (X.F.)
[2] School of Computer Science and Technology, Dalian University of Technology, Dalian 116000, China; liang@dlut.edu.cn
[3] School of Physics and Electronic Engineering, Xinjiang Normal University, Urumqi 830000, China
[4] College of Electronics and Information Engineering, Sichuan University, Chengdu 610000, China
* Correspondence: ren_xjnu@163.com (G.R.); ningq@scu.edu.cn (Q.N.)
† These authors contributed equally to this work.

Abstract: The rapid identification of offensive language in social media is of great significance for preventing viral spread and reducing the spread of malicious information, such as cyberbullying and content related to self-harm. In existing research, the public datasets of offensive language are small; the label quality is uneven; and the performance of the pre-trained models is not satisfactory. To overcome these problems, we proposed a multi-semantic fusion model based on data augmentation (MSF). Data augmentation was carried out by back translation so that it reduced the impact of too-small datasets on performance. At the same time, we used a novel fusion mechanism that combines word-level semantic features and n-grams character features. The experimental results on the two datasets showed that the model proposed in this study can effectively extract the semantic information of offensive language and achieve state-of-the-art performance on both datasets.

Keywords: offensive language; data augmentation; MSF

Citation: Liu, J.; Yang, Y.; Fan, X.; Yang, L.; Ren, G.; Ning, Q. Offensive-Language Detection on Multi-Semantic Fusion Based on Data Augmentation. *Appl. Syst. Innov.* **2022**, *5*, 9. https://doi.org/10.3390/asi5010009

Academic Editor: Andrey Chernov

Received: 30 November 2021
Accepted: 31 December 2021
Published: 4 January 2022

1. Introduction

Offensive language is usually defined as hurtful, derogatory, or obscene comments made by one person to another. These kinds of remarks are not uncommon in social media. Cyberbullying is a serious social problem. How to quickly and accurately automatically detect offensive language in social media has become one of the hot research issues in the field of natural language processing. The rapid identification of offensive language on the Internet and prevention of its viral spread has important practical significance for reducing cyberbullying and self-harm behavior.

Offensive-language detection tasks are usually regarded as supervised text-classification tasks. Offensive-language remarks are intrinsically closely related to the relationships between groups and very much depend on the nuance of language. In many cases, despite the manual methods being used to distinguish whether a sentence constitutes offensive language, the consensus level is very low [1]. In recent years, offensive-language remarks have become more harmful. Therefore, the issue has attracted more and more attention from researchers, and many related datasets and related algorithms have emerged. However, most of the datasets are small in size and low in quality. The algorithms only use artificial features and simple fine-tuning pre-trained models.

Zampie et al. [2] constructed the Offensive Language Identification Dataset (OLID). A sentence in OLID is annotated by multiple annotators. When annotating conflicts, a majority vote is also required. However, this requires a significant amount of labor and material resources; so, OLID only has 14,100 sentences. The GermEval 2018 [3] shared task

focused on offensive-language recognition in German tweets. Its dataset only contains 8500 sentences.

In recent years, traditional deep-learning methods have also been widely used in offensive-language detection. The deep-learning methods are based on the word-embedding representation obtained from large-scale expected training and use a neural network structure to extract and merge semantic features. Gambäck and Sikdar [4] used a convolutional neural network (CNN) to detect offensive language. In addition to the CNN method, Badjatiya et al. [5] also used a Long Short-Term Memory (LSTM) network and FastText [6] methods to detect offensive language. These methods' performance is better than the traditional machine-learning methods. However, traditional deep learning uses static word embedding, which cannot solve the polysemy problem. It generates a significant reduction in the performance of the classifier.

Recently, self-supervised learning to obtain pre-trained models unrelated to specific tasks from large-scale data has been a great success in the field of natural language processing. A pre-trained language model can be defined as a black box that has prior knowledge of natural language and can be applied and fine-tuned to solve various NLP problems. The pre-trained models of Generative Pre-Training (GPT) [7] and Bidirectional Encoder Representations from Transformers (BERT) [8] have achieved the best performance in many natural-language-processing tasks. Although the contextualized word embedding of the pre-trained models solves the word-ambiguity problem, it ignores character-level features, which reduces the performance of offensive-speech detection models.

To solve the above problems, we proposed a multi-semantic fusion network based on data augmentation (MSF) (code is available at https://github.com/RoversCode/OffensiveDetectionMSF, accessed on 2 January 2022). Our work is based on intuition: (1) The same semantics can be expressed in different ways, and the corresponding sentence structure may also be different. (2) Pre-trained language models such as BERT can obtain dynamic word vectors and have a stronger ability to represent offensive language sentences. In addition, character-level features are a significant linguistic feature of offensive language. Based on the above two points, we adopted back translation to achieve data augmentation and proposed the MSF model. The proposed model captures features by BERT and combines with CNN to build the information of n-grams character features. Then, we used an interactive fusion mechanism to fuse two kinds of information in the task of offensive-language detection.

To summarize, our contributions are as follows:

(1) For small-scale offensive-language detection tasks, we adopted the back-translation method to enhance the data so that the model can obtain richer semantic information.

(2) We proposed the character-capture module to capture the n-grams character features and to combine deep semantic features. Then, we utilized an interactive fusion mechanism to combine them. The experimental results show that it is useful for offensive-language detection.

(3) The experimental results on the two public datasets demonstrate that our method achieved state-of-the-art performance compared with strong baselines.

The rest of our article is structured as follows. Section 2 discusses related work on datasets, data augmentation, and offensive-language detection methods. Section 3 gives a detailed description of our MSF model for offensive-language detection. Section 4 shows the extensive experiments to evaluate the effectiveness of our model, and Section 5 summarizes the work and outlines the future direction.

2. Related Work

In this section, we review related works on datasets, data augmentation, deep-learning-based methods, and pre-trained models for offensive-language detection.

There are relatively few datasets in the field of offensive-language detection. Labeling methods can be divided into manual labeling and semi-supervised automatic labeling. Sigurbergsson et al. [9] created the offensive-language dataset DKHATE in Danish with only 3600 sentences. Pitenis et al. [10] constructed the Offensive Greek Tweet Dataset (OGTD),

which contains 10,287 sentences. Compared with other languages, the scale of the offensive-language datasets for English is larger, such as OLID [2] with 13,240 sentences and the Davidson dataset [11] containing more than 20,000 sentences. Rosenthal et al. [12] annotated data in a semi-supervised manner and constructed the SOLID dataset. Although SOLID has 9 million pieces of data, the dataset is much noisier, and the quality is lower. In summary, the manually labeled datasets are small in size and cannot meet the needs of models, and the automatically labeled datasets are large in scale, but the quality cannot be guaranteed.

Data augmentation is a technique for automatically expanding training data. The great success of deep learning is supported by a large amount of data. Wei et al. [13] replaced some words in the sentence with their synonyms to make the augmented data fit the original semantics as much as possible. Luque et al. [14] and Zhang et al. [15] transformed the original document into text in other languages through translation and then translated it back to get the new text in the original language to achieve the purpose of data expansion. In addition to the above methods, there is also word vector-based replacement [16], simple pattern-matching changes at the word level, and so on [17]. Previous studies have shown that data augmentation can effectively improve the performance and robustness of the models. Our intuition is that introducing data-augmentation methods into offensive-language-detection tasks can effectively improve model performance.

Traditional deep-learning methods filter and extract the semantic information of sentences by constructing different neural-network structures to obtain features with strong representation and use them for offensive-language-detection tasks. Zhang et al. [18] proposed a hybrid CNN-LSTM method for offensive-language-detection tasks. The experimental results show that the performance of this method is not much better than when using static word vectors. Gambäck and Sikdar [4] used CNN to extract four-gram character-level features for offensive-language detection, and character-level features can effectively improve the performance of offensive-language detection. Shen et al. [19] used maximum pooling and average pooling to fuse features of different dimensions after the word-embedding layer. It was found that the fusion strategy is significantly better than the single pooling strategy. In conclusion, the static word vector is not conducive to detecting offensive language and the combination of multiple pooling helps to improve the performance of models. More importantly, the character-level features can effectively improve the performance of offensive-language-detection tasks.

The transformer-based pre-training model has achieved good performance in natural-language-understanding tasks. Representative models include ELMO [20], GPT, and BERT. Alatawi et al. [21] fine-tuned BERT to detect offensive language. Sohn and Lee [22] proposed multi-channel BERT, and each BERT channel corresponds to a pre-trained BERT in different languages. Sohn believes that offensive language expressed in different languages has a similar semantic representation and syntactic structure; so, he translated the dataset into multiple languages and used multi-channel BERT to extract features. Lou et al. [23] used a hybrid BERT-GCN method to recognize offensive language. Compared with static word vectors, the performance of pre-trained language models is better, but they often ignore character-level features, and they are likely to play an important role in offensive language.

A comparison of the above method and MSF shows Table 1.

Table 1. Comparison of offensive speech-detection methods

Methods	Data Augmentation	Character Features	Dynamic Embedding
TDLM	×	✓	×
TPTM	×	×	✓
MSF	✓	✓	✓

Note: TDLM: traditional deep-learning methods; TPTM: transformer-based pre-traing model; and MSF is the method we proposed in Section 1.

3. Methodology

In this section, we introduce our proposed MSF model in detail.

The overall structure of the MSF model is shown in Figure 1. MSF can be divided into three levels: (1) data augmentation; (2) semantic understanding; and (3) the interaction fusion mechanism. So, we give a detailed summary of each part. In order to expand the dataset, we used back translation to double the size of the dataset. The semantic understanding can be divided into two parts: the deep semantic module and the character-capture module. For the deep semantic part, we directly used BERT. In the character-capture part, we used CNN to extract semantic-information-carrying character features. To better integrate deep semantic features and character-level features, we adopted a features-interaction fusion mechanism. Next, we introduce each part in detail.

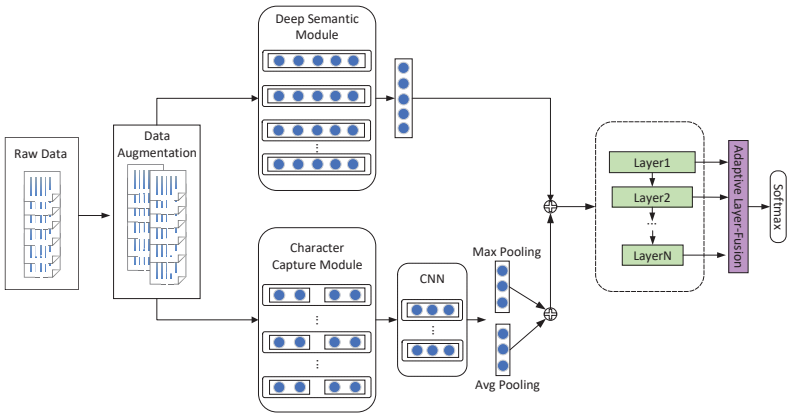

Figure 1. The overall framework of our proposed based on data augmentation multi-semantic fusion neural network model (MSF).

3.1. Data Augmentation

We used the back-translation method for data augmentation. Back translation [14] does not directly replace individual words one by one based on the relationship of synonyms but retells sentences in a generated way to achieve the purpose of data expansion. For example:

exp1 original: Fuckkk I know I seen bitch that's a slapper lmao.
translation: Shit, I know I've seen a bitch. That's a slut.
exp2 original: fuck no bitch we decided we both was gone steepppp.
translation: No fucking bitch, we decided to leave quickly.

As can be seen from the above two examples, firstly, compared with other data-augmentation methods, back translation can use existing translation tools to generate enhanced data, and it ensures that the sentence is grammatically correct; the semantics are fluent and do not deviate from the original sentence. In this way, the model can learn the correct semantic and grammatical features. Secondly, after the sentence is translated, the structure of some sentences changes slightly, such as exp1; some sentences have large changes in patterns, such as exp2. Although the sentence pattern has changed, its semantics remain unchanged, which enables the model to learn richer semantic features. Lastly, most text on social media is in colloquial form, so the words are more casual. When the users make offensive remarks, they might repeat certain characters or word forms to express the urgency of things or the intensity of emotions. These misspelled words have a high probability of being treated as unregistered words. With augmented data generated by back translation, on the one hand, misspelled words can be corrected, such as *Fuckkk* in exp1 and *steepppp* in exp2; on the other hand, some implicit dirty words are made explicit—for example, *slapper-slut* in exp1.

There is a significant amount of noise in the sentence, which will affect the quality of the back translation; so, we first carried out data preprocessing. In the preprocessing process, we deleted useless symbols, such as *url*, @, #, and *user* and converted emoticons

into text content. After this processing, we finally used the Google Translate API to achieve data augmentation.

3.2. Semantic Understanding

Due to the language characteristics of offensive language, semantic understanding is divided into the deep semantic module (DSM) and the character capture module (CCM).

3.2.1. Deep Semantics Module

This module is a pre-trained BERT based on the encoder network structure of the transformer [24]. BERT implements contextualized word embedding, which solves the problem of ambiguity in offensive language. At the same time, BERT continuously extracts the high-level vector representation of the current input through the multi-head attention mechanism and the feedforward network. The formula is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where the input vector X is mapped into matrix Q, K, V by the parameter matrix W_Q, W_K, W_V. d_k is the number of dimensions of K. The final output representation of the multi-head attention mechanism is the concatenation of the output of each attention:

$$\begin{aligned} MultiHead &= concat(M_1, M_2, ..., M_A)W_O \\ M_i &= Attention(Q, K, V) \end{aligned} \tag{2}$$

3.2.2. Character-Capture Module

Character-level n-grams are more predictive than word-level features in offensive language [25]. These misspelled words are often key words in offensive language. To better fine-tune the word-vector representation, we introduced n-grams character features and used a neural network model CNN that can better extract local features to extract sentence information that carries n-grams characters. The character-capture model consists of an embedding layer, a convolutional layer, and a pooling layer. The specific implementation details of each layer are described in detail below.

Embedding Layer. In this layer, words can be expressed as a combination of word vectors and n-grams character vectors. Let the sentence $S = \{w_1, w_2, \ldots, w_N\}$, $w_i \in \mathbb{R}^d$, and we convert it into $w_i = \{c_1, c_2, ..., c_m\}$, $w_i \in \mathbb{R}^{d'}$; c_i is the character of a word. Where d and d' are the dimensional vectors, and N is the length of the sentence. The n-grams character can be represented as $G = \{c_1c_2..c_n, c_2c_3..c_{n+1}, ..., c_sc_{s+1}c_m\}$, $s \geqslant m - n$. The final word can be expressed as $w_i^{'} = w_i \oplus G$, which carries both word semantic and n-grams character features. Taking the word *like* and *n-grams* $= 2$ as an example, it can be represented *like={like,li,ik,ke}*. For the character embedding, we randomly initiate.

Convolutional Layer. A convolution strategy can better extract local semantic features. The convolution layer uses a convolution kernel with a window size of h to extract local features. It is calculated as follows:

$$C = f(wg_{i:i+h-1} + b) \tag{3}$$

where $g_{i:i+h-1}$ represents the $i - th$ to $i + h - 1$ columns of the input vector g, with h as the convolution kernel size. f is the nonlinear activation function ELU. w is the parameter matrix, and b is the bias term.

Pooling Layer. The maximum pooling strategy was used to obtain key information features. Average pooling supplements the semantic information ignored by maximum

pooling. We used max pooling and average pooling to extract more comprehensive semantic information:

$$P_{max} = Max(C) \tag{4}$$

$$P_{avg} = Avg(C) \tag{5}$$

$$P = P_{max} \oplus P_{avg} \tag{6}$$

where C is the output of the convolutional, layer and $\oplus$ represents concatenation.

3.3. Interactive Fusion Mechanism

Because the simple fully connected layer connection does not take into account the interaction between potential features, after getting the output of the sentence by DSM and CCM to represent V_{DSM} and V_{CCM}, we used an interaction mechanism to interact between the two vectors instead of simply connecting them. In detail, this interaction mechanism uses two strategies to learn sentence representation.

Pyramid Layers. He et al. [26] proposed that the model can use a small number of hidden units as a higher-level model to learn more abstract features. Chen et al. [27], inspired by this, proposed the pyramid structure. It better integrates features. The pyramid layer is composed of N MLPs where the bottom layer is the widest, and each successive layer has a smaller number of neurons. The output of the MLP of each layer is not only the input of the MLP of the next layer but also the input of the adaptive layer.

In this study, we let the sentence vector $v_i^n \in \mathbb{R}^{\bar{d}\cdot(\frac{1}{2})^{n-1}}$ of the *n-th* layer be defined as:

$$v_i^n = ELU(W^n v_i^{n-1}) \tag{7}$$

where $v_i^1 = concat(v_i^{DSM}, v_i^{CCM})$, and $\bar{d} = dimension(V^{DSM}) + dimension(V^{CCM})$. W_n are parameters. $n \in [1, N]$ denotes the layer index.

Adaptive Layers Fusion. The output of the MLP of each layer of the pyramid layer is connected in series to be the input of the adaptive layer. The formula is as follows:

$$v_i^n = ELU(W^n v_i^{n-1}) \tag{8}$$

where W_r are parameters. $\alpha = [\alpha_1, ..., \alpha_N]$ is a normalized weight learned during training.

3.4. Model Training

We used the cross-entropy loss function to train our model in an end-to-end manner. The objective of learning θ is to minimize the loss function as follows:

$$loss = -\sum_i \sum_j y_i^j log\hat{y}_i^j + \lambda \|\theta\|^2 \tag{9}$$

where i is the index of sentences; j is the index of class; λ is the $L2$ regularization parameter; and θ denotes all trainable parameters.

4. Experimentation

In this section, we first introduce the datasets, evaluation metrics, and implementation details. Then, we compare our model with several strong baselines. Finally, a detailed analysis is presented.

4.1. Experimental Settings

Data Settings. To prove the validity of the model, we conducted experiments on two public offensive-language datasets. On the two datasets, we only performed data augmentation on the training set. The amount of enhanced data was twice that of the original training set. The details of the datasets are shown in Table 2.

Table 2. Statistics of datasets.

Dataset	Total	Classes
OLID	14,100	OFF (33.23%) NOT (66.77%)
Davidson Dataset	24,783	Hate (5.77%) Offensive (77.43%) Neither (16.80%)

The Offensive Language Identification Dataset (OLID) released by SemEval-2019 Task 6 has three subtasks. We only focused on subtask A: Offensive-Language Detection. The test set had 820 sentences, and the training set had 13,420 sentences. OLID is an unbalanced dataset, and the ratio of positive samples to negative samples is about 1:2.

The Davidson Dataset (DV) was created by [11]. The DV dataset contains more than 20,000 sentences, which is relatively large. DV contains three categories, including hate speech, offensive language, and neither. It is also an unbalanced dataset, with the ratio of the three categories of 1:13:3, respectively.

Data imbalance is fairly common in ecology, and it usually reflects an unequal distribution of classes within a dataset. Models built on imbalanced datasets will be constrained by its ability to predict rare and minority points. On the other hand, the predictive capabilities of the model are better by balancing the data compared with imbalanced data [28]. Therefore, whether to train on a balanced dataset does not affect the experimental results.

Evaluation Metrics. To facilitate comparison with the baseline methods, we used different evaluation metrics. To be consistent with the SemEval competition, we used standard Acc and macro-F1. For the DV dataset, we used the five-fold cross-validation mechanism and used Acc and weighted F1. The formulas are as followings.

$$macro - F1 = \frac{F_p + F_n}{2} \tag{10}$$

$$Weighted - F1 = \frac{F_p * T_p + F_n * T_n}{T_p + T_n} \tag{11}$$

where F_p and F_n represent positive F1 and negative F1, respectively. T_p is the total number of positive samples. T_n is the negative samples.

Implementation Details. In our experiment, we augmented the dataset through the Google Translate API. The input sentence length was 128. For the CCM module, we randomly initialized the n-grams vector, and its dimension was 300. Then, we used the CNN with two layers of convolution kernels: 3 and 5, respectively. For the DSM module, we used BERTbase ($L = 12$, $H = 768$, $A = 12$, and $total parameters = 110$ M) where L is the number of transformer blocks; H is the hidden size; and A is the number of heads. The pyramid layer has three layers with 128, 64, and 32 hidden units. The adaptive fusion layer has 128 hidden units. There is a dropout layer after the adaptive fusion layer, and the dropout rate was 0.2. Finally, MSF was optimized by the Nadam optimizer where $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate was 10^{-4}. The models were trained by a mini-batch of 16 instances. To prevent overfitting, we used an early stop in the training process.

Table 3. The performance of the proposed method MSF and other baseline methods are compared.

Approaches	Davidson Dataset		OLID	
	Acc	F1 (weighted)	Acc	F1 (macro)
SVM *	-	0.870	-	0.690
CNN *	0.903	0.892	0.710	0.630
BiGRU *	0.907	0.901	0.780	0.740
BiLSTM *	0.895	0.888	0.780	0.730
BiGRU-CNN *	0.914	0.892	0.710	0.630
BiLSTM-Att *	0.895	0.893	0.826	0.768
BERT *	-	0.910	0.839	0.798
BERT-CNN *	-	0.920	-	-
SKS *	0.951	0.963	-	-
MSF	**0.964**	**0.972**	**0.894**	**0.864**

Note: The results with superscript * are imported from the literature.

4.2. Comparison with Baselines

We used the following baseline methods for comparison to comprehensively evaluate the performance of our method.

SVM. This method uses statistical features and linear support vector machine as a classifier to detect offensive language [2,11].

CNN. This method uses a convolutional layer and maximum pooling to detect offensive language [29].

Bi-GRU. This method uses a bidirectional gated recurrent unit network to extract the semantic features of the text [29].

Bi-GRU-CNN. This method uses hybrid Bi-GRU-CNN for offensive-language detection [29].

Bi-LSTM. This method uses the bidirectional long- and short-term memory network to obtain semantic features of the text and to detect offensive language [29].

Bi-LSTM-Att. In addition to Bi-LSTM, this method adds an attention mechanism to extract text semantic features [30].

BERT. This model uses a fine-tuning BERTbase for offensive-language detection [31,32].

BERT-CNN. This method uses hybrid BERT-CNN for offensive-language detection [31].

SKS. This method detects offensive language based on emotional knowledge sharing [33].

MSF. MSF is our proposed model, which detects offensive language based on multi-semantic fusion.

The results of the comparisons are listed in Table 3. From the results, we observe that:

(1) The performance of traditional machine-learning methods based on the bag-of-words model was not satisfactory. Using only n-grams character features and shallow semantic features cannot well characterize offensive language. The experimental results show that the performance of SVM is quite different from mainstream deep-learning methods, and its generalization ability is weak.

(2) The performance of RNN is better than CNN. After the introduction of the attention mechanism, the performance was further improved, and a deeper hierarchical network structure, such as BiGRU-CNN, can better extract deep semantic features.

(3) BERT uses the transformer encoder structure, which can capture longer-distance dependencies. Therefore, it is more efficient than RNN and CNN, and its performance was generally improved by about 1% on offensive-language-detection tasks. Based on BERT, after adding the CNN layer, the performance of the model was further improved.

(4) Zhou et al. [33] adopted a multi-task learning method and introduced more sentiment features. Compared with other methods, SKS performance was significantly improved. However, this model needs to use a significant amount of external sentiment resources, and the model structure is complicated.

(5) MSF, our proposed method, achieved the best performance on both datasets. For the DV dataset, MSF improved upon ordinary BERT by 6.2% for weighted F1 and BERT-CNN by 5.2%. Even compared with the strong baseline SKS, the performance of our model was superior by 0.9%. For the OLID dataset, MSF improved upon ordinary BERT by 6.6%. MSF can learn the deep semantics and character features of offensive language, such as semantic information and n-grams features, and apply interaction fusion mechanisms to fuse and adjust the potential feature interactions between them.

4.3. Ablation Experiment

We analyze the impact of different parts in this section. The results are shown in Figure 2, where "Not Aug" denotes ablation of augmentation. Similarly, "Not DSM" means that a deep semantic module was not used, and "Not CCM" means to cancel the character-capture module; "Not Inter" denotes that there is no interaction fusion mechanism.

Based on the results in Figure 2, we observed that:

(1) Data augmentation has a significant impact on model performance. When data augmentation was used, the performance was comparable to SKS and better than other baselines. When data augmentation was canceled, the performance of the model decreased significantly by about 5%. The reason for this is that data augmentation increases the diversity of data, and the model can learn richer semantic features without the need to introduce external resources.

(2) DSM has little effect on model performance. When the module was canceled, the model performance only decreased by about 2%. The reason for this is that our proposed model uses two modules to extract the semantic information of sentences. The semantic information extracted by the CCM module can better represent offensive language, and the adaptive fusion layer can better integrate the features extracted by the CCM module.

(3) CCM has a great impact on model performance. To our surprise, when CCM was canceled, the performance for OLID decreased by nearly 20%, and it also decreased by 15% for the DV dataset. The main reasons for this were as follows. First, the semantic features of the character level have a significant impact on the offensive-language detection task, which can effectively improve performance. Second, the features extracted by the deep network are not suitable to be used as the input for the interactive-fusion level alone. In our experiments, this structure decreased model performance significantly.

(4) The interactive-fusion level can fuse the semantic features extracted from the two modules of DSM and CCM, which is a useful supplement to the model.

(5) Our proposed model uses data augmentation, DSM, CCM, and an interactive fusion mechanism to achieve state-of-the-art performance.

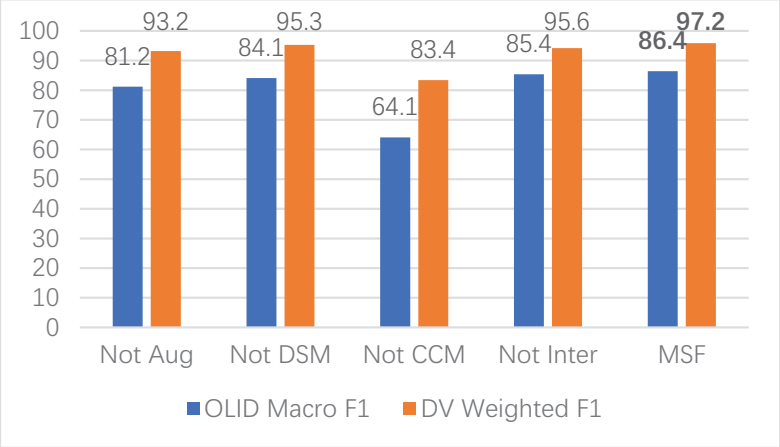

Figure 2. The results of ablation experiment.

4.4. The Influence of the N-Grams

Table 4 shows the influence of the value of n-grams on the performance. When n was small, the performance was relatively poor. As n increased, the performance gradually improved. When $n = 4$, the model achieved the best performance on both datasets. When n continued to increase, the performance decreased. Therefore, the value of $n = 4$ in our experiment.

Table 4. Effect of different values of n-grams.

Model	N-Grams	Dataset	Acc	F1
MSF	3	OLID	0.864	0.829
	4		**0.894**	**0.864**
	5		0.876	0.838
	3	DV	0.876	0.817
	4		**0.964**	**0.972**
	5		0.951	0.953

Note: OLID's F1 is Macro F1, and DV's F1 is Weighted F1.

4.5. Error Analysis

To better study the problems of the model proposed in this study and in the hope of further improvement in future offensive-language detection tasks, we analyzed the error examples. The following is an example that the model does not recognize correctly:

exp1: Alex this is so fucking beautiful.

exp2: #auspol I don't know why he is still in his job. Seriously.

Exp 1 is normal speech, but the model recognizes it as offensive language. This is mainly because the word *fucking* is common in offensive language, but here it is only a means to emphasize the tone. Exp 2 is an offensive sentence, but our model did not classify it correctly. This sentence lacks obvious sentiment words, the expression is more obscure, and its offensive meaning is implicit in the semantics of the text. Similar examples are common in political sentences.

5. Conclusions and Future Work

In this work, we focused on offensive-language detection through data augmentation, n-grams character features, and semantic fusion. For this purpose, we employed back translation to augment the data, used the DSM to extract semantic features, and, to extract n-grams features, we used CCM. Finally, using an interactive fusion mechanism, we fused the features extracted by the two modules. Extensive experiments were conducted on two offensive-language datasets, which showed that back translation significantly improve model performance and that multiple semantic-feature information can complement each other to improve model performance based on the interactive fusion mechanism. Our proposed model outperformed the strong state-of-the-art baselines.

Sentences containing strong negative-sentiment-polarity words are more likely to be offensive-language sentences; so, sentiment features are an important feature for automatic detection of offensive language. How to better dig out the sentiment features of sentences and transfer them is a direction worthy of our attention. In addition, there are many implicit-sentiment-expression sentences in offensive language, and it is difficult to accurately judge whether they contain offensive meaning by simply using monomodal features, such as text features. So, how to introduce multi-modal features is also a direction worthy of further research.

Author Contributions: Conceptualization, J.L.; Data curation, J.L.; Formal analysis, J.L.; Funding acquisition, J.L., Y.Y. and X.F.; Investigation, Y.Y.; Methodology, J.L. and X.F.; Project administration, Y.Y.; Software, G.R.; Supervision, X.F., G.R. and Q.N.; Validation, L.Y. and Q.N.; Visualization, L.Y. and Q.N.; Writing — original draft, J.L.; writing—review and editing, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant from the Natural Science Foundation of China (No. 62066044, 62167008). This work was also supported by XinJiang Uygur Autonomous Region Natural Science Foundation Project No. 2021D01B82, Joint Funds of the Key Project of XinJiang No. U1903215, and major science and technology projects of Yunnan Province No. 202002ab080001-1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

1. Waseem, Z. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In Proceedings of the Association for Computational Linguistics, Austin, TX, USA, 5 November 2016. [CrossRef]
2. Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; Kumar, R. Predicting the type and target of offensive posts in social media. *arXiv* **2019**, arXiv:1902.09666.
3. Wiegand, M.; Siegel, M.; Ruppenhofer, J. Overview of the germeval 2018 shared task on the identification of offensive language. In Proceedings of the GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria, 21 September 2018.
4. Gambäck, B.; Sikdar, U.K. Using Convolutional Neural Networks to Classify Hate-Speech. In Proceedings of the Association for Computational Linguistics, Vancouver, BC, Canada, 4 August 2017. [CrossRef]
5. 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; ACM: New York, NY, USA, 2017. [CrossRef]
6. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the Association for Computational Linguistics, Vancouver, BC, Canada, 4 August 2017. [CrossRef]
7. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. Available online: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 2 January 2022).
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
9. Sigurbergsson, G.I.; Derczynski, L. Offensive language and hate speech detection for Danish. *arXiv* **2019**, arXiv:1908.04531.
10. Pitenis, Z.; Zampieri, M.; Ranasinghe, T. Offensive language identification in Greek. *arXiv* **2020**, arXiv:2003.07459.
11. Davidson, T.; Warmsley, D.; Macy, M.W.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 May 2017; AAAI Press: Palo Alto, CA, USA, 2017.
12. Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Zampieri, M.; Nakov, P. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. *arXiv* **2020**, arXiv:2004.14454.
13. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
14. Luque, F.M. Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis. *arXiv* **2019**, arXiv:1909.11241.
15. Zhang, Y.; Ge, T.; Sun, X. Parallel data augmentation for formality style transfer. *arXiv* **2020**, arXiv:2005.07522.
16. Liu, S.; Lee, K.; Lee, I. Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. *Knowl.-Based Syst.* **2020**, *197*, 105918. [CrossRef]
17. Min, J.; McCoy, R.T.; Das, D.; Pitler, E.; Linzen, T. Syntactic data augmentation increases robustness to inference heuristics. *arXiv* **2020**, arXiv:2004.11999.
18. Zhang, Z.; Robinson, D.; Tepper, J.A. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *The Semantic Web, Proceedings of the 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10843, Lecture Notes in Computer Science, pp. 745–760._48. [CrossRef]
19. Shen, D.; Wang, G.; Wang, W.; Min, M.R.; Su, Q.; Zhang, Y.; Henao, R.; Carin, L. On the Use of Word Embeddings Alone to Represent Natural Language Sequences. Available online: https://openreview.net/pdf?id=Sy5OAyZC- (accessed on 2 January 2022).
20. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018. [CrossRef]

21. Alatawi, H.S.; Alhothali, A.M.; Moria, K.M. Detecting White Supremacist Hate Speech Using Domain Specific Word Embedding With Deep Learning and BERT. *IEEE Access* **2021**, *9*, 106363–106374. [CrossRef]
22. Sohn, H.; Lee, H. MC-BERT4HATE: Hate Speech Detection using Multi-channel BERT for Different Languages and Translations. In Proceedings of the 2019 International Conference on Data Mining Workshops, ICDM Workshops 2019, Beijing, China, 8–11 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 551–559. [CrossRef]
23. Lou, C.; Liang, B.; Gui, L.; He, Y.; Dang, Y.; Xu, R. Affective Dependency Graph for Sarcasm Detection. Available online: http://wrap.warwick.ac.uk/153596/7/WRAP-affective-dependency-graph-sarcasm-detection-Gui-2021.pdf (accessed on 2 January 2022).
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
25. Mehdad, Y.; Tetreault, 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.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 770–778. [CrossRef]
27. Chen, X.; Sun, C.; Wang, J.; Li, S.; Si, L.; Zhang, M.; Zhou, G. Aspect Sentiment Classification with Document-level Sentiment Preference Modeling. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3667–3677. [CrossRef]
28. Salas-Eljatib, C.; Fuentes-Ramirez, A.; Gregoire, T.G.; Altamirano, A.; Yaitul, V. A study on the effects of unbalanced data when fitting logistic regression models in ecology. *Ecol. Indic.* **2018**, *85*, 502–508. [CrossRef]
29. Ong, R. Offensive Language Analysis using Deep Learning Architecture. *arXiv* **2019**, arXiv:1903.05280.
30. Altin, L.S.M.; Serrano, À.B.; Saggion, H. LaSTUS/TALN at SemEval-2019 Task 6: Identification and Categorization of Offensive Language in Social Media with Attention-based Bi-LSTM model. In Proceedings of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019. [CrossRef]
31. Aggarwal, P.; Horsmann, T.; Wojatzki, M.; Zesch, T. LTL-UDE at SemEval-2019 Task 6: BERT and Two-Vote Classification for Categorizing Offensiveness. In Proceedings of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019. [CrossRef]
32. Mozafari, M.; Farahbakhsh, R.; Crespi, N. Hate Speech Detection and Racial Bias Mitigation in Social Media based on BERT model. *arXiv* **2020**, arXiv:2008.06460.
33. 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 Association for Computational Linguistics, online, 6–11 June 2021. [CrossRef]