Identifying Toxicity in User Comments

*

Himanshu Punetha
MT17016
IIIT Delhi

Suman Kumar
MT17060
IIIT Delhi

Vaibhav Garg
MT17064
IIIT Delhi

***Abstract*—While interacting on online platforms, people tend to use toxic words in order to show their aggression. These toxic words can be categorized into obscene, threatening, insulting words etc. Nowadays, online platforms have abundance of such toxic comments which can lead to cyber-bullying or online harassment of users. Thus there is a need to identify such toxicity in comments and prevent them from being posted on online platform. This project aims to address this problem by identifying the toxicity of comments.**

***Index Terms*—word2vec, tf-idf, nlp, social media**

I. Problem Statement

The aim of this project is to classify a given comment into different classes of toxicity (Toxic, Severe Toxic, Obscene, Threat , Insult and Identity Hate). This is a multilabel classification problem in which one comment can be classified into multiple classes or may belong to none of the classes.

II. Literature Review

Various reserachers have been working in something similiar in the past. Joni et al. worked on identifying and classifying hate in online news media and used multiple machine models such as Linear SVM, Logistic regression etc. on TF-IDF features to solve similar multi-label classification problem [1]. Srivastava et. al. worked on Identifying Aggression and Toxicity on Kaggle dataset and proposed a classifier based on Capsule Net [2].

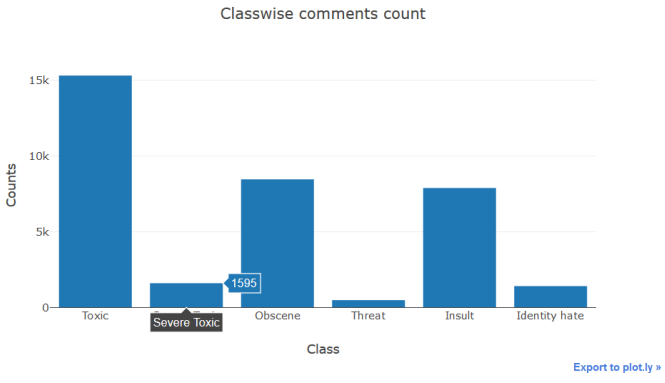


Fig. 1. Classwise count

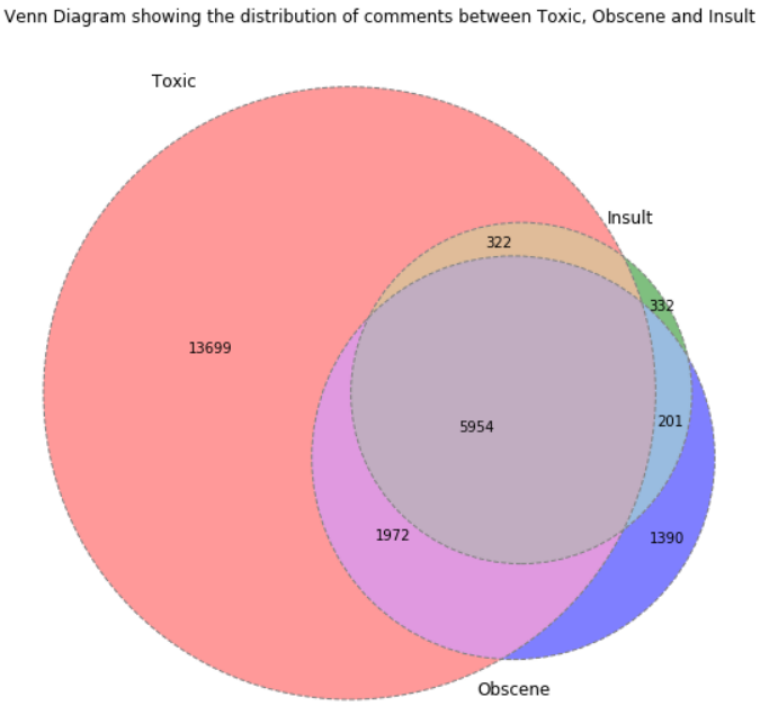


Fig. 2. Venn Diagram

III. Dataset details

We are using data-set from Toxic comment classification challenge hosted on kaggle website. Dataset consists of a training and a test set containing 159571 and 153164 records respectively. Each record contains a comment and a binary value in each of the next six columns depending on whether it contains a specific type of toxicity or not. This is a multilabel classification problem, so multiple toxicity could be present simultaneously or all might be absent for a comment. From our analysis of train set, we observed that most of the comments didn't contain any toxicity.

IV. Feature Engineering

We have used tf-idf and word2vec embeddings for feature generation from the comments. For the given dataset our vocabulary size was around 180K and therefore the tf-idf vectors

for each comment is very high-dimensional (180K features). As tfidf features doesn't keep the contextual information, thus we also trained the word2vec model on the given text data and computed the word2vec embeddings of each comment by summing and averaging individual word vectors. Apart from using tf-idf and word2vec, we have included some other feature engineering techniques which are discussed below:-

After going through the dataset, we observed that comments having toxicity tends to follow different pattern than normal comments. For instance, toxic comments tends to have more Capital letters compared to normal comments and also occurrence of special symbols like (@#%) is more. From our observation, we devised some features such as comment length, Normalized count of Capital letters, Special symbol counts, Unique words count etc. and also checked their correlation with their classes to check whether classes could be uniquely identified using those features. We found that Normalized capital count and Unique word count have maximum variance across all classes. So we used these features in conjunction with tf-idf later for classification tasks.

We also used some other filter based feature selection techniques such as chi-square statistic, entropy on the tf-idf features to select K-best features from the high - dimensional features of tf-idf. To find suitable value of K, we varied K and plotted a graph for K vs the training error and validation error and found that best value of K was between 20K - 30K considering different classes.

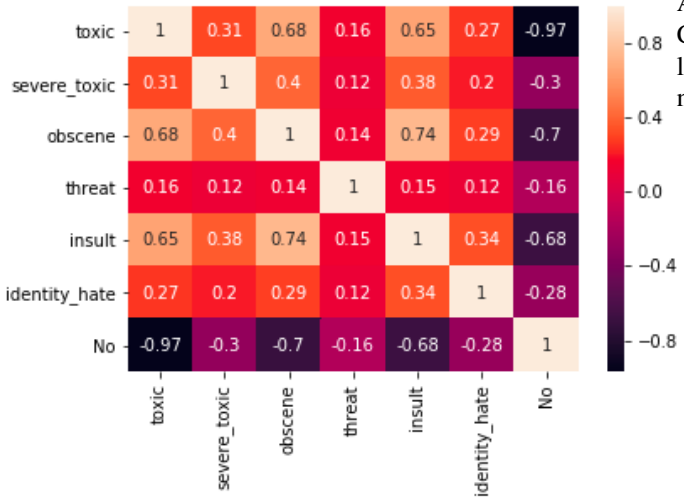


Fig. 3. Correlation between different classes

V. Methodology

We performed analysis of both tfidf and word2vec features by training classifiers on both of them. Precision and Recall were used as the metrics. As identifying majority of toxic comments is more important therefore False Negatives count must be lesser, thus recall metric is of higher significance to us.

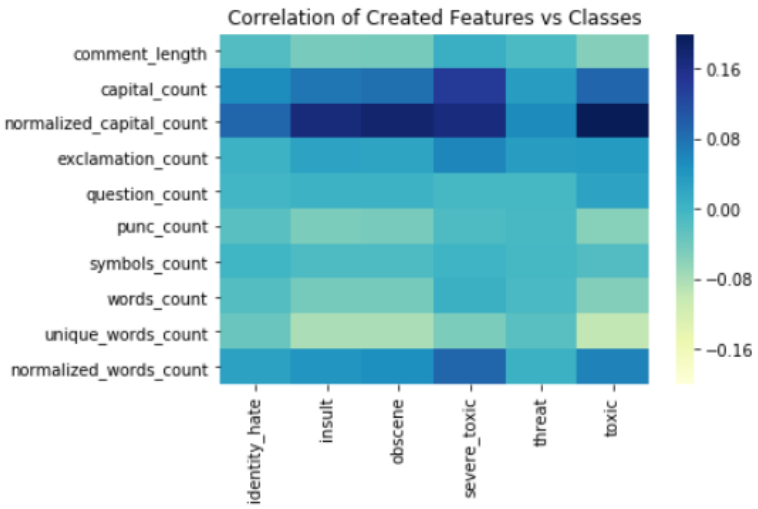


Fig. 4. Correlation between features and class

We have used multiple classifiers such as Linear SVM, Logistic regression, Neural Networks, CNN etc. Logistic regression was used for both the tf-idf and word2vec features and since it performed better for tf-idf, we later implemented Linear SVM classifier on tf-idf features. We also visualized the features obtained by word2vec after reducing their dimensionality (using PCA) for each class label. All the other feature engineering techniques were also done on the tf-idf features. As an exception to above, Neural Networks model such as CNN were implemented for word2vec features since it had less features to work with (100 features) and training neural networks is computationally heavy.

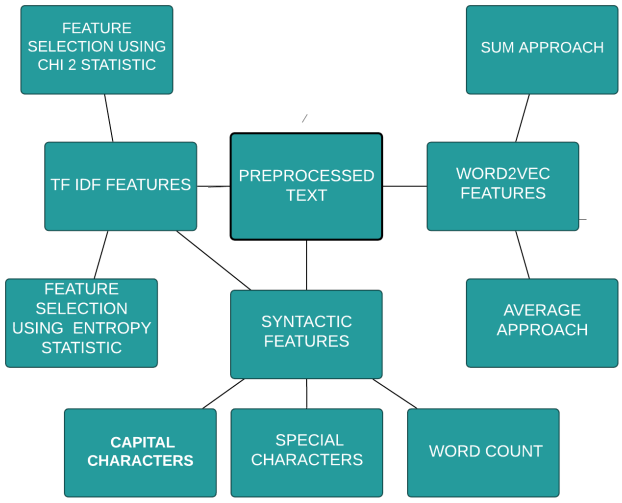


Fig. 5. Methodology

As the data is highly unbalanced for each of the classes, thus we adjusted the class weights while training a separate classifier for predicting each. We then compared our results

Techniques	toxic	severe toxic	obscene	threat	insult	identity hate
Precision for unbalanced class weights	82.13	54.52	82.34	46.99	74.84	42.13
Recall for unbalanced class weights	55.44	25.62	55.19	5.96	44.62	8.29
Precision for Balanced class weights	45.49	13.81	38.17	2.33	33.77	6.72
Recall for Balanced class weights	85.38	92.85	87.54	87.48	87.39	89.67

a

TABLE I
WORD2VEC RESULT FOR AVERAGING WORD VECTORS (PERCENTAGE)

Techniques	toxic	severe toxic	obscene	threat	insult	identity hate
Precision for unbalanced class weights(LR)	98.35	60.44	97.20	0	87.07	88.25
Recall for unbalanced class weights(LR)	31.13	7.00	34.84	0	24.59	1.30
Precision for Balanced class weights(LR)	60.80	27.42	66.94	5.83	52.69	16.84
Recall for Balanced class weights(LR)	83.30	87.02	86.13	81.22	85.04	81.09
Precision for LinearSVM Balanced class	71.57	33.09	76.13	46.59	62.73	37.72
Recall for LinearSVM Balanced class	80.78	60.91	83.15	48.73	77.40	54.28

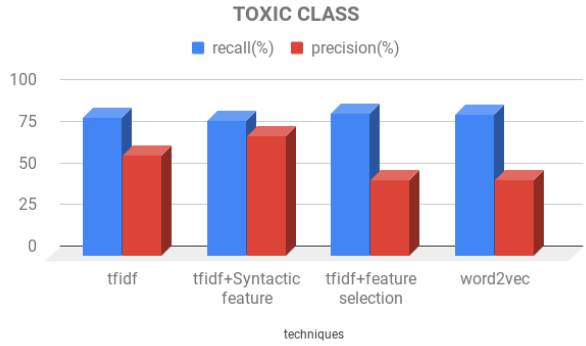
a

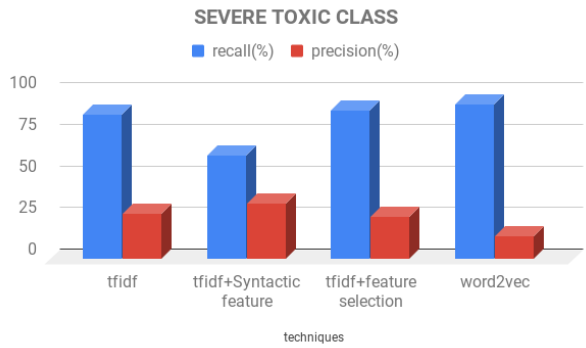
TABLE II
TF-IDF RESULT (PERCENTAGE)

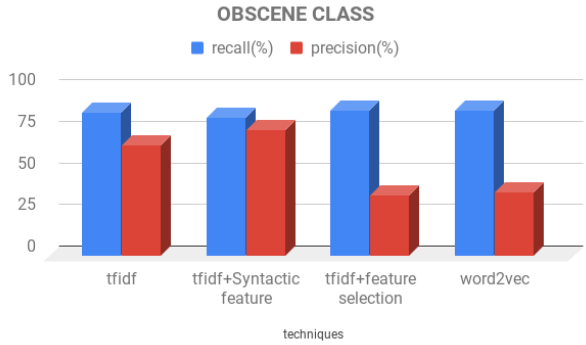
without balancing the class weights and with balancing the class weights. We compared these results in table 1 and 2.

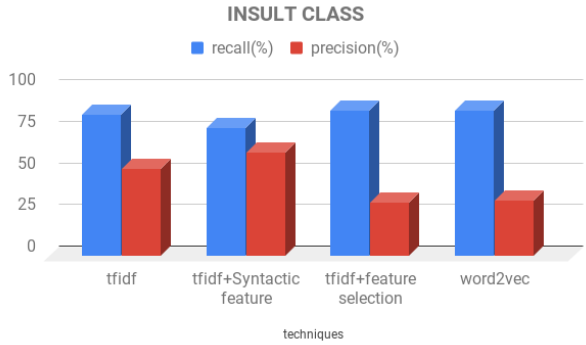
After selecting the BestKFeature from chi-square statistic, we applied the Logistic Regression and SVM classifiers on the resulting features. We also repeated this process for entropy feature selection. Due to limitation of resources, we could only apply entropy feature selection on top 1000 tf idf features and then trained classifiers on them.

VI. RESULTS AND INTERPRETATION



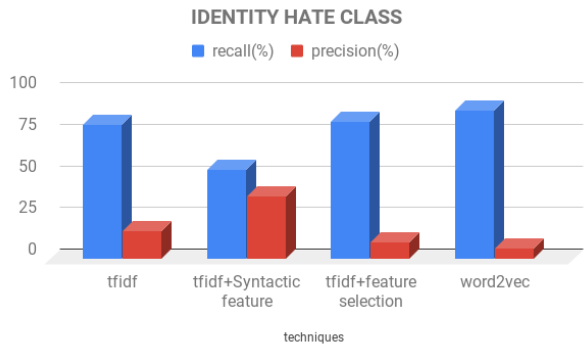






After statistical analysis, we got to know that for each class label, the datapoints in negative class (representing non-toxic comments) were a lot more than the positive points. Moreover, severe toxic class is a subset of toxic class whereas obscene, insult and toxic classes have high correlation with each other. By the class wise visualization of data, we inferred that the tokens(having high frequencies) of threat and insult class are bit different from those of other classes.

After training logistic regression on tfidf features on imbalanced data, we get low recall and high precision for each class. Since threat comments were only 0.3% of the total samples,



for threat class model predicted all samples as non toxic, leading both precision and recall to zero. Even for identity hate class, the recall score was only 1-2%, because the identity hate comments were only 0.8% of the total samples.

We then did the same experiment of tfidf vectors after maintaining class balance. We noticed that for each class, there is significant increase in false positives and decrease in false negatives. In other words, after changing weights model was predicting positive class more frequently such that recall increases significantly and precision deceases a bit.

A similar pattern was observed while training logistic regression on word2vec features.

We also applied Linear SVM on tf-idf features and found that accuracy was improved when compared with logistic regression classifier but since we focus more on High Recall for our use-case and also since data is imbalanced for all the classes when compared to the non-toxic comments, therefore we preferred results of Logistic regression over SVM. We also observed that on adding two additional syntactic features (Normalized capital count and unique words count) to the tf-idf features, the accuracy of the classifiers improved slightly.

After applying Neural Neworks and CNN, we observed that the result weren't very promising. This could be attributed to the fact that any balancing using class-weights wasn't done and we only applied these methods on word2vec embeddings and not on tf-idf features. However,we did perform SMOTE analysis for upsampling the minor class but results were still not satisfactory. Since the minor class in some of the cases is only 0.3% of total, therefore SMOTE didnt peform very good.

In general, we observed that word2vec and tfidf+feature selection gave higher results of recall value of each class. We think that word2vec features capture contextual information and thus were able to produce good results as that of tfidf+feature selection. Moreover, in many classes tfidf features along with syntactic features gave slightly higher precision values however compromising slightly on recall score as compared to the original results obtained for tf-idf features.

VII. Individual Contribution

Himanshu has done data visualization and extracted syntactic features to train different models. Suman has implemented 2 approaches of word2vec, he also implemented CNN approach. Vaibhav implemented tfidf approaches along with different variants of feature selection. We all then repeated these experiments with different classifiers and different data(balanced and unbalanced).

References

[1] Joni Salminen, Hind Almerekhi, Milica Milenkovi, Soon-gyo Jung, Jisun An, Haewoon Kwak,Bernard J. Jansen.Anatomy of Online Hate: Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media.Proceedings of the Twelfth International AAAI Conference on Web and Social Media.

[2] Srivastava, Saurabh, Prerna Khurana, and Vartika Tewari. "Identifying Aggression and Toxicity in Comments using Capsule Network." Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). 2018.

[3] Dataset:https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data