Hate Speech Detection based on Sentiment Knowledge Sharing

Xianbing Zhou[1], **Yong Yang**[1], **Xiaochao Fan**[1]*,
Ge Ren[1], **Yunfeng Song**[1], **Yufeng Diao**[2], **Liang Yang**[2], **Hongfei Lin**[2]*
[1]School of Computer Science and Technology, Xinjiang Normal University, China
[2]Department of Computer Science and Technology, Dalian University of Technology, China
{1783696285, 68523593, 37769630, 236789497, 1697277502}@qq.com
{diaoyufeng, liang}@mail.dlut.edu.cn, hflin@dlut.edu.cn

Abstract

The wanton spread of hate speech on the internet brings great harm to society and families. It is urgent to establish and improve automatic detection and active avoidance mechanisms for hate speech. While there exist methods for hate speech detection, they stereotype words and hence suffer from inherently biased training. In other words, getting more affective features from other affective resources will significantly affect the performance of hate speech detection. In this paper, we propose a hate speech detection framework based on sentiment knowledge sharing. While extracting the affective features of the target sentence itself, we make better use of the sentiment features from external resources, and finally fuse features from different feature extraction units to detect hate speech. Experimental results on two public datasets demonstrate the effectiveness of our model.

1 Introducon

With the prevalence of mobile Internet and social media, phenomena such as the malicious spread of hate speech have gradually become widespread. This often has incalculable consequences and has become a serious social problem. How to quickly and accurately detect hate speech automatically, and then better intervene to prevent it has become one of the hot research issues in the field of natural language processing. The automatic detection of hate speech can prevent the viral spread of hate speech, thereby reducing the malicious spread of cyberbullying and harmful information. In the field of public opinion analysis, monitoring and intervention, hate speech detection has extensive value in application.

In recent years, the hate speech detection has been paid more attention, and many research results have appeared. However, the task is quite challenging due to the inherent complexity of the natural language constructs. Most of the existing works revolves either around rules (Krause and Grassegger, 2016) or manual feature extraction (Gitari et al., 2015). Rule-based methods do not involve learning and typically rely on a pre-compiled list or dictionary of subjectivity clues (Haralambous and Lenca, 2014). Chen et al. (2012) proposed a variety of linguistic rules to determine whether a sentence constitutes hate speech or not. For example, if a second-person pronoun and a derogatory word appear at the same time, such as "<you, gay>", the sentence is judged to be insulting. This type of method not only requires manual formulation of rules, but also requires dictionaries of derogatory words. There have also been many attempts to detect hate speech using traditional machine learning methods. Mehdad and Tetreault (2016) extracted the n-gram, character-level and sentiment features of text and used support vector machines (SVM) to detect hate speech. However, artificial features can only reflect the shallow features of text and cannot understand content from the deep semantic features.

Deep learning methods have been widely used in the field of hate speech detection and have achieved good performance (Badjatiya et al., 2019; Qian et al., 2018) in recent years. Wang (2018) compared the performance of various neural network models in detecting hate speech and used visualization techniques to give the models better interpretability. The semantics of hate speech contains a strong negative sentiment tendency. The deep learning methods of predecessors often only used pre-trained models or deeper networks to obtain semantic features, ignoring the sentiment features of the target sentences and external sentiment resources, which also makes the performance of neural networks unsatisfactory in hate speech detection.

To overcome the weaknesses of previous works,

čorresponding author: Xiaochao Fan, Hongfei Lin.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 7158–7166
August 1–6, 2021.

we propose a hate speech detection framework based on sentiment knowledge sharing (SKS)[1]. Our intuition is that most hate speech contains words with strong negative emotions, which are usually the most direct clues to hate speech. Meanwhile, as claimed by Davidson et al. (2017), lexical detection methods tend to have low precision because they classify all messages containing particular terms as hate speech. Therefore, we hope to make better use of external sentiment resources so that the model can learn sentiment features and share them, which will greatly affect the performance of hate speech detection. In addition, inspired by the recent MoE layer (Shazeer et al., 2017) and the Multi-gate Mixture-of-Experts (MMoE) model (Ma et al., 2018), we use multiple feature extraction units and use a gated attention mechanism to fuse features. The main contributions of this work are summarised as follows:

(1) In view of the lack of the use of sentiment information in previous works, we not only integrate the derogatory words of target sentences into the neural network, but also use multi-task learning to make the model learn and share external sentiment knowledge.

(2) In order to better capture shared task or task-specific information, we propose a new framework which uses multiple feature extraction units where each extraction unit uses the multi-head attention mechanism and a feedforward neural network to extract features, and finally uses gated attention fuse features.

(3) Experimental results on the SemEval-2019 task-5 and Davidson datasets demonstrate that our method achieves state-of-the-art performance compared with strong baselines, and then further detailed examples verify the effectiveness of our presented model for hate speech detection.

2 Related Work

Hate speech is very dependent on the nuance of language. Even if it is manually distinguished whether certain sentences contain hate semantics, consensus is rare (Waseem, 2016). Recently, automatically detecting hate speech has been widely studied by researchers. In this section, we will review related works on traditional machine learning-based methods, deep learning-based methods, and multi-task learning-based methods of hate speech detection.

[1]Code is available at https://github.com/1783696285/SKS.

Machine learning-based methods based on feature engineering are widely used in the field of hate speech detection. Malmasi and Zampieri (2018) provided empirical evidence that n-gram features and sentiment features can be successfully applied to the task of hate speech detection. Rodríguez et al. (2019) constructed a dataset of hate speech from Facebook, and proposed a rich set of sentiment features, including negative sentiment words and negative sentiment symbols, to detect hate speech. Del Vigna12 et al. (2017) used the sentimental value of words as the main feature to measure whether a sentence constitutes hate speech. Gitari et al. (2015) designed several sentiment features and achieved good performance in experiments. Previous studies have shown that sentiment features play an important role in hate speech detection.

Recently, deep learning-based methods have garnered considerable success in hate speech detection. Zhang et al. (2018) fed input into a convolutional neural network (CNN) and a gated recurrent unit (GRU) to learn higher-level features. Kshirsagar et al. (2018) proposed a transformed word embedding model (TWEM), which had a simple structure but can achieve better performance than many complex models. Badjatiya et al. (2019) found that due to the limitation of the training set, the deep learning model would have "bias" and he designed and implemented a "bias removal" strategy to detect hate speech. Tekiroglu et al. (2020) constructed a large-scale dataset based on hate speech and its responses and used the pre-trained language model, GPT-2, to detect hate speech. Obviously, deep learning models can extract the latent semantic features of text, which can provide the most direct clues for detecting hate speech.

Multi-task learning can learn multiple related tasks and share knowledge at the same time. In recent years, there have been some achievements in the field of hate speech detection. Kapil and Ekbal (2020) proposed a deep multi-task learning (MTL) framework to leverage useful information from multiple related classification tasks in order to improve the performance of hate speech detection. Liu et al. (2019) introduced a novel formulation of a hate speech type identification problem in the setting of multi-task learning through their proposed fuzzy ensemble approach. Ousidhoum et al. (2019) presented a new multilingual multi-aspect hate speech analysis dataset and used

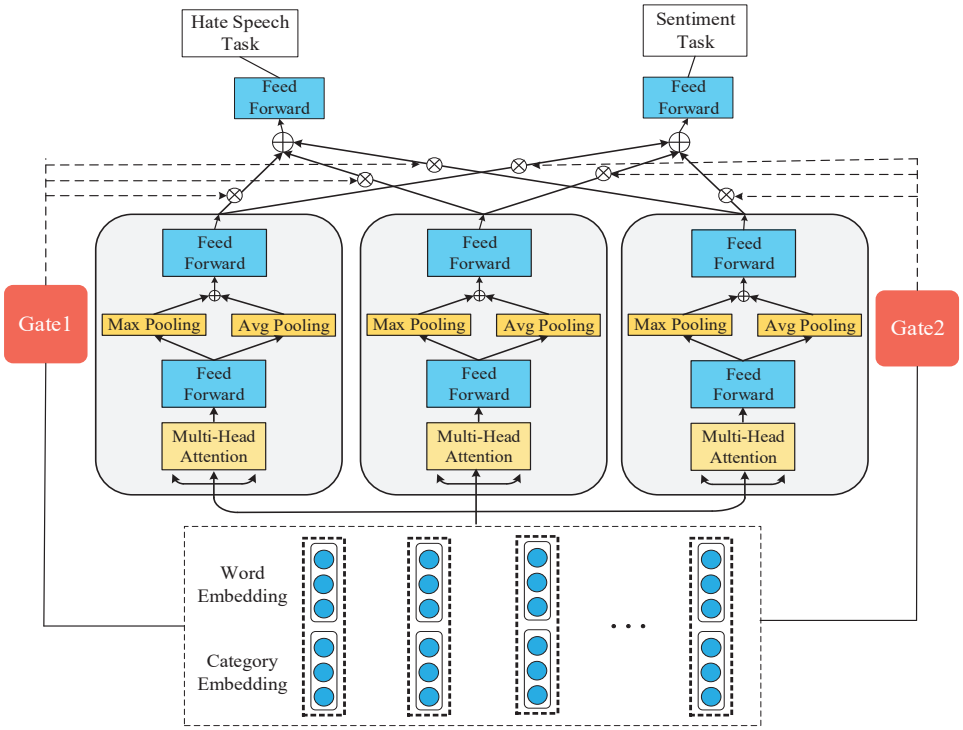

Figure 1: The overall framework of our proposed Hate Speech Detection based on Sentiment Knowledge Sharing(SKS).

it to test the current state-of-the-art multilingual multitask learning approaches. Ousidhoum et al. (2019) proposed BERT-based multi-task learning for offensive language detection. Some studies have shown that multi-task learning can improve the performance and generalization ability of models in hate speech detection by using the correlation between the task of sentiment analysis and hate speech detection.

3 Methodology

In this section we introduce our model, SKS. Our model is able to improve hate speech detection by considering both target sentence sentiment and external sentiment knowledge.

The overall architecture of SKS is shown in Figure 1. The framework consists mainly of three layers: 1) Input layer. In order to better obtain the sentiment features of the sentence itself, we use a derogatory words dictionary to judge whether each word is a hate word, and then append the category information to the word embedding. 2) Sentiment knowledge sharing layer. Since sentiment analysis and hate speech detection are highly correlated, we use the multi-task learning framework to model task relationships and learn task-specific features to take advantage of shared sentiment knowledge. We use multiple feature extraction units composed of a multi-head attention layer and a feedforward neural network. 3) Gated attention layer. A gated attention mechanism is used to output the probability that the feature extraction unit is selected. Finally, a feedforward neural network is used to detect hate speech.

3.1 Input Layer

Hate speech often contains obvious negative sentiment words because of the strong negative sentiment.

exp1: Go fucking kill yourself and die already useless ugly pile of shit scumbag.

The words “fucking”, “ugly”, and “shit scumbag” in exp1 are all obviously insulting and offensive, and they contain strong negative sentiment. Obviously, whether the word in the target sentence is a derogatory word is the most direct clue to judge hate speech. Therefore, paying attention to capturing derogatory words in a sentence can help us improve hate speech detection.

Word Embedding. Word Embedding is based on distributed assumptions and mapped words into a high dimension feature space and maintaining the semantic information. For each target sentence $S = \{w_1, w_2, , w_N\}$, we transform each token w_i

into a real valued vector x_i using word embedding, where $x_i \in \mathbb{R}^d$ is the word vector, d is dimensions of word vectors.

Category Embedding. Our work is strongly based on the intuition that hate speech arises from derogatory words. In other words, some specific words that are extremely insulting will make a greater contribution to judging hate speech. Therefore, we have established a derogatory word dictionary. The vocabulary comes from Wikipedia[2] and another website[3], including Hate Speech, Disability, LGBT, Ethnic, and Religious, with 5 categories. Since the vocabulary contains 2 or 3 word phrases, when judging whether it is a dirty word, we use n-gram, n ∈ [1,2,3].

The derogatory word dictionary is used to divide tweet into two categories, either containing derogatory words or not containing derogatory words,and then assign the two categories to each word in the tweet. The category of each word is initialized randomly as vector $C = (c_1, c_2, , c_n)$, $c_i \in \mathbb{R}^{d'}$.

Since the common word embedding representations exhibit a linear structure, that makes it possible to meaningfully combine words by an element-wise addition of their vector representations. In order to better take advantage of information within derogatory words, we append the category representation to each word embedding. The embedding of a word x_i for a category embedding c_i is $x_i^{'} = x_i \oplus c_i$, where $\oplus$ is the vector concatenation operation.

3.2 Sentiment Knowledge Sharing Layer

Due to the influence of different countries, regions, religions and cultures, insulting meanings in many languages are hidden in the underlying semantics, rather than just reflected in sentiment words.

exp2: Jews are lower class pigs.

exp3: i'm so fucking ready!

There are no obvious negative sentiment words in exp2, but the sentence constitutes hate speech. Although "pig" is a neutral word, most people equate the word "pig" with stupid and clumsy. Comparing "Jews" and "pig" is obviously an insult to "Jews". Latent semantics and common sense of sentiment are the keys to correctly judging the sentence. Exp3 contains the word "fucking" with a strong negative sentiment. This word often appears in hate speech. However, in this sentence, "fucking" does not specifically refer to a person, but is just an adverb of degree, which strengthens the tone. It is not hate speech. It can be seen from the above example that although hate speech often contains negative sentiment words, only using the sentiment information of the target sentence itself to detect hate speech often makes it difficult to obtain satisfactory performance.

Deep learning methods require a large amount of labelled data for supervised learning, which needs more human effort and prior knowledge of this particular task. High-quality annotation data is scarce in hate speech detection, which makes the task stereotype words and hence suffer from inherently biased training. Sentiment analysis research has been carried out for many years, and there are abundant high-quality labelled datasets. There is a high degree of correlation between two tasks, and multi-task learning can use the correlation between multiple tasks to improve the performance and generalization ability of the model in each task. Therefore, we adopt a multi-task learning method for sentiment knowledge sharing, so as to better extract sentiment features and apply them to hate speech detection.

The framework of multi-task learning widely uses a shared-bottom structure, and different tasks share the bottom hidden layer. This structure can essentially reduce the risk of overfitting, but the effect may be affected by task differences and data distribution. We adopt the framework structure of Mix-of-Expert (MoE). The MoE layer has multiple identical feature extraction units, which share the output of the previous layer as input and outputs to a successive layer. Then, the whole model is trained in an end-to-end way. Our feature extraction units layer is composed of a multi-head attention layer and two feed forward neural networks.

Multi-head Attention Layer. The self-attention mechanism connects any two words in a sentence by calculating the semantic similarity and semantic features of each word in the sentence and other words so as to better obtain the long-distance dependency. The multi-head self-attention proposed by Vaswani et al. (2017) is used in this section. For a given query $Q \in \mathbb{R}^{(n_1 \times d_1)}$, key $K \in \mathbb{R}^{(n_1 \times d_1)}$, value $V \in \mathbb{R}^{(n_1 \times d_1)}$, we use the dot product to calculate attention parameters. The formula is as follows:

$$\text{Attention(Q, K, V)} = \text{softmax}\left(\frac{QK^T}{d_1}\right)\text{V} \quad (1)$$

[2]https://www.wikipedia.org/

[3]https://www.noswearing.com/

where d_1 is the number of hidden layer unites.

The multi-head attention mechanism maps the input vector X to query, key, and value using linear changes. In our task, key=value. Then, the model learns the semantic features between words through the l-time attention. For the i-th attention head, let the parameter matrix $\mathrm{W}_i^Q \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$, $\mathrm{W}_i^K \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$,$\mathrm{W}_i^V \in \mathbb{R}^{n_1 \times \frac{d_1}{l}}$, we use the dot product to calculate the semantic features between them:

$$M_i = \mathrm{Attention}(\mathrm{QW}_i^Q, \mathrm{KW}_i^K, \mathrm{VW}_i^V) \quad (2)$$

The vector representation obtained by the multi-head attention mechanism is concatenated to obtain the final feature representation:

$$H^s = \mathrm{concat}\left(M_1, M_2, \ldots, M_l\right) W_o \quad (3)$$

Pooling Layer. Shen et al. (2018) used maximum pooling and average pooling to fuse features. Experimental results showed that the performance of this method is significantly better than using a single pooling strategy. Therefore, we use maximum pooling and average pooling at the same time. The formula is as follows:

$$P_m = \mathrm{Pooling\_max}\left(\mathrm{H}^s\right) \quad (4)$$

$$P_a = \mathrm{Pooling\_average}\left(\mathrm{H}^s\right) \quad (5)$$

$$P_s = \mathrm{concat}\left(\mathrm{P_m}, \mathrm{P_a}\right) \quad (6)$$

3.3 Gated Attention

Gated attention can learn to select a subset of the feature extraction units to use, conditioned on the input. For different tasks, the weight selection of the model is different, so each task has a Gate. The output of a specific gate k represents the probability of a different feature extract unit being selected, and multiple units are weighted and summed to obtain the final representation of the sentence, which will be passed into the exclusive layer of the task. Our gating unit has the same structure as the feature extraction unit. The formula is as follows:

$$g^k(x) = \mathrm{softmax}\left(W_{gn} * gate\left(x\right)\right) \quad (7)$$

$$f^k(x) = \sum_{i=1}^{n} g^k(x)_i f_i(x) \quad (8)$$

Dataset	total	Classes
SE	11,971	hate (5,035) non-hate (6,936)
DV	24,783	hate (1,430) non-hate (23,353)
SA	31,962	negative(2,242) positive(29,720)

Table 1: Statistics of datasets used in the experiment.

$$y_k = \mathrm{h}^k \mathrm{f}^k(x) \quad (9)$$

where k is the number of tasks and h is the hidden layer representation.

3.4 Model Training

For training process, the whole parameters can be optimized from our networks. Then, cross entropy is applied with L2 regularization as the loss function, which is defined as:

$$loss = -\sum_i \sum_j y_i^j \log \hat{y}_i^j + \lambda \|\theta\|^2 \quad (10)$$

where i is the index of sentences, j is the index of class, λ is the L_2 regularization term, θ is the parameter set.

4 Experiments

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

4.1 Datasets and evaluation metrics

We try to explore whether sharing sentiment knowledge can improve the performance of hate speech detection. Therefore, two public hate speech datasets and one sentiment dataset is used in our experiment. The details of the datasets are shown in Table 1.

SemEval2019 task5 (SE) (Basile et al., 2019). The SE comes from SemEval 2019 task 5, and subtask A is hate speech detection. The dataset is divided into three subsets. The training contains 9000 cases, the validation contains 1000 cases, and the test contains 2971 cases.

Davidson dataset (DV) (Davidson et al., 2017). The DV dataset was constructed by Davidson who implemented a web-based bootstrapping algorithm to automatically collect a large number of hate

speech examples from Tweets. This is an unbalanced dataset with less hate speech.

Sentiment Analysis (SA)[4]. The SA is a sentiment dataset from Kaggle2018. The SA contains more positive cases, but fewer negative cases. Since the test set is unlabelled, we only use the training set.

For comparison with baseline methods, Accuracy (Acc) and F-measure (F1) are used as evaluation metrics in our hate speech detection.

4.2 Training Details

In SemEval2019 evaluation, the performance of the test set is the final result. To compare with published papers, the results of the test set are used on the dataset and we use Acc and micro F1 as metrics. For the DV dataset, we use a 5-fold cross-validation method to measure the performance of the proposed model. To compare with previous works, We report results of DV using the standard Accuracy and weighted F1.

In our experiments, for the input layer, all word vectors are initialized by Glove Common Crawl Embeddings (840B Token), and the dimension is 300. The category embeddings are initialized randomly, and the dimension is 100. For the sentiment knowledge sharing layer, the multi-head attention has 4 heads. The first Feed-Forward network has one layer with 400 neurons and the second has two layers with 200 neurons. The dropout is used after each layer, and the rate is 0.1. The optimizer is RMSprop, and the learning rate is 0.001. The models are trained by a mini-batch of 512 instances. To prevent overfitting, we use the learning rate decay and early stop in the training process.

4.3 Comparison with Baselines

We compare our proposed model with several strong baselines:

SVM. It is proposed by Zhang et al. (2018) and Basile et al. (2019). The author implemented several features, such as n-gram, misspellings, derogatory words.

LSTM and GRU. The method was proposed by Ding et al. (2019). LSTM and GRU were used to extract the features of target sentences.

CNN-GRU. Zhang et al. (2018) employed word embedding and learnt the latent semantic representations through a hybrid neural network CNN-GRU.

[4]https://www.kaggle.com/dv1453/twitter-sentiment-analysis-analytics-vidya

BiGRU-Capsule. This baseline was proposed by Ding et al. (2019). Two-layer BiGRU and a capsule layer were used to detect hate speech.

Universal Encoder. It was proposed by Indurthi et al. (2019). The author used sentence embeddings, such as lexical vectors and deep contextualized word representations, to detect hate speech.

BERT and GPT. They were proposed by Benballa et al. (2019). The pre-trained model BERT and GPT were used to capture the features to detect hate speech.

SKS. SKS is our proposed model which detects hate speech based on sentiment knowledge sharing.

The overall performance comparison of SKS is shown in Table 2. From Table 2, we can see that:

(1) Overall, the performance of the model is quite different on the two datasets. For the DV dataset, the F1 value is about 90%, while for the SE dataset, the F1 value is less than 60%. This is mainly because there are few negative examples in teh DV, and the model does not learn enough useful features. Furthermore, the nuance of the language can significantly affect the performance of the model.

(2) The performance of SVM based on features is much worse than the neural network. Especially on the SE dataset, performance is unacceptable. This indicates that the neural network can better capture the semantic relationships of words for hate speech detection.

(3) The performance of the hybrid neural network is better than the simple Recurrent Neural Network (RNN). Compared with the traditional RNNs, such as LSTM and so on, whether CNN-GRU or BiGRU-capsule, its performance has a small improvement. By stacking a layer of a neural network onto another, a deep learning model is helpful for better learning of high-level features. The traditional RNNs, such as LSTM and GRU, have almost the same performance.

(4) BERT achieves better performance on the DV dataset. However, both BERT and GPT achieve worse performance on the SE dataset. The experimental results show that the pre-training model is very dependent on the training data. For the specific field, it is difficult to provide good feature representations without suitable and sufficient data.

(5) Our proposed method, SKS, achieves the

Model	DV		SE	
	Acc	F1(wei)	Acc	F1(macro)
SVM*	-	87.0	49.2	45.1
LSTM*	94.5	93.7	55.0	53.0
GRU*	94.5	93.9	54.0	52.0
CNN-GRU*	-	94.0	62.0	61.5
BiLSTM*	94.4	93.7	53.5	51.9
BiGRU_Stacked*	-	-	56.0	54.6
USE_SVM*	-	-	65.3	65.1
BERT*	94.8	95.8	-	48.8
GPT*	-	-	-	51.5
SKS	**95.1**	**96.3**	**65.9**	**65.2**

Table 2: Comparison with existing methods. The results with superscript * are imported from the literature. The best results in each type are highlighted.

Model	DV		SE	
	Acc	F1(wei)	Acc	F1(macro)
-sc	94.0	94.0	59.6	59.3
-s	94.5	94.3	61.3	61.3
SKS	**95.1**	**96.3**	**65.9**	**65.2**

Table 3: the results of ablation experiments The best results in each type are highlighted.

Model	DV		SE	
	Acc	F1(wei)	Acc	F1(macro)
no-gate	94.8	95.9	64.7	64.3
SKS	**95.1**	**96.3**	**65.9**	**65.2**

Table 4: the influence of gated attention.

best performance for F1. Compared with other neural networks, including LSTM, GRU and BiLSTM, the F1 value of SKS is increased by nearly 3% on the DV dataset, and on the SE dataset, the performance of SKS greatly improves to nearly 10%. Even compared with the strong baseline model, universal encoder, our model is superior. The SKS is easier to implement and has fewer parameters.

4.4 Ablation Experiments

We then analyze the influence of different parts of our model. The results are shown in Table 3, where "–sc" denotes ablation of sentiment knowledge sharing and the category embedding. Similarly, "-s" means that sentiment data is not used as input for the model, and it only uses category embedding.

Based on the results in Table 3, we can see that: 1) The performance on the two datasets decreases significantly with the model ablation of sentiment knowledge sharing and category embedding. However, the performance of the model is better than the existing hybrid neural networks. It is shown that this framework can better learn the latent semantic features of the target sentence. 2) The performance of our model is improved slightly when the category embedding is used. The main reason is that the information of derogatory words is highly related to hate speech, but it will also make the model too sensitive. Therefore, the direct extraction of derogatory words' sentiment features has a limited impact on the performance. 3) SKS outperforms the other models, which proves the effectiveness of sentiment knowledge sharing directly.

We also analyse the role of gated attention in our model. As shown in Table 4, the performance of the model is further improved on both datasets when the gated attention is used. This framework is able to model the task relationships in a sophisticated way by deciding how the separations resulting from different gates overlap with each other (Ma et al., 2018). Each gated network can learn to select which feature extraction unit is used on the input cases. If the tasks are highly related, then sharing knowledge will achieve better performance.

4.5 The Influence of The Scale of Sentiment Dataset

Hate speech detection and sentiment analysis are highly correlated, so that sentiment knowledge sharing can improve the performance of hate

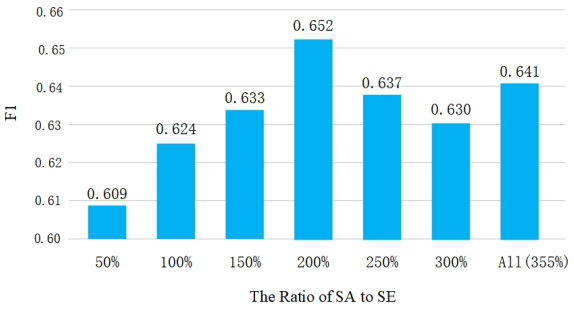

Figure 2: the influence of the scale of sentiment data set.

speech detection. But we cannot ignore the impact of the scale of the sentiment dataset on the performance. Since the scale of the DV is similar to the SA dataset, we focus our analysis on the SE dataset.

As shown in Figure 2, the performance of the model is poor when the ratio of the two types of data is 1:2. As the ratio of sentiment data increases, the performance of the model is improved. When the ratio is 2:1, the performance reaches a peak, and then maintains a declining trend. It is observed that the ratio of multi-task data will also directly affect the performance.

5 Conclusion and Future Work

In this paper, we explore the effectiveness of multi-task learning in hate speech detection tasks. The main idea is to use multiple feature extraction units to share multi-task parameters so that the model can better share sentiment knowledge, and then gated attention is used to fuse features for hate speech detection. The proposed model can make full use of the sentiment information of the target and external sentiment resources. We show that sentiment knowledge sharing improves system performance over the baselines and advances hate speech detection. Finally, the detailed analysis further proves the validity and interpretability of our model.

Overall, our experiments give us a better understanding of the relationship between hate speech detection and sentiment analysis through multi-task learning. We have laid the groundwork for future efforts in better modelling and data selection, including different types of hate speech, the type and scale of sentiment data, and so on.

6 Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by grant from the Natural Science Foundation of China (No.62066044, 61632011, 62076046). This work was also supported by Xinjiang Uygur Autonomous Region Natural Science Foundation Project No.2021D01B72 and National Youth Science Fund Project No.62006130.

References

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, pages 49–59.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Miriam Benballa, Sebastien Collet, and Romain Picot-Clemente. 2019. Saagie at semeval-2019 task 5: From universal text embeddings and classical features to domain-specific text classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 469–475.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Fabio Del Vigna12, Andrea Cimino23, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, pages 86–95.

Yunxia Ding, Xiaobing Zhou, and Xuejie Zhang. 2019. Ynu_dyx at semeval-2019 task 5: A stacked bigru model based on capsule network in detection of hate. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 535–539.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based

approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Yannis Haralambous and Philippe Lenca. 2014. Text classification using association rules, dependency pruning and hyperonymization. *arXiv preprint arXiv:1407.7357*.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. Fermi at semeval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74.

Prashant Kapil and Asif Ekbal. 2020. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458.

Till Krause and Hannes Grassegger. 2016. Facebook's secret rules of deletion. *Süddeutsche Zeitung*.

Rohan Kshirsagar, Tyrus Cukuvac, Kathy Mckeown, and Susan Mcgregor. 2018. Predictive embeddings for hate speech detection on twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online, EMNLP 2018, Brussels, Belgium*, pages 26–32.

Han Liu, Pete Burnap, Wafa Alorainy, and Matthew L Williams. 2019. Fuzzy multi-task learning for hate speech type identification. In *The World Wide Web Conference*, pages 3006–3012.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. *arXiv preprint arXiv:1908.11049*.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018. Hierarchical cvae for fine-grained hate speech classification. *arXiv preprint arXiv:1809.00088*.

Axel Rodríguez, Carlos Argueta, and Yi-Ling Chen. 2019. Automatic detection of hate speech on facebook using sentiment and emotion analysis. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 169–174. IEEE.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Ricardo Henao, and Lawrence Carin. 2018. On the use of word embeddings alone to represent natural language sequences.

Serra Sinem Tekiroglu, Yi-Ling Chung, and Marco Guerini. 2020. Generating counter narratives against online hate speech: Data and strategies. *arXiv preprint arXiv:2004.04216*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Cindy Wang. 2018. Interpreting neural network hate speech classifiers. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 86–92.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer.