GUIR at SemEval-2020 Task 12:
Domain-Tuned Contextualized Models for Offensive Language Detection

Sajad Sotudeh[*][†]**, Tong Xiang**[*][‡]**, Hao-Ren Yao**[†]**, Sean MacAvaney**[†]**,**
Eugene Yang[†]**, Nazli Goharian**[†]**, Ophir Frieder**[†]
Information Retrieval Lab
Georgetown University, USA
[†]`{firstname}@ir.cs.georgetown.edu`
[‡]`tx39@georgetown.edu`

Abstract

Offensive language detection is an important and challenging task in natural language processing. We present our submissions to the OffensEval 2020 shared task, which includes three English sub-tasks: identifying the presence of offensive language (Sub-task A), identifying the presence of target in offensive language (Sub-task B), and identifying the categories of the target (Sub-task C). Our experiments explore using a domain-tuned contextualized language model (namely, BERT) for this task. We also experiment with different components and configurations (e.g., a multi-view SVM) stacked upon BERT models for specific sub-tasks. Our submissions achieve F1 scores of 91.7% in Sub-task A, 66.5% in Sub-task B, and 63.2% in Sub-task C. We perform an ablation study which reveals that domain tuning considerably improves the classification performance. Furthermore, error analysis shows common misclassification errors made by our model and outlines research directions for future.

1 Introduction

The rapid development of user-generated content in social media has given millions of people the ability to easily share their ideas with each other. While users can publicly communicate their beliefs with others, their published content may be offensive to other individuals or groups. Since offensive speech can jeopardize others' ability to express themselves, many social media platforms restrict the type of acceptable content on their platforms. Manually detecting such content is expensive and time-consuming. Thus, automatic detection of these behaviors in social media has attracted researchers' attention. Although this topic has been explored in prior works (Yao, 2019; Rizos et al., 2019; Zhang et al., 2018; Nobata et al., 2016), the task of offensive language detection still remains challenging. The OffensEval 2020 shared task (Zampieri et al., 2020) aims to encourage continued work in this area, and we progress the study of offensive language detection through our participation in this shared task utilizing contextualized language models.

OffensEval 2020 evaluates various aspects of offensive language following the scheme of the OLID dataset (Zampieri et al., 2019a), including identifying the presence of offensive language (Sub-task A), identifying whether the offensive language is targeted (Sub-task B), and identifying whether the target of the offensive language is an individual, group, or something else (Sub-task C). The task extends the prior work by introducing a new data collection from Twitter that spans multiple languages.

Our experiments explore various versions of the BERT model (Bidirectional Encoder Representations from Transformers (Devlin et al., 2019)) tuned for offensive language identification. As previous studies have mentioned, dealing with social media content can be challenging because it is often short and noisy (Qian et al., 2018). To alleviate this problem, we first fine-tune a BERT model on a large amount of unlabelled data gathered from Twitter. We pre-process the tweets by splitting hashtags and replacing emoji with textual descriptions. We also explore an ensemble learning method (i.e., a multi-view SVM model akin to (MacAvaney et al., 2019)) which combines different word n-gram features of the input

---

[*] equal contribution

text as views and predict the output based on the combination of these views. We report competitive performance of these models, which achieve macro F1 scores of 91.7%, 66.5%, and 63.2% on Sub-tasks A, B and, C, ranking 5th, 6th, and 11th for each sub-task out of 85, 43, 39 participated teams respectively.

Our contributions are as follows: 1) we present variations of BERT model tuned for different aspects of offensive language detection; 2) we report competitive results of our models with detailed comparisons; 3) we perform an ablation study to examine the effect of different components in the proposed systems; and 4) we conduct an error analysis to provide insights into how our models perform the classification, and where to focus in future work.

2   Related work

The research of automatic offensive language detection has gained attention in the past decade. Most prior work in the domain employs the supervised learning paradigm (Schmidt and Wiegand, 2017). Traditional models often made use of rule-based methods, such as template-based strategy (Warner and Hirschberg, 2012) or pre-defined black-lists (Xiang et al., 2012). Aside of methods, a widely used category of features are surface-level features, e.g., n-gram features (Schmidt and Wiegand, 2017). These features are often highly predictive, and can be easily combined with other approaches. Others explored the combination of n-gram features with part-of-speech (Nobata et al., 2016) as well as dependency information (Chen et al., 2012) for offensive language detection. These approaches leverage prior linguistic knowledge in order to generate features. However, the generated features are usually derived from pre-existing natural language processing systems, which could lead to the propagation of errors in the models (Zeng et al., 2014). These features are usually combined with classical machine learning classifiers such as SVM (Yin et al., 2009; Malmasi and Zampieri, 2018) and Logistic Regression (Waseem and Hovy, 2016; Davidson et al., 2017). Others have explored using a multi-view learning paradigm (Zhao et al., 2017) with an ensemble classifier (MacAvaney et al., 2019).

Deep learning techniques have also been shown to be effective for offensive language detection (Badjatiya et al., 2017; Park and Fung, 2017; Liu et al., 2019a), with approaches such as Convolution Neural Networks (CNN) (Gambäck and Sikdar, 2017) and Long Short-Term Memory (LSTM) Networks (Pitsilis et al., 2018). More recently, pre-trained transformer-based networks, such as BERT (Devlin et al., 2019) have shown great advantages in learning context-sensitive word representations. In OffensEval 2019, BERT-based and ensemble methods were the most effective approaches (Zampieri et al., 2019b; Liu et al., 2019b; Han et al., 2019).

3   Methodology

In our experiments, we utilize a pre-trained contextualized language model, namely BERT (Devlin et al., 2019), to identify the offensive language. Also, we explore techniques for pre-processing, contextualized language modeling, and model outputs ensembling.

Pre-processing. Given the unique conventions of language on Twitter, we explore several tokenization pre-processing techniques to enable the downstream models to encode information more effectively. Since the length of a tweet is limited to 280 characters, emoji are often used to convey emotions and tones efficiently. To address differences in user preferences among users, we replace emoji with a textual description of the icons in order to shorten the domain gap between the corpus and the tweets. We use the the mapping of emoji to English descriptions from the open source Python package `emoji`.[1]

Hashtags are another common convention in tweets. They are used to describe topics related to certain tweets. They often consist of several words concatenated together, such as *#VoteRedSaveAmerica* and *#trumptrain*. Since hashtags do not contain whitespaces at word boundaries, additional logic is required for segmentation. We utilize the open source `wordsegment` [2] library to obtain the boundaries and further construct the original textual tokens.

The OffensEval 2020 dataset replaces all usernames with a *@USER* placeholder. This can result in some long strings of redundant and repetitive placeholders because tweets are often prefixed with

---

[1] `https://github.com/carpedm20/emoji`
[2] `https://github.com/grantjenks/python-wordsegment`

numerous users. We tokenize using the `nltk` tweet tokenizer (Bird et al., 2009) and drop *@USER* tokens if repeated more than three times consecutively to avoid redundant information (similar to Liu et al. (2019b)). Furthermore, the token *URL*, which is the artificial placeholder for any URL encountered in tweets, is also replaced with *http* to match the vocabulary in the pre-trained embeddings.

Contextualized language modeling. We utilize the BERT contextualized language model (Devlin et al., 2019). Since there are language differences between the formal text that BERT is trained on (i.e., Wikipedia and books) and social media posts (i.e., tweets), we first tune the model to the particular domain, akin to the domain pre-training approach described in (Gururangan et al., 2020). This is accomplished by taking the original model and continuing to train the masked language model and next sentence prediction objectives using a large amount of unlabeled tweets. Note that we do not extend the vocabulary; we rely on the model's original WordPiece tokens.

We then fine-tune the model for identifying offensive language utilizing labeled training data. Since the task is sequence classification, we utilize the classification mechanism of the model (i.e., a linear layer on top of BERT's classification token). We train the model minimizing the cross-entropy loss, as compared to the gold training labels. We train a separate model for each sub-task during experiments.

Model outputs ensembling. As mentioned in Section 2, ensemble approaches are often beneficial for offensive language detection and related tasks. In this work, we extend the multi-view SVM approach from (MacAvaney et al., 2019) with the addition of features from the contextualized language model classifier. Specifically, linear SVM classifiers (view-classifiers) using various n-gram ranges [3] are first trained for each sub-task in addition to the BERT-based classifier. Then, the outputs of the view-classifiers (probability output from SVM and sigmoid output from BERT) are concatenated as a feature vector for a final linear SVM classifier (the meta-classifier). For the SVM view-classifiers, we explore using both L1 and L2 regularization.

4 Experiment

In this section, we present settings, results and analysis for our experiments. We first give a brief introduction of the dataset used for training and evaluation. Then we show our experimental settings, and perform a comprehensive analysis including experimental results analysis, ablation analysis, and error analysis[4] over our models' results.

4.1 Data

Zampieri et al. (2019a) introduced the Offensive Language Identification Dataset (OLID), a large-scale dataset of English tweets constructed by searching for specific keywords that may include offensive words on Twitter. They developed a hierarchical annotation schema that determines: 1) if the tweet is offensive (OFF) or non-offensive (NOT); 2) if an OFF tweet is targeted (TIN), or untargeted (UNT); and 3) if a TIN tweet is targeted toward individual (IND), group (GRP), or others (OTH). We refer the readers to Zampieri et al. (2019a) for more details about the dataset characteristics.

The OffensEval 2020 task (Zampieri et al., 2020) offered a multilingual offensive language detection dataset. We participate in the three sub-tasks under the English track. For this track, the training set from OLID is used as training data and the test set from OLID is treated as development data. The annotation for newly annotated test data follows the same hierarchical schema as OLID, which was used during the evaluation phase. The task also provides a distant dataset (Rosenthal et al., 2020) including over *9M* tweets with predicated labels from an ensemble of classifiers. For our experiments, we disregard the labels and only make use of the text as pre-training data for Twitter domain adaptation.

4.2 Experimental settings

For domain pre-training (as described in Section 3), we utilize the tweets from the distant dataset provided by the shared task (disregarding labels). We tune the `BERT-Base` model using these data via training on the language modeling task and the default hyper-parameters provided by the BERT authors for training

---

[3]While we experimented with different ranges of n-grams, 6-gram feature was the optimal one so we fixed n-gram at 6.

[4]Our error analysis contains tweet examples and words that are offensive in nature.

(a) Sub-task A						
Model	TKN	TWD	LT	mSVM	Dev (F1)	Test (F1)
BERT	-	-	-	-	0.805	0.915
BERT	✓	-	-	-	0.806	0.904
BERT	✓	✓	-	L1	0.822	0.915
BERT	✓	✓	-	-	0.823	0.911
BERT	-	✓	✓	-	**0.828**	* **0.917**
Top OffensEval 2019 / 2020					0.829	0.922
Median OffensEval 2019 / 2020					0.739	0.909
Mean OffensEval 2019 / 2020					-	0.871
Min OffensEval 2019 / 2020					0.171	0.073
Std OffensEval 2019 / 2020					-	0.127

(b) Sub-task B						
Model	TKN	TWD	LT	mSVM	Dev (F1)	Test (F1)
BERT	✓	-	-	-	0.718	0.651
BERT	-	-	-	-	0.745	0.287
BERT	-	✓	✓	-	0.761	**0.701**
BERT	✓	✓	-	-	0.815	0.648
BERT	✓	✓	-	L2	**0.843**	* 0.665
Top OffensEval 2019 / 2020					0.755	0.746
Median OffensEval 2019 / 2020					0.638	0.569
Mean OffensEval 2019 / 2020					-	0.555
Min OffensEval 2019 / 2020					0.121	0.278
Std OffensEval 2019 / 2020					-	0.120

(c) Sub-task C						
Model	TKN	TWD	LT	mSVM	Dev (F1)	Test (F1)
mSVM	-	-	-	L1	0.488	0.388
BERT	-	-	-	-	0.502	0.536
BERT	✓	✓	-	L2	0.602	**0.654**
BERT	✓	✓	-	L1	0.625	0.649
BERT	✓	✓	-	-	**0.631**	* 0.632
Top OffensEval 2019 / 2020					0.660	0.715
Median OffensEval 2019 / 2020					0.515	0.581
Mean OffensEval 2019 / 2020					-	0.560
Min OffensEval 2019 / 2020					0.090	0.057
Std OffensEval 2019 / 2020					-	0.121

(d) Abbreviation details	
Description	Abbreviation
Utilizing distant dataset for domain tuning	TWD
Additional tokenization pre-processing	TKN
Utilizing loss value as early stop signal	LT
Multi-view SVM with L1 regularization	mSVM-L1
Multi-view SVM with L2 regularization	mSVM-L2

Table 1: (a), (b) and (c) describe macro-averaged F1 scores for each of our 5 submitted models on the development (Dev) and test set (Test). (d) defines abbreviations used for the experimental settings. We bold the highest scores in our experiments. * indicates our official submissions to the shared task. Top, median, mean, minimum (Min) and standard derivation (Std) values for OffensEval 2019 and 2020 (systems differ) are from Zampieri et al. (2019b, 2020).

(learning rate: 2×10^{-5}, masking rate: 0.15, maximum sequence length: 128). For better reproducibility, we use the authors' original implementation for this tuning.[5]

To tune the BERT model for the specific task, we utilize the OLID training data. The input sentences are directly tokenized into subword units by the BERT WordPiece tokenizer. Additionally, each input sentence is concatenated with a special token `[CLS]` at the beginning. Since tweets have a character length limitation, we define the maximum sequence length to be 256 tokens. Some of our experiments also make use of additional tokenization pre-processing techniques described in Section 3. For the task tuning, we utilize the `transformers` library (Wolf et al., 2019).

We tune hyperparameters based on the F1 score on the development set using two approaches. First, we try a simple approach in which the development F1 performance is evaluated after each training epoch (1 to 10). Second, we explore utilizing the training loss as an early stopping signal. Once the loss value reaches a pre-defined range, the model is evaluated on the development set.

We ensemble n-gram SVM view classifiers with the BERT models. The meta classifier consumes the probabilistic prediction from each view classifier to provide the final prediction. L1 and L2 regularization strategies are applied to all SVM models with the inverse regularization penalty C to be 10^{-5}.

4.3 Results and discussion

Table 1 (a-c) shows the performance of our 5 submitted models selected based on their development set performance. Table 1 (d) defines abbreviations for the experimental settings used in this section.

The tuned BERT-TWD-LT model achieves the best performance among our models both on development and test sets for Sub-task A, showing that adaptation of `BERT-Base` model on Twitter data can substantially boost the performance, and the loss tuning approach can further enhance the model performance. While surpassing official median scores, tuned BERT-TWD-LT model lags behind the top official score by 0.5% on F1 score, leading to rank 5 on this sub-task. For Sub-task B, we also observe that the BERT-TWD-LT model outperforms other models and official results on the test set, but performs

---
[5]https://github.com/google-research/bert

	Tweet	Prediction	Gold
Task A	*(A1)* @USER He deserve the worst of this all	OFF	NOT
	(A2) @USER @USER Dehumanize? He barely has a reflection of human	NOT	OFF
Task B	*(B1)* @USER @USER @USER I'm just checking ... is it actually illegal to have an ass that perfect???*	UNT	TIN
	(B2) Then to top it completely the f*** off my trainer was my boyfriend...*	TIN	UNT
Task C	*(C1)* Can't and won't f***ing deal with LIARS. Goodbye.*	GRP	IND
	(C2) Honestly they're not even pretty and the music sucks....*	GRP	OTH
	(C3) "The Democrats created the KKK" yeah my ni*** I'm gonna need you to pick up...*	OTH	IND

Table 2: Examples of misclassified Tweets made by our model for each sub-task.

worse than the mSVM-L2 + BERT-TKN-TWD on the development set, and thus was not our official scored submission. This discrepancy in performance suggests different distributional characteristics between the two sets on Sub-task B. For Sub-task C, the mSVM-L2 + BERT-TKN-TWD outperforms our other systems on test set, demonstrating that utilizing an ensemble approach can be effective. We observed that named entities are quite important on Sub-task C. While mSVM performs reasonably well at capturing named entities, when it is further combined with BERT-TKN-TWD it is able to capture hidden relationships among tweet tokens. This leads to a considerable boost in F1 score. Since this model under-performed on the development set, it was not our official submission.

Ablation analysis. To gain a sense of how different components aid our model at performing the classification tasks, we perform an ablation study for each sub-task. When comparing models' performance in Table 1, we see that among the given components, domain tuning (TWD) yields consistent improvements across all sub-tasks, justifying that training vanilla `Bert-Base` on task-related data improves the performance significantly. Interestingly, it does not hold for tokenization approach (TKN). while it was a significant component of the previous top system (Liu et al., 2019b). Multi-view ensemble approach (mSVM) does not provide much improvement, although it achieves the best on the test set of Sub-task C. In Sub-task A and B, it is still behind the top scores. This might be because named entities play a crucial role in the identification of offense targets (i.e., Sub-task C).

Error analysis. To understand the limitation and qualities of the models, we qualitatively analyze the predictions from our best-performing models on several examples as shown in Table 2. By investigating the misclassified cases of each sub-task, we identified the following qualities of the misclassification examples. 1) *Annotation issues (A1, B1, C1)*: There are tweets with labels that are not in line with interpretation of the annotation guidelines. For instance, cases such as B1 while contain profanity, they do not seem to be targeted toward certain group/individual. 2) *Absurdity (B2):* Social media texts can often be obscure. As such, comprehending the tweet is not only hard for the model but in some cases, humans also have trouble understanding it. Tweets like B2 can be considered offensive due to the profanity, but do not appear to contain a threat or insult and thus should not be considered targeted. 3) *Sarcasm/Metaphor (A2):* as also discussed in prior works (MacAvaney et al., 2019), tweets that contain high levels of sarcasm or metaphor are hard to be picked up by predictive models; 4) *Multi-targets: (C3)* For cases in which multiple offense targets have been mentioned, it appears that the model has difficulty in picking up the true offense target.

5 Conclusions

In this study, we investigated three English sub-tasks of OffensEval 2020: 1) Offensive language identification; 2) Detection if the language is targeted; and 3) Identification of the target. Specifically, we explored fine-tuning BERT model with different configurations for each sub-task. We also investigated an ensemble learning method, multi-view SVM (i.e., mSVM) model, and further combined it with BERT models to improve model performance. Our experiments demonstrate the efficacy of our approaches. Our ablation study revealed that adaptation of BERT model to task-specific data can significantly improve the classification results. Furthermore, we conducted an error analysis over the predicted labels and identified 4 common errors which can be good directions for future work.

Acknowledgements

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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*. IEEE.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Jiahui Han, Shengtan Wu, and Xinyu Liu. 2019. jhan014 at SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Han Liu, Pete Burnap, Wafa Alorainy, and Matthew L. Williams. 2019a. Fuzzy Multi-task Learning for Hate Speech Type Identification. In *The World Wide Web Conference*.

Ping Liu, Wen Li, and Liang Zou. 2019b. NULI at SemEval-2019 Task 6: transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8).

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

Chikashi Nobata, Joel R. Tetreault, Achint Oommen Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*.

Ji Ho Park and Pascale Fung. 2017. One-step and Two-step Classification for Abusive Language Detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Jing Qian, Mai ElSherief, Elizabeth M. Belding-Royer, and William Yang Wang. 2018. Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Georgios Rizos, Konstantin Hemker, and Björn W. Schuller. 2019. Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. *arXiv preprint arXiv:2004.14454*.

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*.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*.

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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*.

Mengfan Yao. 2019. Robust Detection of Cyberbullying in Social Media. In *Companion Proceedings of The 2019 World Wide Web Conference*.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2:1–7.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *arXiv preprint arXiv:2006.07235*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*.

Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *ESWC*.

Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. 2017. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54.