Identifying and Categorizing Offensive Language on Twitter

Arunav Saikia
M.S. Data Science
Indiana University - Bloomington
arsaikia@iu.edu

Abstract

Objective

Offensive language, hate speech and cyberbullying have become increasing more pervasive in social media. Individuals frequently take advantage of the perceived anonymity on social media platforms, to engage in brash and disrespectful behaviour that many of them would not consider in real life. The goal of this project is to use a hierarchical model to not only identify tweets/messages with offensive language but categorize the type and the target of offensive messages on social media.

1 INTRODUCTION

Motivation

Methods and techniques for abuse detection are important to detect cyberbullying and hate speech on social media platforms, which affect users from all demographics. It is estimated that approximately 680 million tweets, 4.3 billion messages on Facebook and 300 billion emails are sent everyday (1). A survey by Pew Research Centre states that 40% of adults have personally experienced harassment online (2). There has also been an increase in instances of hate posts that stir violence across societal groups in multiple countries (3). Hate posts with aggressive content if left unmoderated can polarize communities to become dangerously violent and can even lead to riots. White supremacist attacks in the US (4), ethnic cleansing of Rohingya Muslim minorities in Myanmar (5), mob lynching and communal violence in India (6) are just a few examples of this mayhem in recent years.

While most social media websites have inbuilt provisions for its users to flag and report offensive or harmful content, only 8.8% victims actually use this option (2). This statistic highlights the importance of automatic methods to curb and moderate offensive content as well as abusive profiles because passive and manual methods are neither effective nor scalable in real time.

Related work

Most previous work on this topic focused on building supervised classification system to detect specific types of offensive contents - eg racist/sexist/neither (7) or hateful/offensive/neither (8) . In this work, we propose an end-to-end framework to identify the type and target of offensive content on social media website - Twitter. The assumption is that the target of offensive messages is an important indicator that allows us to discriminate between, e.g., hate speech, which often consists of insults targeted toward a group, and cyberbullying, which typically targets individuals etc.

Our proposed method is a hierarchical approach with three sub-tasks:

Tasks

- Sub-task A: Offensive language identification as offensive (OFF) or not offensive (NOT)
- Sub-task B: Automatic categorization of offense types as either targeted insult (TIN) or untargeted (UNT)
- Sub-task C: Offense target identification as individual (IND), group (GRP) or other (OTH)

More motivation

These sub-tasks were released as part of *SemEval 2019 Task 6: Identifying and Categorizing Offensive Language in Social Media* (OffensEval) (9). Our goal is to recreate the results of supervised classification on these three sub-tasks from (10). The combination of labels in each sub-task can be used to identify specific types of offensive language like profane or hate speech or cyber-bullying etc as shown in Table 1. This framework will enable organizations to automatically detect, categorize and take targeted measures based on the type of offensive language. Identifying category of offense is also important for explaining reason behind potential preventive measures which companies might have to take.

Example walkthrough

Tweet	A	B	C	Category
@USER He is so generous with his offers.	NOT	—	—	Clean
WORST EXPERIENCE OF MY FUCKING LIFE	OFF	UNT	—	Profane
@USER she is human garbage that is what is wrong	OFF	TIN	IND	Cyber-bullying
@USER Leftists showing yet again how vile they can be	OFF	TIN	GRP	Hate speech

Table 1: Four tweets from the OLID dataset, with their labels for each level and final categorization.

2 RELATED WORKS

Using supervised learning to detect aggressive or abusive language from text is a widely used approach. Early approaches used simple heuristic-based features to represent text. Spertus (11) used handcrafted lexical features to capture syntax and semantics of text, followed by a decision tree classifier to detect abusive messages. Consequently, Yin et al (12) came up with a formal definition for 'harassment' on Web2.0 and used bag-of-words, TF-IDF and heuristic based semantic features to detect harassment in online forums.

Advances in neural networks has led to vector-based representation of words. These low dimensional vectors ensure that words having similar context occur together in the embedding space. These word embedding techniques like word2vec (13) and GloVe (14) led to classifiers (15) leveraging low dimensional vector representation in addition to linguistic (length, number of punctuations, etc) and syntactic features (POS) applied to comments on social media. Pavlopoulos et al (16) suggested using RNN and CNN based architectures with word embeddings to detect abusive user comments on Wikipedia.

In (17) Singh et al use CNNs and LSTMs to detect aggression in social media text involving multilingual speakers. Detecting and filtering such content on social media is a challenging task because most multilingual populations use 'code-mixing' to express their opinions. Code mixing refers to the practice of using linguistic units from multiple languages in a single sentence. The authors in (17; 18) focus their work on detecting aggressive markers on code mixed Hindi-English text generated on Facebook and Twitter by Indians.

All of the above approaches solely rely on linguistics and NLP to detect abusive content. But recent advances in graph mining and graph representation learning has led to social network analyses approach to tackle this problem. Emphasizing the principle of homophily, Mishra at el (19) mentioned that abusive content comes from users who share a set of common stereotypes and form communities around them. The authors proposed a representation learning approach to create 'profiles' of tweeter users. Specifically, the authors leveraged a node embedding framework called node2vec (20) to learn node representations from a community graph where nodes represent tweeter users and edges represent follower-following relationship. The authors used these profile representations along with text-based features from (22) and a GBDT classifier to detect racist and sexist tweets.

Most of these work do not distinguish between the different types of offensive language that exists on the social media platforms. They either 1) use terms like 'cyberbullying', 'abusive', 'hate speech' interchangeably when there exist clear definition (23) for each form of offensive language based on the content and the target or 2) work on specific forms of offensive language like racist vs sexist (7) while ignoring others. Identification and proper categorization of all forms of offensive language is important so that the organizations and government entities can take proper well informed mitigation steps.

3 DATA

The data used in this work was released as part of *SemEval 2019 Task 6: Identifying and Categorizing Offensive Language in Social Media* (OffensEval) (9). Offensive Language Identification Dataset (OLID) dataset, is a large collection of English tweets annotated using a hierarchical three-layer annotation model to distinguish between whether the language is offensive or not (A), its type (B), and its target (C). It contains 14,100 annotated tweets divided into a training partition of 13,240 tweets and a testing partition of 860 tweets. Each level is described in more detail in the following subsections.

High level statistics

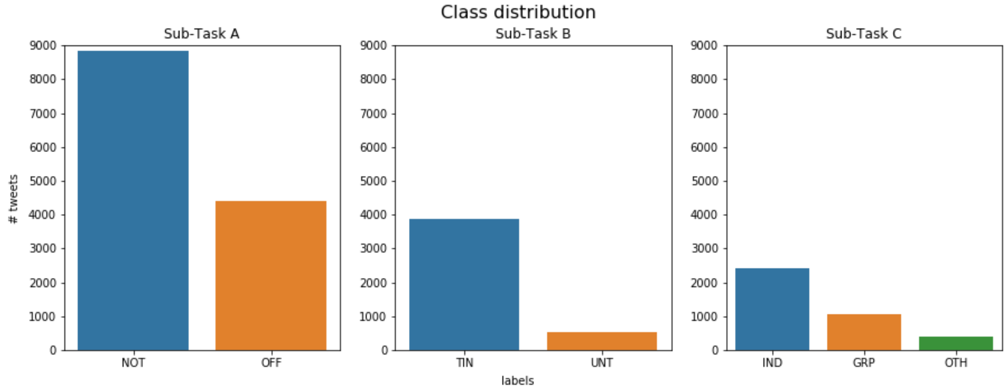

Figure 1: Distribution of labels on the training data for each subtask

3.1 Level A:

- **Not Offensive (NOT):** Posts that do not contain offense or profanity;
- **Offensive (OFF):** Posts containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct. This includes insults, threats, and posts containing profane language or swear words

3.2 Level B:

- **Targeted Insult (TIN):** Posts containing insult/threat to an individual, a group, or others;
- **Untargeted (UNT):** Posts containing non-targeted profanity and swearing. Posts with general profanity are not targeted, but they contain non-acceptable language.

3.3 Level C:

- **Individual (IND):** Posts targeting an individual. This can be a famous person, a named individual or an unnamed participant in the conversation. Insults and threats targeted at individuals are often defined as cyberbulling.
- **Group (GRP):** Posts targeting a group of people considered as a unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or other common characteristic. Many of the insults and threats targeted at a group correspond to what is commonly understood as hate speech.
- **Other (OTH):** The target of these offensive posts does not belong to any of the previous two categories (e.g., an organization, a situation, an event, or an issue).

A	B	C	Train	Test	Total
OFF	TIN	IND	2,407	100	2,507
OFF	TIN	OTH	395	35	430
OFF	TIN	GRP	1,074	78	1,152
OFF	UNT	-	524	27	551
NOT	-	-	8,840	620	9,460
All			13,240	860	14,100

Table 2: Distribution of label combinations in OLID.

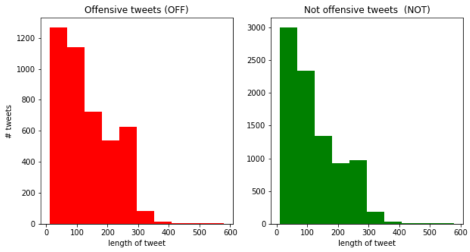

Figure 2: Histogram of # characters in tweets for sub-task A

The distribution of the labels in OLID is shown in Figure 1. We can see the labels are imbalanced for each task, with Sub-task B and Sub-task C having severe class imbalance. The dataset was annotated using the crowdsourcing platform Figure Eight. More information about the data collection and data annotation process can be found here (10). A few high level statistics for the tweets in Sub-task A can be seen in Figure 2, Figure 3 and Figure 4.

4 METHODS

Since each sub-task can be modeled as a supervised classification problem we experimented with multiple classifiers. Before creating numeric features from the text we performed the following text pre-processing steps. We removed all non alpha

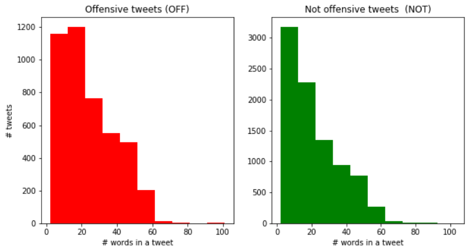

Figure 3: Histogram of # words in a tweet from sub-task A

Setup

numeric characters like emoticons, icons etc; hashtags; URLs; direct @ mentions; and punctuation marks. For each sub-task the pipeline was similar. We split the training data to a train and validation set with random 70-30 split. We trained the classifiers on the train set and tuned the model by looking at the performance of the validation set. Finally, the results and performance for each classifier was reported on the hold out test set. We used 'nltk' for text processing, 'sklearn' for machine learning and 'keras' for deep learning. The different classifiers we experimented with are as follows -

Tools

4.1 NBSVM

Quick summary of the models

NBSVM is an approach to text classification proposed by Wang and Manning (31) that takes a linear model such as SVM (or logistic regression) and infuses it with Bayesian probabilities by replacing word count features with Naive Bayes log-count ratios. Despite its simplicity, NBSVM models act as very strong baselines and have been shown to be both fast and powerful across a wide range of different text classification tasks.

4.2 LSTM

The architecture for our LSTM classifier has an input word embedding layer and a single layer unidirectional LSTM layer with 64 dimensional hidden vector. We initialized the input layer with 100 dimensional pretrained Glove vector embeddings (14). These embeddings are updatable through backpropagation during training. The last hidden state of the LSTM layer is passed through a dense layer and finally through a softmax to get output probabilities. To avoid overfitting, we run our model with early stopping, monitoring the validation loss after each epoch.

4.3 CNN-Text

Our final model is a convolutional neural network based on the architecture proposed by Kim (32). A 1D convolution filter is applied to each possible window of words in the sentence to produce a feature map. We then apply max pooling over time to get a single feature corresponding to a filter. We can have multiple filters of different sizes to obtain multiple features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is the probability distribution over labels. This is basic variant of this model where the input layer has a single channel of trainable word vectors. We also experimented with a complicated version of this architecture (see Figure 5), where the input layer consists of two channels - one has fixed embeddings and the other is trainable through backpropagation.

5 RESULTS

Since the label distribution is highly imbalanced (Figure 1), we evaluate and compare the performance of the different models using macro-averaged F1-score. We further report per-class Precision (P), Recall (R), and F1-score (F1), and weighted average.

5.1 Sub-task A

Table 3 shows the performance of the four models in discriminating between offensive (OFF) and non-offensive (NOT) tweets. We can see that all models perform significantly better than a majority classifier. Neural models are slightly better than NBSVM with LSTM and CNN-Textv2, achieving a macro-F1 score of 0.77.

5.2 Sub-task B

Table 4 shows the performance of the four models in discriminating between targeted insults (TIN) and untargeted (UNT) tweets, which generally refers to profanity. Again, we can see that all models perform significantly better than the majority classifier. LSTM performs best, achieving a macro-F1 score of 0.70. Due to class imbalance we see the models in subtask B are better at identifying targeted insults than untargeted profanity.

5.3 Sub-task C

Finally, Table 5 shows the performance of the four models in identifying the target of offense. Unlike the previous cases this is a multi-class classification

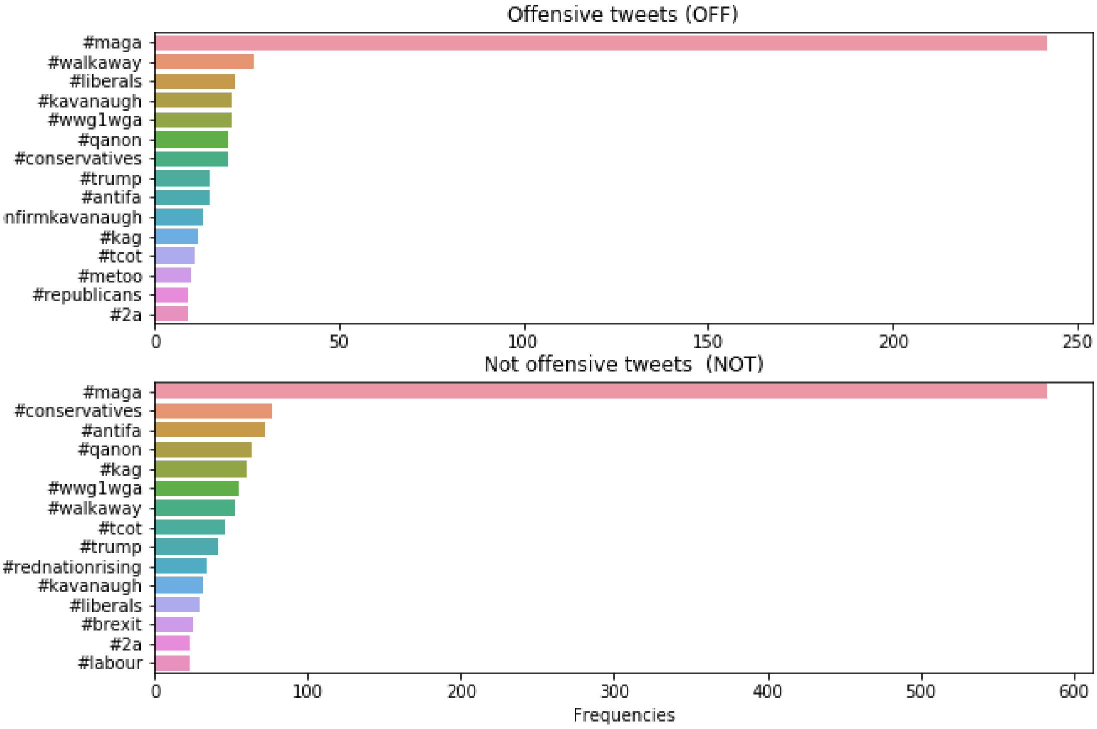

Figure 4: Histogram of hashtags in sub-task A

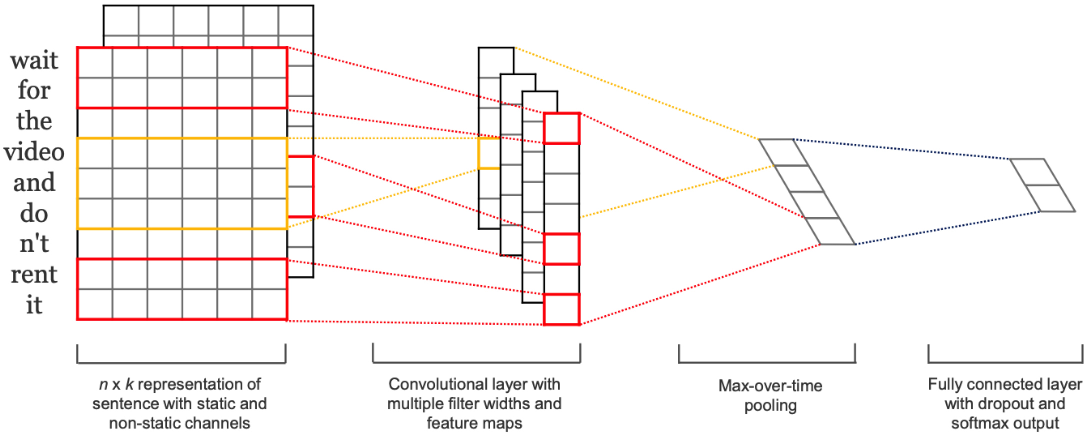
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 5: CNN-Text model architecture with two channels for an example sentence

Results

	NOT			OFF			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
NBSVM	0.83	0.93	0.88	0.75	0.51	0.61	0.81	0.82	0.80	0.74
LSTM	0.85	0.94	0.89	0.78	0.57	0.66	0.83	0.83	0.83	0.77
CNN-Textv1	0.84	0.94	0.89	0.78	0.54	0.64	0.82	0.83	0.92	0.76
CNN-Textv2	0.84	0.94	0.89	0.78	0.55	0.65	0.83	0.83	0.82	0.77
All NOT	0.72	1.00	0.84	0.00	0.00	0.00	0.52	0.72	0.60	0.42

Table 3: Results of Sub-task A: Offensive language identification with the threshold probability at 0.4

Results contd.

	TIN			UNT			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
NBSVM	0.93	0.93	0.93	0.44	0.41	0.42	0.87	0.88	0.87	0.68
LSTM	0.94	0.92	0.93	0.44	0.52	0.47	0.88	0.87	0.88	0.70
CNN-Textv1	0.93	0.90	0.92	0.38	0.48	0.43	0.87	0.85	0.86	0.67
CNN-Textv2	0.92	0.92	0.92	0.38	0.41	0.39	0.86	0.86	0.86	0.66
All TIN	0.89	1.00	0.94	0.00	0.00	0.00	0.79	0.89	0.83	0.47

Table 4: Results of Sub-task B: Offensive language categorization with the threshold probability at 0.2

	GRP			IND			OTH			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
NBSVM	0.70	0.40	0.51	0.55	0.93	0.69	0.00	0.00	0.00	0.52	0.58	0.51	0.40
LSTM	0.65	0.68	0.67	0.67	0.88	0.76	0.00	0.00	0.00	0.55	0.66	0.60	0.48
CNN-Textv1	0.73	0.42	0.54	0.58	0.97	0.72	0.00	0.00	0.00	0.54	0.61	0.54	0.42
CNN-Textv2	0.71	0.38	0.50	0.56	0.95	0.71	0.50	0.03	0.05	0.61	0.59	0.52	0.42
All IND	0.00	0.00	0.00	0.47	1.00	0.64	0.00	0.00	0.00	0.22	0.47	0.30	0.21

Table 5: Results of Sub-task C: Offensive language target identification

problem. The models perform slightly better than the majority classifier, with LSTM having the best result. Further, it is important to note that the performance of all models for the OTH class is 0. This is because, unlike the two other classes, OTH is a heterogeneous collection of targets. It includes offensive tweets targeted at organizations, situations, events, etc., thus making it more challenging for models to learn discriminative properties for this class. Second, there are fewer training instances for this class: only 395 instances for OTH vs. 1,074 for GRP and 2,407 for IND.

6 CONCLUSION

In this work we used a divide and conquer strategy to break down offensive language detection into sub-tasks. A hierarchical end-to-end framework can be used to categorize type of offense based on target and content. We observed that deep learning based supervised classifiers does a better job at individual classification tasks than a simple majority classifier. Both individual classifiers and end-to-end frameworks are important for stakeholders at governments and social media companies to take proper mitigation steps.

The major bottleneck of this work and offensive language categorization in general is the limited amount of annotated data that is available. Literature survey showed there is only two datasets that is annotated in such a hierarchical manner for different types of offense. It would be interesting to add other sub-tasks for eg. to detect specific forms of hate speech like racist, sexist, religiocentric, or homophobic. Another possible direction of research is building reliable semi-supervised methods of annotating data (33).

interpretability, other language, code mixed text

7 SOURCE CODE

Source code for all experiments and results can be found here `https://github.com/arunavsk/OffenseEval2019`.

8 ACKNOWLEDGEMENTS

The author wishes to thank Prof Allen Riddell for his instructions and support. This work was part of Social Media Mining (ILS-Z639) course for Fall 2020 at Indiana University, Bloomington.

References

[1] Online - `https://www.gwava.com/blog/internet-data-created-daily`

[2] Online - `https://www.pewresearch.org/internet/2017/07/11/online-harassment/`

[3] Online - `https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons`

[4] Online - https://www.newyorker.com/magazine/2017/02/06/inside-the-trial-of-dylann-roof

[5] Online - https://iwpr.net/global-voices/how-social-media-spurred-myanmars-latest

[6] Online - https://www.washingtonpost.com/graphics/2018/world/reports-of-hate-crime-cases-have-spiked/-in-india/

[7] Waseem, Zeerak, and Dirk Hovy. "Hateful symbols or hateful people? predictive features for hate speech detection on twitter." In Proceedings of the NAACL student research workshop, pp. 88-93. 2016.

[8] Davidson, Thomas, Dana Warmsley, Michael Macy, and Ingmar Weber. "Automated hate speech detection and the problem of offensive language." arXiv preprint arXiv:1703.04009 (2017).

[9] Zampieri, Marcos, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)." arXiv preprint arXiv:1903.08983 (2019).

[10] Zampieri, Marcos, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. "Predicting the type and target of offensive posts in social media." arXiv preprint arXiv:1902.09666 (2019).

[11] Spertus, Ellen. "Smokey: Automatic recognition of hostile messages." In Aaai/iaai, pp. 1058-1065. 1997.

[12] Yin, Dawei, Zhenzhen Xue, Liangjie Hong, Brian D. Davison, April Kontostathis, and Lynne Edwards. "Detection of harassment on web 2.0." Proceedings of the Content Analysis in the WEB 2 (2009): 1-7.

[13] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013)

[14] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[15] Nobata, Chikashi, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. "Abusive language detection in online user content." In Proceedings of the 25th international conference on world wide web, pp. 145-153. 2016.

[16] Pavlopoulos, John, Prodromos Malakasiotis, and Ion Androutsopoulos. "Deeper attention to abusive user content moderation." In Proceedings of the 2017 conference on empirical methods in natural language processing, pp. 1125-1135. 2017.

[17] Singh, Vinay, Aman Varshney, Syed Sarfaraz Akhtar, Deepanshu Vijay, and Manish Shrivastava. "Aggression detection on social media text using deep neural networks." In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pp. 43-50. 2018.

[18] Bohra, Aditya, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. "A dataset of Hindi-English code-mixed social media text for hate speech detection." In Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media, pp. 36-41. 2018.

[19] Mishra, Pushkar, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. "Author profiling for abuse detection." In Proceedings of the 27th International Conference on Computational Linguistics, pp. 1088-1098. 2018.

[20] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855-864. 2016.

[21] Mishra, Pushkar, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. "Abusive language detection with graph convolutional networks." arXiv preprint arXiv:1904.04073 (2019).

[22] Badjatiya, Pinkesh, Shashank Gupta, Manish Gupta, and Vasudeva Varma. "Deep learning for hate speech detection in tweets." In Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759-760. 2017

[23] Fortuna, Paula, and Sérgio Nunes. "A survey on automatic detection of hate speech in text." ACM Computing Surveys (CSUR) 51, no. 4 (2018): 1-30.

[24] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701-710. 2014.

[25] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

[26] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." In Advances in neural information processing systems, pp. 1024-1034. 2017.

[27] Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).

[28] Schlichtkrull, Michael, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. "Modeling relational data with graph convolutional networks." In European Semantic Web Conference, pp. 593-607. Springer, Cham, 2018.

[29] Zhang, Chuxu, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. "Heterogeneous graph neural network." In Proceedings of the 25th

ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 793-803. 2019.

[30] Wang, Xiao, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. ”Heterogeneous graph attention network.” In The World Wide Web Conference, pp. 2022-2032. 2019.

[31] Wang, Sida I., and Christopher D. Manning. ”Baselines and bigrams: Simple, good sentiment and topic classification.” In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 90-94. 2012.

[32] Kim, Yoon. ”Convolutional neural networks for sentence classification.” arXiv preprint arXiv:1408.5882 (2014).

[33] Miok, Kristian, Gregor Pirs, and Marko Robnik-Sikonja. ”Bayesian Methods for Semi-supervised Text Annotation.” arXiv preprint arXiv:2010.14872 (2020).