Cyberbullying Detection in Twitter Using Machine Learning

Amirmohammad Shahbandegan
Depatrment of Computer Science
Lakehead University
Thunderbay, Canada
1172613

Lakshmi Preethi Kamak
Depatrment of Computer Science
Lakehead University
Thunderbay, Canada
1160111

Mohammad Ghadiri
Depatrment of Computer Science
Lakehead University
Thunderbay, Canada
1170979

***Abstract*—Cyberbullying can have serious legal consequences in Canada including jail time and fines. Social media companies such as Twitter, Facebook etc., have resources and guides on cyberbullying and are relying on passive reporting mechanisms. However, 90% of cyberbullying activities go unreported making the presence of an active cyberbullying detection system crucial. Our proposed architecture detects cyberbullying using a two-step multiclass classification method using traditional machine learning algorithms in a balanced dataset distributed into six cyberbullying classes. Our model tackles both balanced classes and imbalanced classes in the dataset and outperforms the current ML and DNN baselines. This work experiments with multiple text embedding methods to compare and find the most suitable strategy in detecting cyberbullying. Our results provide significant insights into the effectiveness of constructing architectures using traditional ML models rather than implementing deep learning methods to overcome the cyberbullying issue. We have released our models and code.**

***Index Terms*—Cyberbullying Detection, Social Media, Machine learning**

I. Introduction

With the growth of social media platforms, cyberbullying is a significant concern among the younger population. Cyberbullying can have adverse effects on vulnerable people and is considered a severe threat. Cyberbullying can be defined as "the use of digital technology to inflict harm repeatedly or to bully" [1]. Statistics show that about 36% of people felt cyberbullied in their lifetime, 60% of teenagers experienced some cyberbullying, and 87% of young people have observed cyberbullying [2]. In Canada, cyberbullying can have serious legal consequences. Cyberbullies can face jail time, have their devices taken away, and may even have to pay their victims [3]. Many Social media companies like Facebook, Twitter, Instagram, etc., have resources and guides on cyberbullying. Although social media companies rely on passive reporting mechanisms, 90% of cyberbullying activities go unreported [4]. Therefore, the presence of an active cyberbullying detection system is crucial. In this project we experiment with various classifiers using a proposed two step classification architecture to get achieve a higher accuracy in the cyberbullying dataset , while using traditional ML models rather than Deep learning models.

II. Literature Review

A. SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection

Wang et al. [6] developed a Graph Convolutional Neural Network (GCN) based model with eight different tweet embedding methods and six different classification models as a baseline to compare the performance of cyberbullying detection. They employed a GCN model by generating a Tweet graph using the cosine similarity of the Tweets. They leverage and present a case for the use of Dynamic Query Expansion (DQE) data mining technique in a novel way to combat severe class imbalance in their dataset curation process; the class distribution was 0.995% for age, 1.64% for ethnicity, 39.1% for gender, 11.7% for religion, and 46.6% for other. This imbalance significantly affects the training process because of the resulting bias towards the Gender and Other classes, discounting Age and Ethnicity.

They gathered more data via semi-supervised learning by augmenting current datasets to solve class imbalance and integrated GetOldTweets3 for real-time updates and new data. By using a combination of real-time queries and executing separate processes for the fine-grained classes, they built a labeled dataset in a semi-supervised fashion. The baseline evaluation metrics were test accuracy and F1-Score. Wang et al. [6] achieved decent accuracy and F1-Score with simple word embedding methods like Bag of Words and TF-IDF with traditional classifiers, such as decision trees.

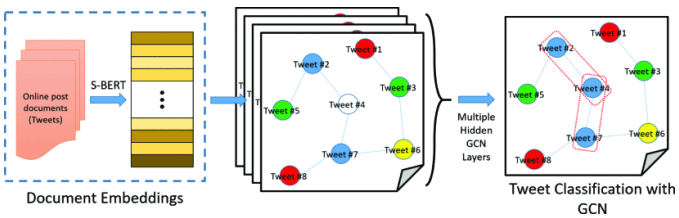

Fig. 1. Classification architecture of Wang et al. [6]

B. Am I Being Bullied on Social Media? An Ensemble Approach to Categorize Cyberbullying same dataset

Ahmed et al. [7] proposed a neural ensemble method of transformer-based architectures with an attention mechanism. Four transformer-based models are combined to achieve higher accuracy. Max voting and probability averaging are the two ensemble methods used in their work. They evaluate their model for the FGCD dataset, five classes (40,000 tweets), to find the types of cyberbullying along with the 'not cyberbullying (Not CB)' class.

In comparison, Ahmed et al. [7] claim that their ensemble model of probability averaging outperforms Wang et al.'s [6] best model by 1.22% in test accuracy in 1.15% in F1-Score. Their proposed architecture can learn abstract features with the attention mechanism and performs better on these datasets than the feature-based approaches Wang et al. [6].

Fig. 2. Classification architecture of Ahmed et al. [7]

C. Comparison with Previous Works

The proposed project is similar to the previously discussed studies [6], [7] in a few ways. We follow Wang et al.'s [6] baseline models and text embedding techniques for our implementation. Ahmed et al. [7] have also done a five-class classification; hence, we will compare our results with them for the benchmark.

Our work will differ from Wang et al. [6] in the approach to handling class imbalance. In the first step of the proposed algorithm, there is a class imbalance between identifying cyberbullying with classes Cb and NotCb. We will be performing undersampling in the dataset for the first step using techniques like random undersampling. In contrast to the previous literature that has implemented deep learning and pretrained language models, we use traditional ML methods with the state of the art embedding techniques.

1) Similarity:

- Implementation of the baseline models and text embedding techniques is based on Wang et al. [6]
- We have compared our results with Ahmed et al. [7] who has done both five-class and six-class classification

2) Difference:

- Our work is experimenting with both five-class and six-class classification
- Traditional ML methods with a two-stage pipeline vs Deep learning implementation by fine-tuning language models (400M parameters)

III. Dataset

A. Data source

Wang et al. [6] developed a dataset - FGCD by combining six different datasets and classifying the tweets by labelling and grouping the same classes. Fine-grained balanced cyberbullying dataset released in 2020 [5]. FGCD is a balanced dataset of about 48,000 Twitter comments distributed into six cyberbullying classes, 'Age', 'Ethnicity', 'Gender', 'Religion', 'Other', and 'NotCb (not cyberbullying)'. To remove the class imbalance, they used Dynamic Query Expansion (DQE) and increased the number of samples of each class in a semi-supervised manner. They randomly sampled 8000 tweets of each class from the different datasets and formed a balanced dataset of size 48000.

B. Data description

The data has a simple structure with only two columns: tweet and label. It is a fully balanced dataset with almost 8k tweets in each class, with an average of 24 words per tweet. The textual data needs to be converted to a feature vector to be used with a learning algorithm. There are no missing values in the FGCD dataset. Currently, the dataset has six balanced classes. But there is an imbalance between the Cb and Not Cbtweets. We propose to perform oversampling techniques on NotCb class by generating around NotCb 39,750 tweets before the binary classification step. We might also consider undersampling by removing the cyberbullying classes to reduce the majority and fix the imbalance.

IV. Methodology

A. Project Architecture

Our proposed architecture detects cyberbullying using a two-step multiclass classification method using machine learning classifiers like XG-Boost, SVM and other traditional models. The Fine-Grained Cyberbullying Dataset (FGCD), developed by Wang et al., is a balanced dataset of about 48,000 Twitter comments distributed into six cyberbullying classes, 'Age', 'Ethnicity', 'Gender', 'Religion', 'Other', and 'NotCb (not cyberbullying)'. The first step will be a binary classification model that can identify cyberbullying in a tweet with classes cyberbullying (Cb) and Not cyberbullying (NotCb). The second step will be a fine-grained multiclass classification that determines the characteristics of the target

from the remaining five cyberbullying classes.

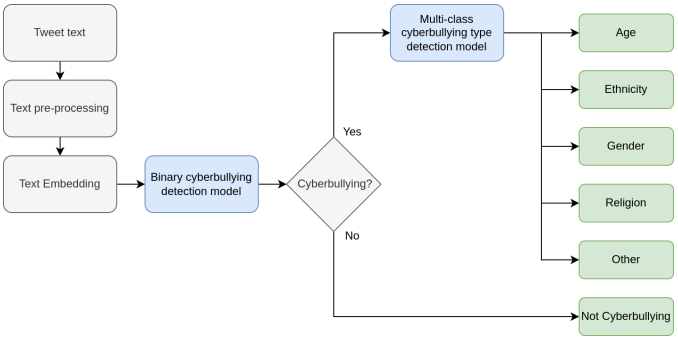

Fig. 3. Our proposed two-step classification architecture

B. Environment

- Python 3
- Text processing: gensim, emoji , nltk, spacy, contractions, sentence_transformers
- scikit-learn
- imblearn
- Google colab environment - for model development purposes
- Compute Canada - for high performance, parallel processing purposes

C. Text pre-processing

Text preprocessing is fundamental for natural language processing (NLP) tasks. It is a method to clean and standardize the text data and make it readable by the model. Text data contains noise in various forms like emojis, punctuation, text in various cases.

The following preprocessing on the tweet text was implemented.

- Stripped links, mentions, retweet flag, stop words, and punctuation
- Hashtags were not removed
- Removed extra whitespaces, special characters, and numbers.
- Emojis are replaced with a corresponding word.
- Expanded contractions and lowercased all text.
- Performed lemmatization only for the bag of words and TF-IDF embeddings.

D. Text embedding

Text embedding is used for analysis in the form of a vector that encodes the meaning of words that are closer in the vector space are expected to be similar in meaning [12].

After the text preprocessing, each tweet is converted to a feature vector keeping the semantics of the tweet. The following text embedding methods were used in our experimentation.

- Bag of Words (BOW) - BOW represents the text as a set of its words in which the frequency of occurrence of each word is used as a feature.
- TF-IDF - "TF-IDF, short for Term Frequency-Inverse Document Frequency, is a numerical statistic that reflects a word's importance in a document in a collection or corpus" [8].
- word2vec - The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text [9]. This study will use the google news pre-trained version of this model.
- GloVe - GloVe obtains word representations in an unsupervised manner by performing aggregated global word-word co-occurrence statistics from a corpus [10].
- fastText - fastText is a word embedding method developed by Facebook research. It works similar to Word2vec but generalizes better to unseen words.
- Sentence BERT (SBERT) - "SBERT is a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity" [11].

E. Binary Classification

The data is loaded from the embeddings package. Given that the data is balanced among 6 classes, the binary model suffers from class imbalanc. The Binary labels are generated for the cyberbullying and not cyberbullying classes and are marked True & False respectively. To overcome this problem approaches like Near Miss algorithm and Random under sampling were applied.Multiple classifiers were used in the experiment to find a classifier best suited for this problem. The classifiers with default hyperparameters were fitted with the data. 5-fold Cross-Validation to train and test the models was applied. All the experiments are executed in parallel using the multiprocessing module.

The undersampling algorithm in the experiments are discussed below:

- Near Miss - This technique removes the datapoint from the majority class when two points in the distribution belonging to different classes are relatively close to each other, attempting to balance the distribution.
- Random Undersampling - In the random under-sampling, the majority of class instances are discarded at random until a more balanced distribution is reached.

F. Fine Grained Classification

The data is loaded from the embeddings package. Not-cyberbullying samples were removed from the dataset. Multiple classifiers were used to find the best-suited classifier for this problem. 5-fold Cross-Validation to train and test the models was applied. The accuracy and F1 score was measured as the average of 5-fold and the results was saved.

The 5 classifiers in the experiments are discussed below:

- Logistic Regression (LR) - LR is a statistical method that uses a logistic function to model a binary dependant variable.
- K-Nearest Neighbors (KNN) - KNN is a non-parametric supervised learning algorithm where the function is only

approximated locally and all computation is deferred until function evaluation.

- Support Vector Machine (SVM) - SVMs being one the most robust prediction methods, are supervised learning models based on statistical learning frameworks. SVM maps training examples to points in space so as to maximize the width of the gap between the two categories.
- eXtreme Gradient Boosting(XGBoost) - "XGBoost is an implementation of gradient boosted decision trees designed towards speed and performance enhancement".
- Multi-layer Perceptron (MLP) - MLP is a class of neural networks with at least three layers: an input layer, a hidden layer, and an output layer. MLP uses non-linear activation functions and a supervised learning technique called backpropagation for training.

V. Experimental Results

The results are categorized into two main sections for individual classifiers and the final pipeline model. In the first stage, the best model for the binary and multi-class models are established using exhaustive search. The results from the first set of experiments are then used to build the final pipeline to undertake the 6-class classification problem.

A. Individual Classifiers

Multiple embeddings and classifiers are studied and experimented with to find the most suitable model for the binary and multiclass problems. Every possible permutation of the following configurations are tested seperately using 5-fold cross validation resulting in 450 different iterations running concurrently.

- 3 models (5-class, binary without undersampling and binary with undersampling)
- 6 text embedding methods
- 5 classifiers

Tables I and II show the accuracy and F1 score for the 5 class classifiers with different embeddings. It can be seen that the combination of the Bag of Words embedding and XGBoost classifier achieves the best results among all of the candidate methods.

The results for the binary model without under sampling are shown in Tables III and IV and the results for the same model with random under sampling can been seen in Tables V and VI. Comparing the results from these two binary models reveals that the best results are achieved without under sampling and the best combination for this model is the Sentence Bert embdding and the SVM classifier.

B. Pipeline Model

Following the results from the previous experiments, the final model incorporates a two step pipeline where the tweets are first sent to a binary model built with sentence bert and SVM. If the binary model flags the tweet as a cyberbullying class, the tweet is then sent to the second model in the pipeline which is the multi-class model built with bag of words and XGBoost to detect the type of cyberbullying. The final

TABLE I
F1 scores for 5-class models.

	knn	lr	lsvm	mlp	svm	xgb
bow	73.06	94.51	93.88	92.58	94.3	**94.75**
ft	78.64	88.17	89.58	92.26	92.44	91.59
glove	80.62	88.89	89.45	91.59	92.68	91.71
sbert	89.85	91.94	92.50	92.31	93.33	91.43
tfidf	36.93	94.17	94.33	92.46	94.30	94.64
w2v	NaN	89.31	89.47	92.29	93.08	91.79

TABLE II
Accuracy scores for 5-class models.

	knn	lr	lsvm	mlp	svm	xgb
bow	73.06	94.51	93.88	92.58	94.30	**94.75**
ft	78.64	88.17	89.58	92.26	92.44	91.59
glove	80.62	88.89	89.45	91.59	92.68	91.71
sbert	89.85	91.94	92.50	92.31	93.33	91.43
tfidf	36.93	94.17	94.33	92.46	94.30	94.64
w2v	NaN	89.31	89.47	92.29	93.08	91.79

TABLE III
F1 scores for binary models.

	knn	lr	lsvm	mlp	svm	xgb
bow	NaN	92.03	90.67	90.49	92.82	**92.93**
ft	NaN	91.74	91.86	91.67	92.16	90.62
glove	NaN	92.10	92.11	90.55	92.64	90.85
sbert	92.59	92.43	92.61	90.94	93.16	91.02
tfidf	NaN	92.63	91.75	90.10	92.27	92.81
w2v	NaN	91.88	91.93	90.81	92.40	90.58

TABLE IV
Accuracy scores for binary models.

	knn	lr	lsvm	mlp	svm	xgb
bow	NaN	86.37	84.32	84.08	87.35	87.59
ft	NaN	85.32	85.53	85.72	86.05	83.92
glove	NaN	86.15	86.06	84.08	87.04	84.32
sbert	87.25	86.92	87.24	84.75	**88.09**	84.59
tfidf	NaN	87.09	85.90	83.41	86.47	87.40
w2v	NaN	85.71	85.72	84.50	86.63	83.86

TABLE V
F1 scores for binary with under sampling models.

	knn	lr	lsvm	mlp	svm	xgb
bow	67.36	86.25	84.83	84.10	85.86	86.17
ft	90.12	84.76	84.73	87.20	86.54	85.73
glove	**90.26**	85.20	85.18	86.77	86.46	85.61
sbert	87.91	86.73	86.70	86.44	**85.98**	85.44
tfidf	29.27	86.11	85.83	83.02	85.36	85.77
w2v	NaN	84.81	84.65	87.17	86.49	85.71

TABLE VI
Accuracy scores for binary with under sampling models.

	knn	lr	lsvm	mlp	svm	xgb
bow	58.24	79.39	77.23	76.12	79.10	79.56
ft	83.30	77.01	76.98	80.38	79.67	78.48
glove	**83.74**	77.56	77.55	79.73	79.61	78.31
sbert	81.29	79.87	79.85	79.40	**79.11**	78.07
tfidf	30.02	79.20	78.65	74.46	78.31	79.00
w2v	NaN	77.13	76.92	80.31	79.69	78.48

architecture of the pipeline model is depicted on Fig. 4. Table VII shows the class-wise F1 scores of our proposed model and compares it with previous work [7].

TABLE VII
CLASS-WISE F1 SCORES OF THE PIPELINE MODEL COMPERD WITH PREVIOUS WORK

Class	Ahmed et al. [7]	Our pipeline model
Age	0.93	**0.98**
Ethnicity	0.95	**0.98**
Gender	0.86	**0.87**
NotCB	**0.56**	0.55
Others	0.61	**0.67**
Religion	0.93	**0.95**
Average	0.80	**0.83**

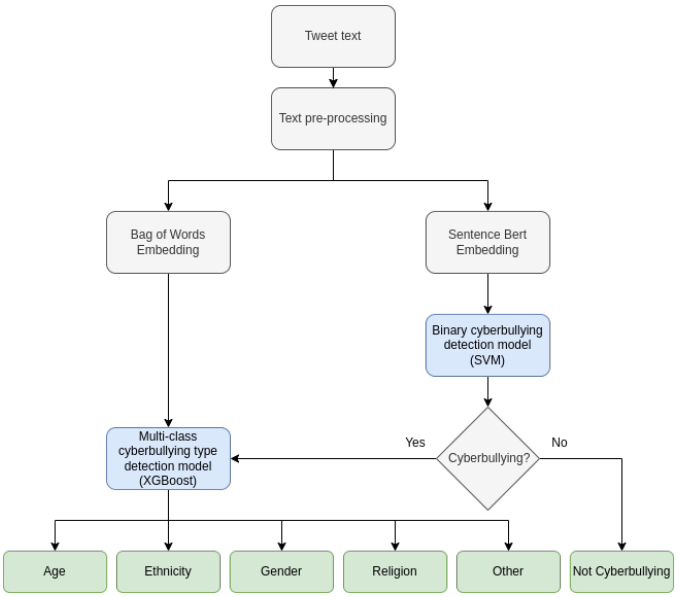

Fig. 4. Our final best classification architecture

VI. Discussion and Conclusion

A. Challenges Faced

1) Long Execution time: Running all of the 450 experiments in the first set was challenging since it required a huge amount of time to complete. To overcome the slow execution time of the models, the experiments were conducted in parallel on high-preformance compute nodes in Compute Canada clusters

2) Random Under Sampling: The use of random under sampling had a negative effect on model's performance. This can be due to the huge loss of information that happens when using this under sampling method. It is possible to experiment with over sampling techniques in the future to overcome this issue.

3) Long Execution time of the near miss algorithm: The authors initial plan was to experiment with near miss as a potential under sampling method to achieve better results in the binary model. However, due to the large number of samples and the high-dimensional embeddings used, the near miss algorithm was not able to find the candidate points in a timely manner. Therefore this method was not used and the experimental results for this method are not available.

4) High memory consumption of the KNN classifier: The KNN classifier is a relatively memory-extensive algorithm compared to the other methods used in this work. Given that these experiments were running concurrently on a single machine with hundreds of CPU cores and a limited amount of main memory, the results for this classifier are not complete and some of the experiments stopped when they ran out of memory.

B. Future Scope

The authors are planning to improve this work in the future in two ways. First, the effect of the hyper-parameters on the final pipeline model are not studied in this work. It is possible to run more experiments and find the optimal hyper-parameters for the binary and multi-class models and therefore increase the performance of the model.

Second, as mentioned before, the under sampling technique used in the binary classifier resulted in poor performance. One way to improve the quality of the binary classifier is by employing over sampling methods instead. There are two general strategies in generating synthetic text. One way is to use the general methods such as SMOTE and ADASYN on the embedded text. The other can be achieved by generating new synthetic text using methods such as back-translation and word-replacement and then generating new embeddings for these synthetic text data. Both of the mentioned methods could be a successful way to improve the performance of the binary model and are great directions to work in the future.

Acknowledgements

References

[1] E. Englander, E. Donnerstein, R. Kowalski, C. A. Lin, and K. Parti, "Defining cyberbullying," Pediatrics, vol. 140, no. Supplement 2, pp. S148–S151, 2017.

[2] All the latest cyber bullying statistics and what they mean in 2022. BroadbandSearch.net. (n.d.). Retrieved April 7, 2022, from https://www.broadbandsearch.net/blog/cyber-bullying-statistics

[3] Canada, P. S. (2021, February 5). Government of Canada. Cyberbullying can be against the law - Canada.ca. Retrieved April 7, 2022, from https://www.canada.ca/en/public-safety-canada/campaigns/cyberbullying/cyberbullying-against-law.html

[4] Hatfield, H. (n.d.). Stop school bullying and cyber-bullying. WebMD. Retrieved April 7, 2022, from https://www.webmd.com/parenting/features/prevent-cyberbullying-and-school-bullying

[5] J. Wang, K. Fu and C.-T. Lu, "Fine-grained balanced cyberbullying dataset", 2020.

[6] J. Wang, K. Fu and C. -T. Lu, "SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection," 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 1699-1708, doi: 10.1109/BigData50022.2020.9378065.

[7] T. Ahmed, M. Kabir, S. Ivan, H. Mahmud and K. Hasan, ”Am I Being Bullied on Social Media? An Ensemble Approach to Categorize Cyberbullying,” 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 2442-2453, doi: 10.1109/BigData52589.2021.9671594.

[8] Rajaraman, Anand, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2011. Mikolov, Tomas, et al. ”Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).

[9] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[10] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[11] Jurafsky, Daniel; H. James, Martin (2000). Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, N.J.: Prentice Hall. ISBN 978-0-13-095069-7.