KAREN: Unifying Hate Speech Detection and Benchmarking

Tiago Antunes
2020280599
Tsinghua University
vazama10@mails.tsinghua.edu.cn

Yuqing Cui
2021403079
Tsinghua University
cuiyq20@mails.tsinghua.edu.cn

Samuel Pegg
2020280261
Tsinghua University
peggsr10@mails.tsinghua.edu.cn

Abstract

Hate speech has existed for a very long time within our societies, but the recent development of the internet and social media has boosted the prevalence and visibility of hate speech and facilitated its spread. The call for automatized hate speech detection tools has seen a growing response from both the industry and academia, but many problems like the lack of a baseline dataset and a standardisation of evaluation metrics makes it hard to compare the different proposed solutions. This has lead to many models experiencing poor generalisability when applied to different datasets, or used in the real world.

Inspired by (Cen et al., 2021), we propose **KAREN**, a new and extensible framework that provides an easy to use toolkit containing different datasets and baseline models, together with an NLP toolkit containing different tools ready to be used. With this easy to use environment, implementation of models using KAREN is extremely easy and fast, thus allowing researchers to spend less time writing the evaluation environment and directly compare the results in their own system. KAREN is publicly available on our GitHub[1].

1 Introduction

Hate speech is defined as "any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characterstic" (Nockleby, 2000). The increasing accessibility of the internet and the popularity of social media and online forums has enabled more people to express themselves online, while also facilitating the spread of inappropriate content, including hate speech.

With the massive volume of online traffic, automated hate speech detection is not only essential for maintaining a healthy environment in online platforms, but also has deeper social implications in terms of crime incident prediction and prevention, e.g. (Wang et al., 2012).

Interest and studies in the field of hate speech detection have emerged in the past decade. Both traditional methods such as SVM, LDA, Logistic Regression, and deep learning methods, such as CNN, RNN and Transformer-based architectures have been explored by researchers. Such works have been developed using a wide range of datasets collected from the Internet, often private and only specific to the task presented in the work, leading to lack of generalizability for such methods (Yin and Zubiaga, 2021).

In this work, we propose **KAREN**, a benchmark framework that enables the training and evaluation of deep-learning models using a selection of datasets on various metrics. We believe such a framework can benefit researchers by providing insights on the performance of their models on a set of benchmark datasets with standardized metrics, alongside providing an easy comparison to other methods.

KAREN provides and intuitive model/dataset implementation that allows researchers and enthusiasts alike to focus on their models and theory, rather than spending hours on setting up a whole system to test and validate their experiments. We also allow completely customizable user defined dataset parsing for any new datasets they wish to implement – though this is of course not necessary, as they can use the datasets already implemented. It also tackles the previously mentioned problems by providing a compatibility evaluation system, filtering which models can run on each dataset and providing an intuitive way of accessing the required data. To give a general baseline for researchers to compare with, we have implemented many models. Simple models have been

[1] https://github.com/TiagoMAntunes/KAREN

implemented as well as some other, more complex ones.

This work is organised as follows: in section 2 we analyse recent work in hate speech and highlight its current problems; in section 3 we introduce our framework's architecture, how it has been designed and how it can be extended; in section 4 we analyse and compare the obtained results from the currently available models and datasets; in section 5 we discuss our findings and highlight key points in hate speech detection performance.

2 Related Work

Continuous effort has been put into this topic. Surveys from (Fortuna and Nunes, 2018) and (Schmidt and Wiegand, 2017) provide an overview of the history of the task and summarize the methods investigated. Their work highlights different problems in current research, such as word sparsity and the lack of context (which can highlight the hate, such as when someone uses sarcasm), and also points to some findings of current research such as the importance of having meta-data (by, for example, performing sentiment analysis on the content, as it is done in (Awal et al., 2021)) which can increase the accuracy of detection.

At the same time, multiple datasets were collected from online forums, such as Reddit (Mollas et al., 2020) and Twitter (Davidson et al., 2017a), and have been annotated by humans as categories of being hate speech, offensive speech, etc. Some datasets also included the rationale for the annotations, for better interpretability (Mathew et al., 2020) and it also serves as a way to justify its label. The datasets were used in conjunction with a variety of methods, including traditional machine learning models and deep neural network architectures, which resulted in an increased difficulty when comparing models. (Mollas et al., 2020) evaluated the proposed ETHOS dataset on methods such as naive bayes, random forest, logistic regression, deep neural networks, BERT, etc., achieving accuracies ranging from 48.3 (Bernoulli Naive Bayes) to 79.17 (BERT). With the widespread application of deep learning in natural language processing, multiple works proposed deep neural network architectures for hate speech detection. (Zhang, 2017) proposed a convolution-LSTM based network, yielding promising results on various public datasets (Davidson et al., 2017a), (Waseem, 2016), and (Waseem and Hovy, 2016). With the emergence of attention-based models, Transformers and BERT models have also been studied such as in (Awal et al., 2021) and (Caselli et al., 2020). The latter uses a large-scale dataset containing hate speech to pretrain BERT, yielding their solution called HateBERT, which is targeted towards abusive language detection. Recently, (Mathew et al., 2020) proposed HateXPlain, a dataset that they intend to be used a benchmark of hate speech detection, while (Aluru et al., 2021) proposed a collection of deep learning models that aim to generalise the problem of hate speech analysis and detection.

2.1 Problems in the Field

Despite the ongoing research, several problems persistently affect the obtained results. After looking into recent publications, we drew the following conclusions.

1) There is no standard benchmark dataset that allows for comparison across different models. Multiple works, such as (Zhang et al., 2018; Waseem and Hovy, 2016; Gaydhani et al., 2018) even provided a new dataset together with their solutions. The addition of new public datasets is of course beneficial for the growth of this area, but the unfortunate side effect is the **2) lack of generalisability**, as the performance of the model might be highly dependent on the dataset used and might perform better in another one, and vice-versa. This lack of generalisability is also related with the subjectivity of dataset annotation itself: hate speech is highly dependent on the social context and background of the specific individual choosing the category for each line of data (Waseem and Hovy, 2016), so what might be seen as hate speech by one individual, might seem acceptable to another individual. That is, **3) datasets contain an annotator bias** and so being able to test the model on multiple datasets would give a better idea of its real performance and quality. To also tackle this problem (Mathew et al., 2020) compiled HateXplain, a benchmark dataset containing hate speech and annotations that capture human rationale for the labeling of abusive words and phrases which is intended to be used as the standard dataset for evaluation on this topic. In addition, using several datasets to train and test a model reduces the bias, as different groups of people will have done the grading of each dataset.

Given these problems, we concluded that there is a unifying factor missing from this area of research, and that by providing an easy to use framework that allows for easy implementation, training and evaluation, we hope to greatly improve the quality of both existing research, and future developments in this area.

3 Framework Structure

KAREN was designed with two main objectives: easy implementation of both models and datasets for the user, as well an in-built mechanism that could filter which models can be run on some datasets (what we have previously called *compatibility*).

Definition 3.1 (Compatibility) *For a model M with a set of data requirements S_M and a dataset D with a set of available data S_D, we say that M is compatible with D iif. $S_M \subset S_D$.*

This notion of compatibility is the main property that allows KAREN to become scalable without having to worry about keeping track of whether or not a particular model or dataset are compatible. KAREN automatically validates if the set of model properties are a subset of the dataset's available data. This restriction on the models causes two restrictions to happen:

1. A model cannot access any other data that is not explicit in its data requirements. This is a safety measure to guarantee the consistency of the available models.

2. A model that requires a specific type of data will not be able to run on the vast majority of the datasets if they do not also contain this information. This effectively has to do with the available datasets being limited and it provides a new work direction – allowing the community to contribute and extend these datasets to hold more information from the basic data that is available. One could, for example, develop sentiment analysis for previous datasets to allow others models that use this data to be run on a previously unavailable dataset.

Although this concept might seem limiting at first, we have come to realize that its benefits are way more valuable. After we successfully developed the core of the framework, implementing all the models was very fast and easy to do, since all the validations were already handled by KAREN and the model just had to select the type of data that it wanted to use.

As it is seen in Figure 1, there are 4 main pieces that build KAREN: model, dataset, toolkit and training.

Training is carried out on our default training cycle, which we have used for all the results that are available in section 4. Due to the usage of different datasets and the nature of the task, we consider the task to be, at all times, a **Multi Class Classification Task**, even if the dataset is made for binary classification. This training cycle also handles the split of the datasets into train, dev and test datasets, hence allowing the users to evaluate the bias and variance of models easily. In addition, KAREN handles and manages all memory that needs to be transferred to the GPU to make use of GPU acceleration, another thing the user does not have to worry about. The training loop automatically saves the model with the best validation accuracy to be run on the testing data. After training, the best saved model is used to test the model against the test dataset, yielding its **accuracy, precision, recall and F1-score**. Although the last 3 metrics are only for binary classification, it is possible to do a class-wise analysis and select the relevant class for our analysis (the one that yields the results for hate speech).

Toolkit is where KAREN provides its NLP related tools. At the time of writing, this toolkit consists of three different types of Glove embeddings, each trained on a different dataset. This toolkit can be extended to add any other relevant tools. Due to the size of the pretrained embeddings we use, we present a default download method. On the first time using a specific type of embeddings, KAREN will download it to a local folder, unzip it and convert to binary format for reduced storage size.

Models are the models that are available in KAREN. Each model is required to extend the *BaseModel* class that defines its interface. Defining a model is as easy as defining its *init*, *forward* and *data requirements* functions, so implementing a full model from scratch is a very short and simple process.

Datasets are the datasets which models will be trained on. Similar to models, datasets need to extend a *BaseDataset* class and override the corresponding selection of methods for it to be able to

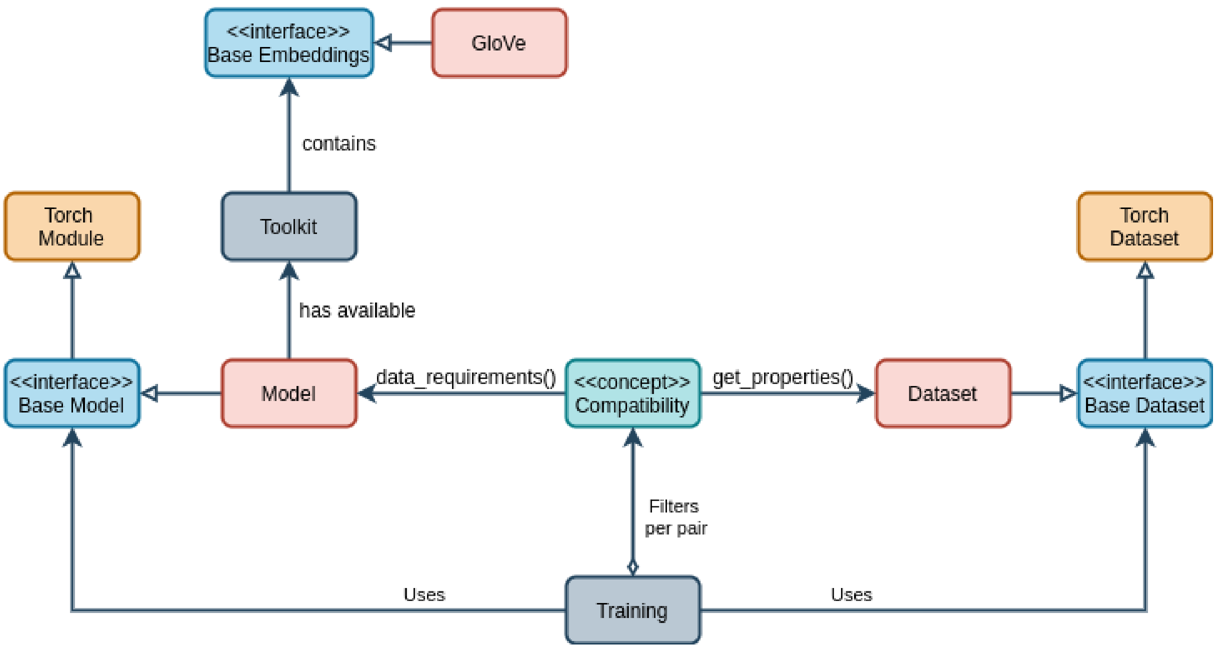

Figure 1: KAREN framework architecture

```python
@RegisterModel()
class ModelName(BaseModel):
    """
        This is a template file of a model implementation
    """

    def __init__(self):
        super(ModelName, self).__init__()

    def forward(self, data):
        pass

    @staticmethod
    def add_required_arguments(parser):
        pass

    @staticmethod
    def make_model(args):
        pass

    @staticmethod
    def data_requirements():
        pass
```

Figure 2: The interface of a model in KAREN. The user only needs to write this set of functions to easily and quickly implement a model in the framework.

run on the training script. Implementing datasets is a little more time consuming than models, as it must include some more information. One problem that was raised when developing was the way data was being sent to the models. Not all data can be transformed into a tensor, and some of the data is independent of the batch that is currently being run (for example, user metadata). For this, we designed datasets to output 3 types of data: tensor-like data, non-tensor data, always present data. We managed to shift the complex logic to the training instead of complicating the dataset, and implementing datasets is simple enough for anyone to quickly contribute theirs. We also implement a download logic for the datasets.

One thing that is not shown in Figure 1 is the run script. This script eases the manipulation of the framework by connecting all of its components together. Models and datasets both have a set of arguments that are customizable, and this script eases that configuration by allowing the users to run the framework from the command line by adding a set of parameters to a parser. To run, for example, a transformer model with 1 layer and dropout of 0.15 on the HateXPlain dataset using embeddings trained on twitter of size 100, the user simply needs to input `python3 run.py --model Transformer --dataset HateXPlain --transformer-n-layers 2 --dropout 0.15 --embeddings twitterglove --embedding-dim 100` which highly simplifies the testing method.

4 Results

In this section we will discuss the obtained results. Our measures are shown in Table 1, Table 3 and Table 3. We implemented three different datasets to serve as the baseline for our solution: HateXPlain (Mathew et al., 2020), HSAOL (Davidson et al., 2017b) and Ethos (Mollas et al., 2020). The

Model	Accuracy	Precision	Recall	F1
Bert	**0.689**	**0.777**	**0.752**	**0.764**
CNN	0.613	0.711	0.69	0.7
Softmax Regression	0.366	0.298	0.087	0.135
RNN	0.55	0.684	0.654	0.669
BiLSTM	0.59	0.662	0.764	0.71
NetLSTM	0.61	0.678	0.76	0.716
GRU	0.609	0.666	0.777	0.72
Transformer (1 layer)	0.486	0.495	0.6	0.543
Transformer (2 layers)	0.532	0.551	0.732	0.629
CharCNN	0.552	0.65	0.61	0.63
AngryBERT	**0.649**	**0.736**	**0.695**	**0.764**
DistilBERT	**0.646**	**0.766**	**0.704**	**0.734**
RNN + GloVe	0.546	0.59	0.779	0.672
CNN + GloVe	**0.644**	**0.69**	**0.767**	**0.726**
BiLSTM + GloVe	**0.637**	**0.677**	**0.781**	**0.73**
GRU + GloVe	**0.64**	**0.699**	**0.736**	**0.717**
NetLSTM + GloVe	0.616	0.679	0.756	0.715
Transformer (1 layer) + GloVe	0.564	0.581	0.785	0.668
Transformer (2 layers) + GloVe	0.572	0.751	0.609	0.672
CharCNN + Glove	0.573	0.631	0.753	0.686
AngryBERT + Glove	**0.660**	**0.75**	**0.771**	**0.76**
UNet + Glove	0.602	0.714	0.670	0.691
UNet	0.548	0.646	0.670	0.657
VDCNN	0.552	0.694	0.653	0.673
VDCNN + Glove	0.601	0.681	0.694	0.688

Table 1: Results for the HateXPlain dataset. Precision, Recall and F1 score are showing for the hate speech label

Model	Accuracy	Precision	Recall	F1
CNN	0.644	0.639	0.548	0.590
Softmax Regression	0.467	0.465	0.952	0.625
RNN	0.522	0.4	0.048	0.085
BiLSTM	0.589	0.619	0.310	0.413
NetLSTM	0.544	0.6	0.071	0.128
GRU	0.6	0.588	0.476	0.526
Transformer (1 layer)	0.556	0.6	0.143	0.231
Transformer (2 layers)	0.478	0.468	0.881	0.612
UNet	0.478	0.471	0.976	0.636
DistilBERT	**0.744**	**0.757**	**0.667**	**0.709**
AngryBERT	**0.711**	**0.674**	**0.738**	**0.705**
RNN + GloVe	0.544	0.514	0.429	0.468
CNN + GloVe	0.644	0.639	0.548	0.590
BiLSTM + GloVe	0.656	0.628	0.643	0.635
GRU + GloVe	**0.711**	**0.674**	**0.738**	**0.705**
NetLSTM + GloVe	0.644	0.614	0.643	0.628
Transformer (1 layer) + GloVe	0.611	0.569	0.690	0.624
Transformer (2 layers) + GloVe	0.489	0.463	0.595	0.521
UNet + Glove	**0.722**	**0.707**	**0.690**	**0.699**
AngryBERT + Glove	**0.733**	**0.75**	**0.643**	**0.692**
Bert	**0.756**	**0.763**	**0.690**	**0.725**
CharCNN + GloVe	0.533	0	0	0
CharCNN + GloVe	0.533	0	0	0
VDCNN	0.5	0.459	0.405	0.430
VDCNN + Glove	**0.755**	**0.708**	**0.810**	**0.756**

Table 2: Results for the Ethos dataset.

Model	Accuracy	Precision	Recall	F1
CNN	**0.908**	**0.5**	**0.218**	**0.304**
Softmax Regression	0.757	0	0	0
RNN	0.865	0	0	0
BiLSTM	0.896	0	0	0
NetLSTM	0.897	0.377	0.168	0.233
GRU	0.904	0	0	0
Transformer (1 layer)	0.876	0.448	0.104	0.169
Transformer (2 layers)	0.887	0.4	0.192	0.26
UNet	0.897	0.475	0.224	0.304
DistilBERT	0.908	0.441	0.345	0.387
AngryBERT	0.908	0.3	0.024	0.044
RNN + GloVe	0.898	0	0	0
CNN + GloVe	**0.915**	**0.518**	**0.244**	**0.331**
BiLSTM + GloVe	0.906	0	0	0
GRU + GloVe	0.909	0	0	0
NetLSTM + GloVe	0.906	0.397	0.193	0.260
Transformer (1 layer) + GloVe	0.892	0.471	0.128	0.201
Transformer (2 layers) + GloVe	0.907	0.474	0.216	0.287
UNet + Glove	0.912	0.524	0.264	0.351
AngryBERT + Glove	0.913	0.385	0.12	0.183
Bert	**0.918**	**0.552**	**0.384**	**0.453**
VDCNN + Glove	0.598	0.655	0.709	0.681

Table 3: Results for the HSAOL dataset. Due to the dataset being quite unbalanced, our choice of using Precision, Recall and F1 score proves reliable.

datasets were designed to be used for the detection of online hate speech by using content from, for example, Twitter and Reddit, two popular social media websites where users communicate mostly via text.

We collect 4 types of data: accuracy, precision, recall and F1 score. To obtain the last three measures in a multi-class setting, we extract the metrics corresponding to the *hatespeech* class that is present in each dataset. We found that we needed to collect more metrics besides just accuracy since several of the datasets in this field show a high level of imbalance. For example, in the HSAOL dataset, the proportion of hatespeech in the dataset is around 10%. If we consider the binary classification problem – a sentence is either hatespeech or not – then for this dataset, a model that predicts just the not hate speech label already achieves an accuracy of 90%.

Since we are trying to measure a model's performance in detecting hate speech content, datasets that contain a low amount of hate speech labels might hide their poor performance detecting hate speech with a very high accuracy. As mentioned, this kind of behaviour is particularly prevalent in HSAOL, where most recurrent neural network (RNN) based models achieved very low F1 scores – indeed, many models got an F1 score of 0. Compared to these models, one that performed surprisingly well was CNN, a very small and lightweight model which still managed to characterize the different types of labels effectively.

It is easy to see that using BERT models for hate speech detection achieves the highest performance. BERT (Devlin et al., 2018) for sequence classification, which consists of a pre-trained BERT followed by a Feed Forward Network, is the best model across all datasets. It achieves the best accuracy, together with the best F1 score. This means that it performs better than all other models, in all cases. We also experimented with AngryBERT (primary only, since our datasets do not include sentiment detection labels yet), which uses a combination of BiLSTM with BERT in parallel followed by a trainable gate to give different contributions to each side of the computation. Its performance is lower than the default BERT still. We believe that this is related to the low amount of data available, which does not allow the BiLSTM structure to train as well, since for fine-tuning a BERT model we need to use a very low learning rate.

Transformer models did not perform as well as was expected from them. It is a current trend nowadays to use huge models by increasing the amount of parameters, but these very big models also require a very large amount of data. The datasets available for hate speech detection, however, are very small, so training very deep transformers yields very bad results. Shallow transformers, however, did not achieve the best performance overall. For example, running a transformer with 3 layers instead of 2 achieves a lower accuracy, as well as F1 score. Further work can be explored to improve these models. Similar results were found with CNN. Training with more convolution layers did not seem to increase performance, only decrease it.

We also analyse the usage of pretrained embeddings to increase training speed and model performance. For this task, we use the pretrained GloVe embeddings (Pennington et al., 2014), a very popular choice among researchers. For our results specifically, we use the glove embeddings collected from the Twitter crawl, but the other GloVe embeddings are available for use in KAREN. With the GloVe embeddings, a noticeable increase in performance is seen, with all models improving, even if just slightly. We recommend its usage to maximize the detection performance of models.

A recent paper in the text classification field yields VDCNN (Conneau et al., 2016), where the VD stands for "very deep". Since hate speech detection is a direct subset of the text classification/sentiment analysis field, we could apply the model directly to the datasets. However, due to the many convolutions using a stride greater than 1, the model could only run on the HateExplain and Ethos datasets, as the max sentence length of the other datasets was not long enough. The results of this model are pretty good, but they are clearly inferior to Bert.

Besides using NLP techniques, we also experimented with some models designed for image classification, as it is the case of UNet (Ronneberger et al., 2015). UNet is a popular image classification model, first developed for the image segmentation of biomedical images. Recently, UNet has been adapted to work in the audio denoising setting, see (Michelashvili and Wolf, 2019). The implementation of UNet we used actually takes this adaptation of UNet for audio denoising, which uses 1D convolutions instead of 2D convolutions, and further adapts it for sentence classification. After passing the batch through the embeddings layer, we need to convert the final dimension to a power of two so that the dimension of the downsampling and upsampling convolutional layers match. After passing through UNet, the output needs to be transformed using a linear layer to get the required number of output classes. The results of UNet were fairly middle-of-the-pack, not particularly low or high. Similar to the transformer and many of the other models, it seems that the datasets are too small for the model to fit well.

5 Conclusion

Overall, we consider KAREN to fulfill its requirements. We have simplified the process of implementing and testing models and datasets by providing an unified architecture that allows researchers to quickly try out new ideas on different datasets. We have showcased our work to some acquaintances and we got good feedback thanks to it being so easy to use. Personally, all our members found that extending the framework to adapt to any new model was a very simple and efficient process, even though we were expecting to run into design barriers. We find the final results to be very satisfactory and the unification and simplification of the model implementation process is a solid contribution to this research topic.

Regarding models, we have concluded that using pretrained models like BERT yield the best results. Also, using shallow models works best due to the available datasets not having enough data to train deep models. We have also analysed the dataset imbalance problem and how it can affect the results of a model, thus attaching importance to using other metrics such as precision, recall and F1 score. We also analyse the impact of using pretrained word embeddings, which we conclude to increase the performance of all models that use it, even if just a little.

Finally, we welcome any contribution. We think it would be relevant for more people to contribute to this repository and fill it in with more papers from this area, so that we can attract more people to use this contribution and become a standard across research. We would expect this work to be used in future research that analyses the performance increase from using meta-data in the detection, since it would allow for many techniques to be quickly experimented on.

Acknowledgments

We would like to thank Professor Zhiyuan Liu for his feedback which was very insightful and was of great help on choosing the direction of our problem. We would like to also thank Yuan Yao for his help and guidance which were very valuable and allowed us to quickly grasp the knowledge of this area.

References

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2021. A deep dive into multilingual hate speech classification. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, pages 423–439. Springer International Publishing.

Md. Rabiul Awal, Rui Cao, Roy Ka-Wei Lee, and Sandra Mitrovic. 2021. Angrybert: Joint learning target and emotion for hate speech detection. *CoRR*, abs/2103.11800.

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, and Michael Granitzer. 2020. Hatebert: Retraining BERT for abusive language detection in english. *CoRR*, abs/2010.12472.

Yukuo Cen, Zhenyu Hou, Yan Wang, Qibin Chen, Yizhen Luo, Xingcheng Yao, Aohan Zeng, Shiguang Guo, Peng Zhang, Guohao Dai, et al. 2021. Cogdl: An extensive toolkit for deep learning on graphs. *arXiv preprint arXiv:2103.00959*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017a. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51:1 – 30.

Aditya Gaydhani, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv preprint arXiv:1809.08651*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Michael Michelashvili and Lior Wolf. 2019. Speech denoising by accumulating per-frequency modeling fluctuations. *arXiv e-prints*, pages arXiv–1904.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*.

John T. Nockleby. 2000. Hate speech. In *Encyclopedia of the American constitution*, pages 1277–1279.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Xiaofeng Wang, Matthew S. Gerber, and Donald E. Brown. 2012. Automatic crime prediction using events extracted from twitter posts. In *Social Computing, Behavioral - Cultural Modeling and Prediction - 5th International Conference, SBP 2012, College Park, MD, USA, April 3-5, 2012. Proceedings*, volume 7227 of *Lecture Notes in Computer Science*, pages 231–238. Springer.

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, Austin, Texas. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions.

Z. Zhang. 2017. Hate speech detection using a convolution-lstm based deep neural network.

Z. Zhang, D. Robinson, and J. Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *ESWC*.