IITP-AINLPML at SemEval-2020 Task 12: Offensive Tweet Identification and Target Categorization in a Multitask Environment

Soumitra Ghosh, Asif Ekbal, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
{1821cs05, asif, pb}@iitp.ac.in

Abstract

In this paper, we describe the participation of IITP-AINLPML team in the SemEval-2020 Shared Task 12 on Offensive Language Identification and Target Categorization in English Twitter data. Our proposed model learns to extract textual features using a BiGRU-based deep neural network supported by a Hierarchical Attention architecture to focus on the most relevant areas in the text. We leverage the effectiveness of multitask learning while building our models for sub-task A and B. We do necessary undersampling of the over-represented classes in the sub-tasks A and C. During training, we consider a threshold of 0.5 as the separation margin between the instances belonging to classes OFF and NOT in sub-task A and UNT and TIN in sub-task B. For sub-task C, the class corresponding to the maximum score among the given confidence scores of the classes (IND, GRP and OTH) is considered as the final label for an instance. Our proposed model obtains the macro F1-scores of 90.95%, 55.69% and 63.88% in sub-task A, B and C, respectively.

1 Introduction

With the advancement of technology and gaining popularity of several social media platforms, there has been a manifold increase in the number of users active in such channels. More often than not, these users tend to misuse their power of freedom of speech and violate acceptable usage policies of most of the forums. This has necessitated detecting any offensive or obscene posts, comments, images, etc. and prevent further dissemination of the same to curtail its effect on social media. One promising solution is to develop automated tools for identification of offensive language using various Natural Language Processing (NLP) and Machine Learning (ML) techniques. In the past few years, many shared tasks and seminars were introduced to address this problem and provide relevant annotated data collected from various social media. Some of the related tasks are: ALW[1] (related to Abusive Language Identification), TRAC 1 2018 [2] related to Aggression Identification (Kumar et al., 2018), GermEval Task 2 [3], HatEval 2019 [4](i Orts, 2019), HASOC 2019 [5] and OffensEval 2019 Task 6 [6] related to Offensive Language Identification (Zampieri et al., 2019b). Recent works have considered categorizing hate speech problem into sub-classes like abusive, aggressive, or offensive speech. Such categorization of social media posts help law-enforcement agencies with the surveillance of social media.

1.1 Problem definition

Similar to OffensEval-2019 shared task, the OffensEval-2020 shared task-organized at SemEval-2020 involves detection of Offensive Language and Target Identification from Tweets in English. OffensEval-2020 (Zampieri et al., 2020) offers a multilingual *(Arabic, Danish, English, Greek, Turkish)* dataset

[1] https://sites.google.com/site/abusivelanguageworkshop2017/
[2] https://sites.google.com/view/trac1/shared-task
[3] https://projects.fzai.h-da.de/iggsa/
[4] https://competitions.codalab.org/competitions/19935
[5] https://hasocfire.github.io/hasoc/2019/index.html
[6] https://sites.google.com/site/offensevalsharedtask/offenseval2019

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1983–1991
Barcelona, Spain (Online), December 12, 2020.

annotated following the hierarchical tagset introduced in OffensEval 2019. Based on the type and target of offences, the task is categorized into 3 sub-tasks:

- **Sub-task A: Offensive language identification**
  Here, the task is to differentiate between offensive and non-offensive tweets. The class labels in this sub-task are: *OFF* (Offensive) and *NOTOFF* (Non-Offensive). Any profanity, insults, violent remarks etc. qualifies as an OFF tweet. Rest all as NOTOFF.

- **Sub-task B: Automatic categorization of offense types**
  Once a tweet is known to be offensive, the next task is to detect the presence of any target entity to whom the offence is directed. The two classes involved are: *UNT* (Untargeted) and *TIN* (Targeted).

- **Sub-task C: Offense target identification**
  This task aims to find the target of offence with possible categories of *IND* (Individual), *GRP* (GROUP) and *OTH* (Others). Insulting or threat tweets are only considered in this category since such tweets are usually directed to some target.

The efficacy of our models relies heavily on our strategy of mapping confidence scores to the corresponding class labels in the training data. This enables us to propose a hierarchical attention-based BiGRU network that can be employed for all the 3 sub-tasks independently. The hierarchical attention mechanism not only focuses on the important words at the sentence level but also on the important sentences at the document level. We use pre-trained GloVe embedding to get the word representations and fine-tune them on the training data. During testing, we tweak the test instances by appending an additional weak label at the end of each instance (for sub-task A and B only) based on a heuristic measure to aid our models in making predictions. Our model attained F1-scores of 90.95% on detecting offensive language (sub-task A), 55.69% on categorising offence types (sub-task B), and 63.88% on identifying the target of offence (sub-task C).

The rest of this paper is organized as follows. In Section 2 we briefly discuss the related work in this area. In Section 3, we describe the dataset and various preprocessing steps on the dataset. We discuss the methodology in Section 4. In Section 5, we present results and give a brief analysis. In Section 6, we give our conclusion along with future works.

2 Related Work

For quite some time now, researchers have studied and reported their observations and results related to online misuse of social media platforms ((Razavi et al., 2010); (Warner and Hirschberg, 2012); (Ribeiro et al., 2017)). Misuse may take many forms like cyberbullying (Xu et al., 2012), trolling (Kwok and Wang, 2013) and offensive language (Cheng et al., 2017). (Gambäck and Sikdar, 2017) proposed a range of CNN-based deep neural models for classification of tweets into one of the following categories: sexism, racism, either (sexism or racism) and non-hate. Twitter Hate Speech text includes racism, sexism, both and a non-hate-speech classification system. ((Waseem et al., 2017; Waseem, 2016)) introduced a series of sub-tasks related to cyberbullying, online abuse and hate speech. ((Malmasi and Zampieri, 2017; Malmasi and Zampieri, 2018)) addressed the differences in general profanity and hate speech. (Wulczyn et al., 2017) introduced the Wikipedia Comments Corpora for building models to evaluate hate speech classifiers. Usage of word n-grams and sentiment lexicons were reported in (Davidson et al., 2017). In a comprehensive survey by (Schmidt and Wiegand, 2017), various linguistic, lexical, sentiment, surface features, etc. were identified that can be useful to build a classifier for detection of hate speech. A CNN and GRU based approach was proposed by (Zhang et al., 2018) for hate speech detection. An interesting work on predicting future hostility and its intensity looking at the current situation was studied in (Liu et al., 2018).

Though most of the works related to Offensive Language and Hate Speech has been in English, still there are few works in other languages as well. (Pavlopoulos et al., 2017) worked on a large dataset of Sports Comments in Greek and proposed several approaches to handle user content moderation using neural networks and sophisticated attention mechanism. (Mubarak et al., 2017) introduced a corpus

in Arabic consisting of obscene and offensive user comments and words in social media. (Fišer et al., 2017) explored malpractices in social networking sites in Slovenia mainly relating to the legal domain and subsequently introduced a dataset and annotation schema about such practices. (Su et al., 2017) proposed a system to detect and alter obscene and vulgar sentences in Chinese. The GermEval shared task (Schmidt and Wiegand, 2017) was introduced to facilitate research on the offensive language identification in microposts in the German language.

3 Data

3.1 Data Description

The OffensEval-2020 dataset (Rosenthal et al., 2020) (Zampieri et al., 2020) follows the similar hierarchical annotation scheme as used in creating Offensive Language Identification Dataset (OLID v1.0) (Zampieri et al., 2019a). The first level of classification is to distinguish between offensive (OFF) and non-offensive (NOT) instances which attribute to Sub-task A. Sub-task B consists of offensive instances only that needs to be classified based on types of insult: Targeted (TIN) or Untargeted (UNT). Sub-task C comprises of posts with offensive targeted tweets only which need further categorization to identify the target of offence: an individual (IND), a group (GRP) or others (OTH). The training set and test set contains 14,26,522 and 6,159 instances in English respectively. Table 1 shows the data distribution of instances over each sub-task.

Tasks	Train Data	Threshold Criteria	Actual	Undersampled	Test Data
Sub-task A	10,48,575	NOT if $Average < 0.45$	8,47,264	2,01,311	2,807
		OFF if $Average >= 0.45$	2,01,311	2,01,311	1,080
Sub-task B	1,88,974	TIN if $average < 0.5$	1,49,550	Undersampling not performed	850
		UNT if $average >= 0.5$	39,424		572
Sub-task C	1,88,973	IND if $average\_ind$ is max.	1,52,562	11,494	580
		GRP if $average\_grp$ is max.	24,917	11,494	190
		OTH if $average\_oth$ is max.	11,494	11,494	80

Table 1: Data distribution, Conversion criteria and Undersampling statistics

3.2 Conversion of Scores to Labels and Undersampling

The data provided in OffensEval-2020 (Zampieri et al., 2020) contains confidence scores instead of class labels. The scores are generated by unsupervised approaches which need to be represented as actual classes before training our models. Upon manual observation and analysis on a various set of samples of the data provided for all the sub-tasks, we consider threshold values of 0.45 and 0.5 for sub-task A and sub-task B respectively to distinguish instances of the distinct classes in each sub-task. For sub-task C, we compare the confidence scores of the 3 classes (IND, GRP and OTH) and consider that class label as the true class which has the maximum score.

The data provided in the task is highly skewed over the classes for all the sub-tasks. We do require undersampling of the over-represented classes in sub-task A and sub-task C to get a balanced distribution over the classes. We use the original distribution of sub-task B for our experiments. Table 1 depicts the various *Threshold Criteria* and the resulting data distribution (*Actual*) after mapping scores to labels. Distribution after undersampling of some classes is shown under column *Undersampled*.

3.3 Preprocessing

We perform a series of preprocessing of the tweets in the dataset before using them to build our models. We replace smileys (like : $-$) or :) is replaced by the word $Happy$) and emojis[7] by their meaning. We use

[7] https://pypi.org/project/emoji/

a dictionary of popularly used contractions[8] to replace them with their elongated forms. We also perform basic pre-processing steps like URLs and hashtags removal, extra blank spaces removal, conversion to lower case, the omission of non-ASCII characters etc.

4 Methodology

We use pre-trained GloVe[9] (Pennington et al., 2014) word embeddings to initialize our embedding layer and further fine-tune it on our training data while learning. The output from the embedding layer is passed through a BiGRU (256 units) layer which encodes the input representation to hidden representation. We leverage the effectiveness of Hierarchical attention (HATT) based Document Classification technique (Yang et al., 2016) to attend upon the instances more precisely. The attended vector is passed through two separate task-specific fully connected layers followed by their respective output layers (2 neurons with softmax activation for the classification task and 1 neuron with sigmoid activation for the regression task). For sub-task C, we train our model on the classification task only so there is a single output layer with 3 neurons (signifying 3 classes) with softmax activation. Rest of the architecture is similar to the models for sub-task A and B with an exception that the output from the attention layer is passed through a single fully connected layer before moving to the output layer. Figure 1 shows the general model architecture for our experiments.

4.1 Multitask Framework

We train two separate deep neural models for predicting the class labels NOT or OFF and UNT or TIN corresponding to sub-task A and sub-task B respectively. We jointly train our models (for sub-task A and B only) on the class labels along with the $stdev$ scores for each instance provided in the dataset. In other words, we train our classifier on a classification task (detecting class labels) and regression task (predicting the $stddev$ score) jointly. Learning this deviation values along with the labels enables our models to adapt to the necessary variations (to some extent) that may have crept into our framework while mapping confidence scores (generated using unsupervised methods) to class labels (using threshold criteria).

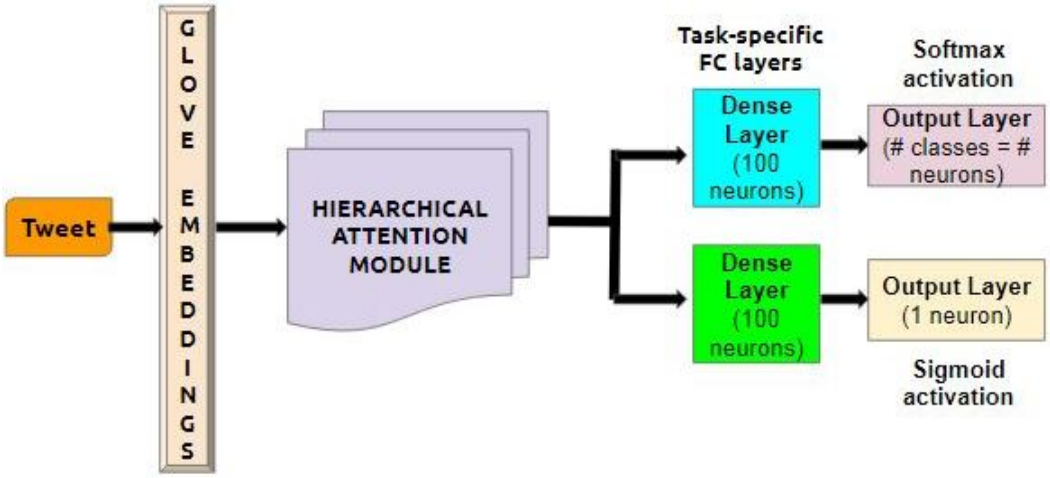

Figure 1: Multitask architecture for sub-task A and sub-task B. Sub-task C follows the same architecture when we omit the dense layer shown in green and its corresponding output layer (with 1 neuron)

4.2 Tweet classification using Hierarchical Attention

The intuition behind hierarchical attention (Yang et al., 2016) is to focus on important words in a sentence as well as focus on the important sentences that contribute most to the meaning of a document (tweet in our case). We tokenize a tweet instance into sentences and pass it through our pre-trained Keras[10]

[8] https://en.wikipedia.org/wiki/Wikipedia:List_of_English_contractions
[9] GloVe: http://nlp.stanford.edu/data/wordvecs/glove.840B.300d.zip
[10] https://keras.io/

(Chollet and others, 2015) embedding layer. Each embedded sentence representation is passed through an independent BiGRU layer (256 units). Each encoded output from a BiGRU layer is passed through an attention layer to focus on the important words in each sentence and a sentence vector is returned. All such sentence vectors of a tweet are further passed through another BiGRU layer (256 units) followed by an Attention layer to result in a document vector. The resultant document vector is passed through a linear layer(s) and finally the output layer. Figure 2 depicts the architecture of the HATT module.

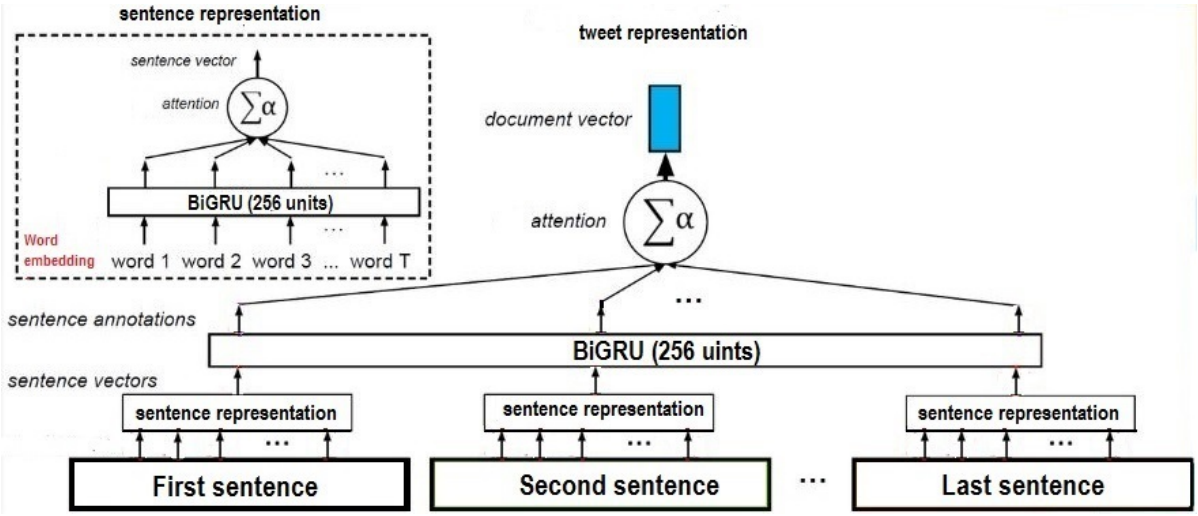

Figure 2: Hierarchical attention for tweet classification

4.3 Model Parameters and Settings

We employ ReLU (Glorot et al., 2011) activation in all the fully-connected layers (100 neurons each) and Tanh activation in GRU cells. We add dropout (Srivastava et al., 2014) of 25% after the attention layers (both at word-level and sentence level) and the fully-connected layers for all the 3 independent models. We train our models for 10 epochs with a batch size of 64. Categorical Crossentropy and Mean Squared Error are the two-loss functions considered for the classification and the regression task, respectively. To backpropagate the loss over the network we use Adam (Kingma and Ba, 2014) optimizer and set the learning rate at 0.001.

All scores are in (%)		**Test Split**				**OLID v1.0**				**OFF'20**			
Sub-Tasks	**Classes**	**P**	**R**	**F**	**#**	**P**	**R**	**F**	**#**	**P**	**R**	**F**	**#**
A	NOT	95	94	95	47738	87	92	90	620	89	99	94	2807
	OFF	92	92	92	32787	77	65	70	240	99	78	88	1080
B	TIN	94	96	95	29844	92	98	95	213	95	65	77	850
	UNT	83	75	79	7951	67	30	41	27	22	75	35	572
C	IND	92	80	86	2198	80	70	79	100	84	88	86	580
	GRP	84	87	85	2292	68	81	74	78	71	64	67	190
	OTH	81	87	84	2407	48	34	40	35	40	37	38	80

Table 2: Per-class Precision, Recall and F1-scores for each sub-task on both the baseline test sets and the actual test set. OFF'20 is the abbreviated form of OFFENSEVAL-2020. # column shows the number of test instances in respective classes.

5 Experiments, Results and Discussion

5.1 Experimental Setup

We use python-based Keras[11] (Chollet and others, 2015) and Scikit learn[12] (Pedregosa et al., 2011) libraries for our experiments. We use macro-F1 score as the metric of evaluation of our models which is also the official metric for all the sub-task in OFFENSEVAL-2020. We evaluate our model's performance on three test scenarios:

- **Test Split:** We split the training data into 80:20 ratio to produce the train and test sets for our experiments.
- **OLID v1.0 Test set:** We consider OFFENSEVAL 2019 (Zampieri et al., 2019a) test set for evaluating our models trained on OFFENSEVAL-2020 (Rosenthal et al., 2020) training data.
- **OFFENSEVAL-2020 Test set (OFF'20):** Finally, we evaluate our models on the OFFENSEVAL-2020 (Rosenthal et al., 2020) test set. Here, we apply a heuristic measure to aid our models for sub-task A and B while making predictions on the test data. Based on the sentiment compound score[13] for any instance in the test set, we append the label *non-offensive* (if compound score $>$ 0.25) or *offensive* (if compound score $<$ 0.25) at the end of the instance in sub-task A. Similarly, in sub-task B, we add the label *positive* (if compound score $>$ 0.25) or *negative* (if compound score $<$ 0.25) to the end of every instance based on their compound score value of sentiment. For sub-task C, we use a text processing tool known as *ekphrasis*[14](Baziotis et al., 2017) to perform tokenization, word normalization, word segmentation and spell correction.

Scores are in (%)	**Test Split**				**OLID v1.0**				**OFFENSEVAL-2020**				
Sub-Tasks	**P**	**R**	**F**	**Acc.**	**P**	**R**	**F**	**Acc.**	**P**	**R**	**F**	**Acc.**	**Rank**
A	93	93	93	94	82	79	80	85	94	89	**90.95**	92	**29**
B	89	86	87	92	79	64	68	90	59	70	**55.69**	66	**27**
C	86	85	85	85	66	64	64	71	65	63	**63.88**	77	**9**

Table 3: Overall macro-averaged Precision, Recall and F1-scores and accuracy values for each sub-task on both the baseline test sets and the actual test set. Numbers in bold show the official F1 scores and ranks achieved in OFFENSEVAL-2020 by our models.

5.2 Results and Discussion

We consider the first two setups as our baseline test sets and the OFF'20 as the official test set. We report the per-class scores of precision, recall and F1 in Table 2 and the overall macro-average scores of the same in Table 3. Our models attained F1-scores of 90.95%, 55.69% and 63.88% in sub-task A, B and C respectively. Our decision of undersampling the overrepresented classes in sub-tasks A and C has helped us to attain the close precision-recall scores (not much distant) for the respective sub-tasks. The effect of undersampling can be felt more from the scores of sub-task B, which we chose not to undersample. It has high recall and comparatively low precision and thus attributing to a moderate F1-score. The above can be realized from the values in Table 3. If we look at the per-class scores for sub-task B in table 2, we will notice that class UNT has a very poor precision score. This can be attributed to the lack of a substantial amount of training instances in the UNT class. Our model for sub-task C finds it tough to separate instances belonging to the OTH class. This may be because of lack of proper signs to identify any target or the implicit nature of such sentences.

[11] https://keras.io/

[12] https://scikit-learn.org/stable/

[13] We use Vader (Hutto and Gilbert, 2014) Sentiment Analyzer to calculate the sentiment compound score for any instance. https://github.com/cjhutto/vaderSentiment

[14] https://github.com/cbaziotis/ekphrasis

6 Error Analysis

Qualitative analysis of the misclassified samples has shown that our application of hierarchical attention-based deep neural network works well for both long and short sentences. On analysis of some offensive instances which are of 2-3 sentences long yet have been misclassified as non-offensive, most of them are found to be implicit in nature. For example,

@USER #WalkAwayFromAllDemocrats Michelle was there 8 years and we sure don't need any more bad advice from her now. We have a new Sheriff in Town, who's #MAGA. Get out and VOTE - ALL COLORS, ALL GENDERS (AKA) ALL AMERICANS!!! URL

The proposed model is also capable of generating actual correct predictions for some instances of which the provided gold labels are incorrect. For example,

@USER @USER @USER She is just nasty

In the above sentence, the gold label is *NOT* whereas our model correctly predicted it as *OFF*.

7 Conclusion

In this work, we present three deep learning based models to identify and categorize offensive tweets based on their type and target. We leveraged the effectiveness of hierarchical attention while learning the semantic information of tweets using BiGRU encoder. This sophisticated attention mechanism enables us to broaden our focus while stressing upon the relevant areas in a tweet. To check the robustness of our models, we test them on two baseline test sets before testing them on the actual task test data. The model for Offensive language identification is ranked 29th among all submissions in the OFFENSEVAL-2020 English sub-task A. The attained test accuracy of 90.95 % is quite close to the top system's (Rank 1) accuracy of 92.23 %. Our model for sub-task B attained 27th rank with attained test accuracy of 55.69 % and 9th rank for sub-task C with test accuracy of 63.88 %. The best systems for sub-task B and C attained accuracies of 74.62 % and 71.45 % respectively. It has been a good learning experience for our team participating in SemEval-2020: Task 12 competition and we look forward to learning from other best-performing entries.

In the future, we will try to address the limitations of our approaches like low scores for the underrepresented classes and choice of optimum thresholds to distinguish among the classes. Also, to generalize our model well to suit other offensive text data, we would like to evaluate our models on other existing dataset on this domain.

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August. Association for Computational Linguistics.

Justin Cheng, Michael Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 1217–1230.

François Chollet et al. 2015. keras.

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*.

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2017. Legal framework, dataset and annotation schema for socially unacceptable online discourse practices in slovene. In *Proceedings of the first workshop on abusive language online*, pages 46–51.

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*, pages 85–90.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.

Òscar Garibo i Orts. 2019. Multilingual detection of hate speech against immigrants and women in twitter at semeval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 460–463.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*.

Ping Liu, Joshua Guberman, Libby Hemphill, and Aron Culotta. 2018. Forecasting the presence and intensity of hostility on instagram using linguistic and social features. In *Twelfth International AAAI Conference on Web and Social Media*.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*.

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

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1125–1135.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher 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.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2017. " like sheep among wolves": Characterizing hateful users on twitter. *arXiv preprint arXiv:1801.00317*.

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

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Hui-Po Su, Zhen-Jie Huang, Hao-Tsung Chang, and Chuan-Jie Lin. 2017. Rephrasing profanity in chinese text. In *Proceedings of the First Workshop on Abusive Language Online*, pages 18–24.

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*, pages 19–26. Association for Computational Linguistics.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*.

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.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

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 (NAACL)*, pages 1415–1420.

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*, pages 75–86.

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 *Proceedings of SemEval*.

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.