Identification of Toxic Comments in On-line Platforms

Mehvish Saleem, Ramanpreet Singh, and Ehsan Montazeri

***Abstract*— In this project, we use differ NLP and supervised machine learning techniques to design an effective model for identification of toxic comments. We use our model on data scraped from multiple Facebook pages to gain insight about the rate and types of toxicities present in different communities.**

I. INTRODUCTION

Toxicity in social interactions in online platforms is very common and is a big problem. It can have severe repercussions on the victims, such as low self esteem, health problems, depression and isolation. In a survey conducted in 2015 [1], over 43% of teens reported to have been targeted by cyber-bullying, among which 64% stated that online toxicities had negatively impacted their feelings of safety and ability to learn at school. The presence of an automated tool for detecting toxicities could greatly help the platforms' moderators in identifying such comments to remove them and take actions against the posters, and as a result, make their communities safer for their users.

This cause motivated us to do an analysis of online toxic comments. In particular, we aimed to perform the following.

- Training a machine learning model to identify toxicity level of comments in online platforms
- Identifying the different types of toxicities present online. In particular, we would like to identifying racist, sexist, and homophobic comments.
- Using our developed models to gain insight about the types of toxicities present in different online communities. For example, we are interested in comparing sports pages to news pages in terms of the amount of toxicities they contain, as well as the types of those toxicities.
- Observe the changes in toxicity level in these platforms over the recent time

To this end, we used two labeled datasets for the training process, namely the Wikipedia talk pages dataset [2] and the SFU Opinion and Comments Corpus (SOCC) [3]. More details about these datasets are provided in section II-A. To perform our analysis, we developed a full data science pipeline, outlined in detail in section II. The pipeline includes exploratory data analysis (EDA), dataset integration and preprocessing, model selection and evaluation, testing the selected model, toxicity categorization, analyzing the social media data using our developed models, and visualizations of the found results.

We preprocessed and integrated the two datasets in a way to have a ternary toxicity level as the class label.

We explored different NLP techniques such as bag of words and Doc2Vec, and multiple supervised machine learning algorithms such as perceptron, random forests, SVM, and recurrent neural networks(RNN). The best validation results were obtained by the use of word embeddings and RNNs with gated recurrent units (GRU).

We then used our model on over 570,000 comments that we scraped from 10 different Facebook pages from three categories of news, entertainment, and sports. The data is from comments that users left on those pages between Jan 2017 - March 2018. Finally, the discovered toxic comments were further analyzed to find the types of toxicities in each category.

There have been some similar studies and projects in the literature and industry. In particular, Perspective API [4] is an attempt by Google to improve online conversations by studying and identifying toxic comments. Their model is trained on datasets from Wikipedia and The New York Times. Moreover, Kaggle recently held a competition [2] to build a model for identifying different types of toxicities like threats, obscenity, and insults, aiming to outperform the Perspective API's current model. They used the Wikipedia Talk Pages dataset for training and validation. We are using the same dataset for our project, even though detecting insults and threats is out of the scope of this work.

In the rest of this report, we first discuss the different stages of our pipeline in section II, which will cover all the explored models and techniques used in our project. We present our findings in section III, and our conclusions and possible future work in section IV.

II. PIPELINE AND IMPLEMENTATION DETAILS

In this section, all the stages of our data science pipeline are explained in details. An overview of the pipeline is shown in Figure 1.

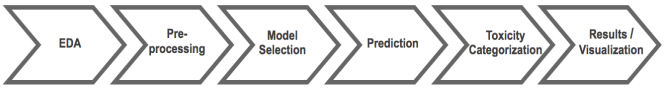

Fig. 1: Overview of the used data science piepline

A. Datasets

We used the following two datasets for training our model.

- Wikipedia talk pages comments:
  Published in Kaggle's Toxic Comment Classification Challenge [2], this dataset contains over 160,000 comments with four binary class labels: toxic, very toxic, offensive, and obscene.

- SFU Opinion and Comments Corpus (SOCC):
  This dataset [3] is prepared by the Simon Fraser University's department of linguistics. It contains over 1,000 comments with a 5-level class label, ranging between 0-4, with 0 meaning non-toxic and 4 most toxic. The data has been extracted from the comment section of Globe and Mail and has been crowd-sourced and verified by experts.

B. EDA

Prior to merging the datasets, we explored both datasets using Pandas to gain more insight about them. The two datasets were very different in terms of toxicity, with Wikipedia containing much more toxic comments. A comment rated as 4 in the SOCC dataset was not as toxic as the comments labeled as 'very toxic' in the Wikipedia dataset. A word cloud made from the toxic comments in this dataset is shown in Figure 2.

Neither dataset had any missing values. They were however

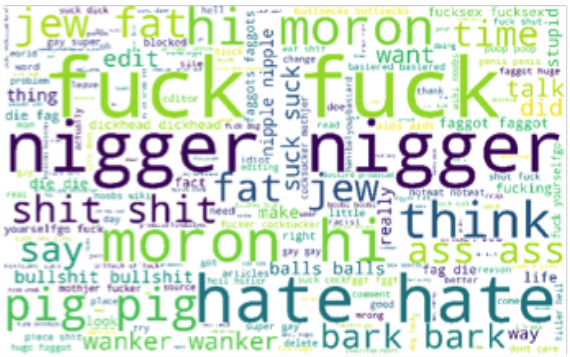

Fig. 2: Word cloud made of toxic comments in the Wikipedia dataset

highly imbalanced, with much more non-toxic comments present in both datasets. Figure 3 shows this imbalance, where the number of comments with label 0 shown in blue is higher than those of the other two classes.

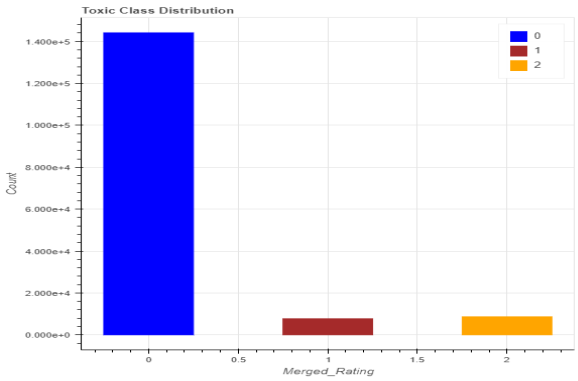

Fig. 3: Class Imbalance: Class 0 is more common than the other two

TABLE I: Mapping between the original class labels and the merged dataset

SOCC	Merged Dataset	Wikipedia	Merged Dataset
4	1	very toxic	2
3	1	obscene	2
2	0	toxic	1
1	0	offensive	1
0	0	Non-toxic	0

C. Data Integration and Pre-Processing

The two datasets had different number of class labels, as well as different notions of toxicity level, as explained in the previous section. In order to integrate them, we used a ternary class label of toxicity level. The rules for converting the original class labels into those of the merged dataset are outlined in Table I. All toxic SOCC comments were inspected to make sure this conversion makes sense. This was feasible since there was a total of 1,000 comments in that dataset. All the conversions and the merging were done using Pandas. For pre-processing, we removed the stop-words and lemmatized all the words using NLTK [5].

D. Model Selection

Since we were dealing with sentences, we needed to first featurize the data to be able to use it with supervised learning algorithms. To do that, we experimented with TF-IDF, word embeddings, and Doc2Vec [6], and then used the resulting featurized data as input to various machine learning models, as explained below.

1) TF-IDF: In this method, each sentence is represented by a very high-dimensional sparse vector, where the vector dimension is equal to the total number of distinct words present in the entire corpus. Each vector entry corresponding to the words that make up that sentence represents the TF-IDF score of the words. All other entries corresponding to the other words are set to zero. We used scikit-learn's TF-IDF module for implementation.

2) Doc2Vec: Doc2Vec is an extension of the popular word2vec[6]. It uses vectors of size N to represent a paragraph. N is a hyper-parameter and was varied in the range 100-500 in our experiments. Paragraphs of similar context are assigned vectors that are close to each other. We used Gensim's Doc2Vec module for our implementation.

3) Classifiers: After featurizing the comments using the above mentioned methods, we experimented with several supervised learning algorithms, such as Perceptron, Random Forrest, and Support Vector Machine. All implementations were done using scikit-learn. The training for models based on Doc2Vec were also implemented in Pyspark ML, due to the large size and denseness of faturized data.

In order to feed the data to the models, the dataset was first split into train, validation, and test datasets, with ratios of 70%,15%, and 15% respectively. Due to having imbalanced classes, we used the smote method (using scikit-learn) to oversample the training data (the 70%) belonging to classes 1 and 2. We also experimented with assigning different weights to the classes, where the weight of each class is

inversely proportional to its ratio. Both methods provided similar results. The models were tuned to provide the best performance on the validation set, where the used evaluation metrics were recall and precision. The results are presented in section III. The hyper-parameters tuned for random forest were number of trees, depth of trees, and maximum features. For Perceptron, the tuned parameters were number of epochs, type of solver, and the step size. Training SVMs took a very long time so were not able to perform parameter tuning. We also did not experiment with multi-layer perceptron due to the very large number of features we had obtained from Doc2Vec and TF-IDF, but rather tried RNNs which are popular for text data.

4) Recurrent Neural Networks: Recurrent Neural Network is a class of artificial neural network that uses the output of hidden states produced by previous input and current input to produce current output. This allows RNN to remember everything that has been fed as the input, unlike other neural networks where inputs are independent of each other.
Gated recurrent unit (GRU) is an improved version of standard recurrent neural networks which do not suffer from the problem of vanishing gradients. A GRU consists of two gates called update gate and reset gate which decide what information to be passed to the output. In our model we have chosen a Bi-directional GRU with 50 units. The full architecture is shown in Figure 4.

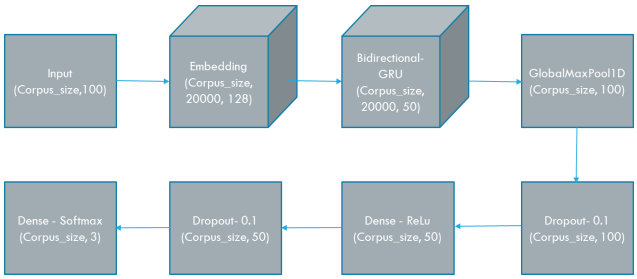

Fig. 4: Architecture of our GRU RNN

We used Keras with TensorFlow back-end to implement RNNs. Unlike the other models we tried, we did not remove any stop-words to keep the sentence sequences. We constructed our own word embeddings by tokenizing each comment and assigning an index to each word so that each comment can have a numerical representation in order to be fed into our model. We chose a vocabulary of 20,000 words with a maximum length of each comment to be not more than 100 words. The comments which were shorter than 100 words were padded with leading 0s and the comments with more than 100 words were trimmed.
The training was performed on the split train data in batches of size 1024, using Adam optimizer and categorical cross-entropy as the loss function.
GRU achieved the best performance among all the other models that we tried. The results are presented in section III.

TABLE II: Facebook pages that were used for scraping and their categories

News	**Entertainment**	**Sports**
CNN	Now This Entertainment	NFL
Fox News	Buzzfeed	NHL
The Young Turks	9Gag	NBA
		Arsenal

E. Running the Model on Facebook Data

Once we selected our best model, which was the GRU, we tested our model on Facebook comments. We scraped over 570,000 comments from the pages shown in table II.

To do so, we used a script taken from [7] and modified it according to our needs. The script uses Facebook Graph API to grab the posts published in each page for a specified time period, and then scrapes all the comments from those posts. The data is collected from comments users have posted on these pages during the period Jan 2017 - March 2018.
After collecting the data, we cleaned it up by removing unnecessary characters such as emojis, GIFs, pictures, and all non-ASCII characters. We did not apply the stop-word removal and lemmatization processes described in section II-C, because the data was used for the GRU model.

F. Toxicity Categorization

Once we identified the toxic comments among the scraped Facebook data, we aimed at identifying the types of toxicities that were present in each Facebook category. In particular, we looked for comments that were either sexist, racist, or homophobic. There are obviously many other types of toxicities, but we decided to limit the scope of our search for this project. To do our analysis, we first tried the Latent Dirichlet allocation method (LDA) for topic modeling. LDA is an unsupervised algorithm that works by going through each document (comment in our case) and randomly assigning each word in the document to one of the topics. Number of topics is decided at the start of the procedure. It then iteratively improves the clustering. We were unable to obtain meaningful clusters however. This could be attributed to the fact that the proportion of sexist, homophobic and racist comments out of all the toxic comments was pretty low (less than 15%, as shown in section III) and since the topic modeling was being performed on all the toxic comments, we did not get the relevant clusters. Eventually, we resorted to a naive implementation for identifying the types. We constructed comprehensive lists of sexist, homophobic and racist words and counted the occurrence of each word in the scraped data. Each category was then labeled depending on the toxicity type with the highest percentage. These results are also presented in section III.

G. Analysis and Visualization

Finally, we used all our developed models to gain insights about our data. We analyzed the amount of toxicity in each Facebook community (news, sports, entertainment) and categorized the toxicities. In order to be able to make inferences about the whole population (in this case Facebook

TABLE III: Comparison of Best Obtained Models

	Perceptron with TF-IDF	Random Forest with Doc2Vec	GRU with Word Embeddings
Precision	0.89	0.87	0.89
Recall	0.91	0.88	0.94
F-Score	0.90	0.87	0.91

as a whole), we performed hypothesis testing. The results are presented in the next section. To visualize these results, we used Matplotlib and Tableau.

III. Evaluation and Results

In this section, we present the results obtained from our NLP and machine learning models. We then show the analysis we performed on the scraped Facebook data.

The precision, recall, and F-score values for our best models were computed using scikit-learn for ternary data and are presented in table III. It can be seen that GRU with word embeddings gave the best results, and was thus selected as our final model for getting predictions from Facebook data. It should be noted that the Facebook data was not labeled and it was mainly used to derive insights about online social media platforms. Figure 5 shows the most commonly used words in the Facebook comments classified as toxic by our model and loosely verifies the effectiveness of our model.

Table IV shows the amount of proportion of toxic comments

Fig. 5: Word cloud made out of the toxic comments detected from Facebook Data

in each Facebook category. It can be seen that with the toxicity rate of over 16.4 %, News was much more toxic than entertainment and sports. In order to be able to generalize the results to the entire population (entire Facebook) from our sample data, we computed the confidence intervals. Due to the large number of analyzed comments (large sample size), the obtained confidence intervals are very tight. Equation 1 was used for computing the confidence intervals, where p is the sample proportion and n is the sample size [8].

$$\sqrt{\frac{p(1-p)}{n}} \tag{1}$$

Figure 6 shows the amount of racism, sexism, and homophobia found within each community out of all its toxic

TABLE IV: Toxicity amount and rate in each Facebook Category

Category	Total Number of Comments	Total Number of Toxic Comments	Toxicity Rate	95% Confidence Interval
Enterta-inment	189,452	8,692	4.711 %	±0.002 %
News	193,769	31,886	16.456 %	±0.003 %
Sports	190,986	9,385	4.913 %	±0.002 %

comments. It should be noted that the percentages reflect the ratio of such comments among the toxic comments, and not the entire data. The confidence intervals were computed using equation 1. It can be seen that the amount of sexist comments was much higher in entertainment compared to the other two types. In news and sports, however, racism was found to be more common. The amount of racist and sexist comments was higher in news compared to sports.

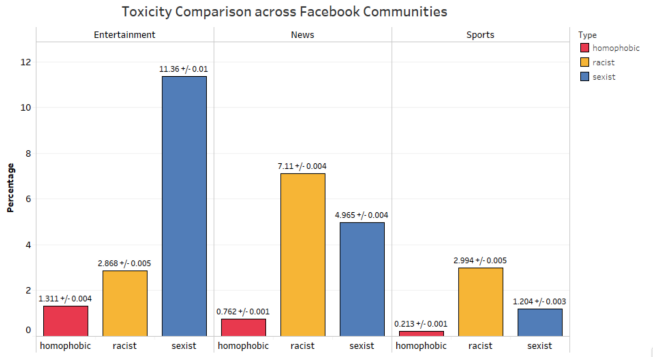

Fig. 6: Toxicity types and their proportions in each community

We performed a statistical test of proportion [9] and generalized our findings to the whole Facebook data as follows. We can safely assume that our sampled data, due to the long time-span of its publication (past 1.5 years) and its size (around 200,000 samples per category) is a good representation of the current trend in the whole Facebook.

- Amount of sexism is more than racism and homophobia in entertainment pages.
- Amount of racism is more than sexism and homophobia in news and sports pages.

In order to perform the hypothesis test, the types of toxicities were compared pair-wise to have a binary test. The Z-statistics is assumed to be normal [9] for each test and is computed using equation 2. For example, to analyze whether sexism is more common than racism in entertainment, the null-hypothesis would be that they are equal. The percentages for both categories are first normalized to add up to 100%. If the null-hypothesis is true, $p_0 = 0.5$. $\hat{p}$ is the actual normalized percentage of racism. For a 0.05 p-value, Z-statistics must be greater than 1.645 to reject the null hypothesis.

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \tag{2}$$

Lastly, we wanted to investigate how the amount of toxicity has been changing during 2017 and 2018. For this analysis, we needed a large amount of data from all months of 2017 and 2018, so we limited the analysis only to the CNN page, which was the most toxic page compared to all the other ones. We scraped over 37,000 comments for each month, totaling to over 600,000 comments, identified the toxic ones, and found the toxicity rate per month. The results are displayed in Figure 7. The horizontal dotted line shows the average amount of toxicity (around 14%) over the entire analyzed period, and the line chart demonstrates the variation in each month. It can be seen that the toxicity level was relatively close to average most of the times, except for a few months. In particular, November 2017 had the highest amount of around 20%

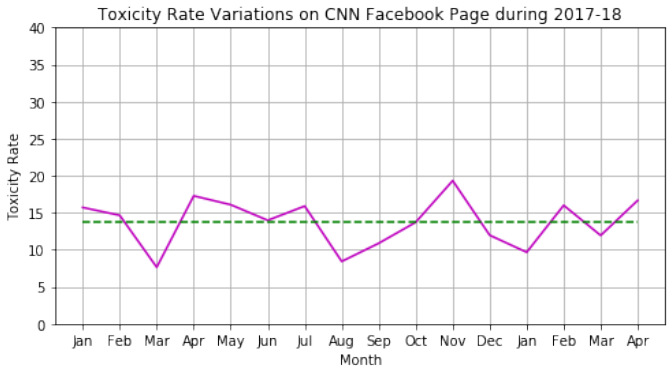

Fig. 7: Variations of toxicity rate(%) in different months of 2017-18

IV. Conclusions and Future Work

In this project, we used NLP and supervised machine learning techniques to come up with a model for detecting toxic comments. We trained our models on datasets from Wikipedia and SOCC and explored TF-IDF, Doc2Vec, and word embeddings to featurize them. We tried several machine learning models, and found GRU RNNs to perform the best on the validation set. We used the model on the data we scraped from multiple sports, news, and entertainment Facebook pages. Among the comments classified as toxic, we identified those that contain racism, sexism, and homophobia. News pages were found to be most toxic, whereas news and entertainment were similar. The most prevalent type of toxicity in news and sports was racism and in entertainment sexism. With the help of statistical hypothesis tests, this analysis can safely be extended to the whole Facebook data. There are several directions at which this project can be extended towards. An immediate extension would be crowd-sourcing the Facebook data to be able to evaluate the model performance in an objective manner. Conducting supervised learning for toxicity categorization is another aspect that can be explored, since it should achieve more accurate results. It would also be interesting to compare the toxicity rates and types across different social media platforms (Facebook vs. Twitter for example). Identification of bots, trolls, and spammers is yet another possible extension which could provide invaluable service to these online platforms.

Appendix: Challenges and Learned Lessons

The first challenge was lack of good dataset. While the Wikipedia dataset is rich, the SOCC dataset contains only 1,000 comments which don't contain much toxicity. Ideally, we would want to use three high large datasets.

Another challenge was the model selection process. Training took time and extensive tuning was required to achieve good results. The large size of data after featurization made it hard to train the models on our own systems, so we ended up reimplementing some of the scripts using Spark to run on the cluster.

Despite all the challenges, we managed to obtain a good model and perform interesting exploratory analysis on the scraped data and to reach there, we learned about several NLP and machine earning techniques. While we were familiar with TF-IDF, methods such as bag of words and Doc2Vec were new ideas to us. Learning about RNNs was also interesting, given their power and importance in text analysis. We also got a good experience about how an end-to-end data science pipeline looks like, even though more practice is required since the range of technologies at each stage is very wide.

Acknowledgment

We would like to thank our instructors for the CMPT 733 course, Dr. Jiannan Wang and Dr. Steven Berner for their excellent teaching and lab material that they provided us with during the course of the semester, as well as the help they gave us at different stages of performing the project. We also would like to thank our TAs, Hiral and Simranjit, who were always available for help and for their feedback during the poster session.

References

[1] *'11 Facts About Cyber Bullying'*. Retrieved from DoSomething.org
[2] *'Toxic Comment Classification Challenge'*. Retrieved from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
[3] *'SFU Opinion and Comments Corpus'*. Retrieved from https://github.com/sfu-discourse-lab/SOCC
[4] *'Google Perspective API'*. Retrieved from https://www.perspectiveapi.com/
[5] *'Natural Language Toolkit'*. Retrieved from https://www.nltk.org/
[6] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).
[7] *'How to Scrape Facebook Page Posts and Comments to Excel (with Python)'*. Retrieved from https://nocodewebscraping.com/facebook-scraper/
[8] *'Confidence Intervals for a Population Proportion'*. Retrieved from https://onlinecourses.science.psu.edu/stat100/node/56
[9] *'Test of Proportion'*. Retrieved from https://onlinecourses.science.psu.edu/statprogram/node/164