TOXIC COMMENTS CLASSIFICATION

PROJECT REPORT

Aarav Varshney and Argha Chakrabarty

Ashoka University

December 2021

Contents

1 Introduction

The project presents an application of NLP to classify texts posted on online platforms as comments as either toxic or non-toxic. While social media is a great medium for sharing opinions, it is more and more being used to bully and harass people. This project aims to classify such comments and hopefully make social media a bit more welcoming. We study the impact of tf-idf and SVMs, Decision Trees, Naive Bayes and CNNs. We evaluated our approaches on comments from the Kaggle Toxic Comments Classification Challenge. Additionally we host the classifier so that users can interact with it.

2 Literature Survey

This are several papers already published on the topic of Toxic Comments Classification. This was helpful for us to understand the topic and the way experts have approached it. We have used the following papers for our analysis:

2.1 Toxic Comment Classification by Zaher et al.

The authors used Naive Bayes (as benchmark), RNN and LSTM to classify toxic comments. They trained the classifier on the same kaggle dataset.

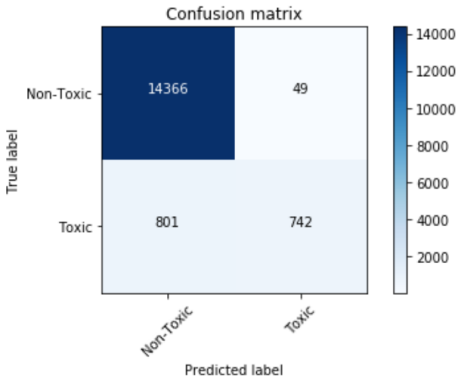


Fig. 5. Evaluation of Naïve Bayes algorithm using confusion metrics.

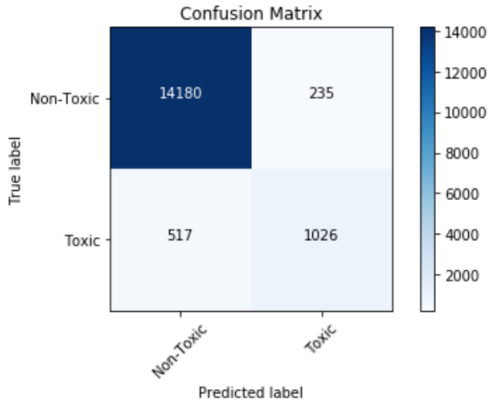
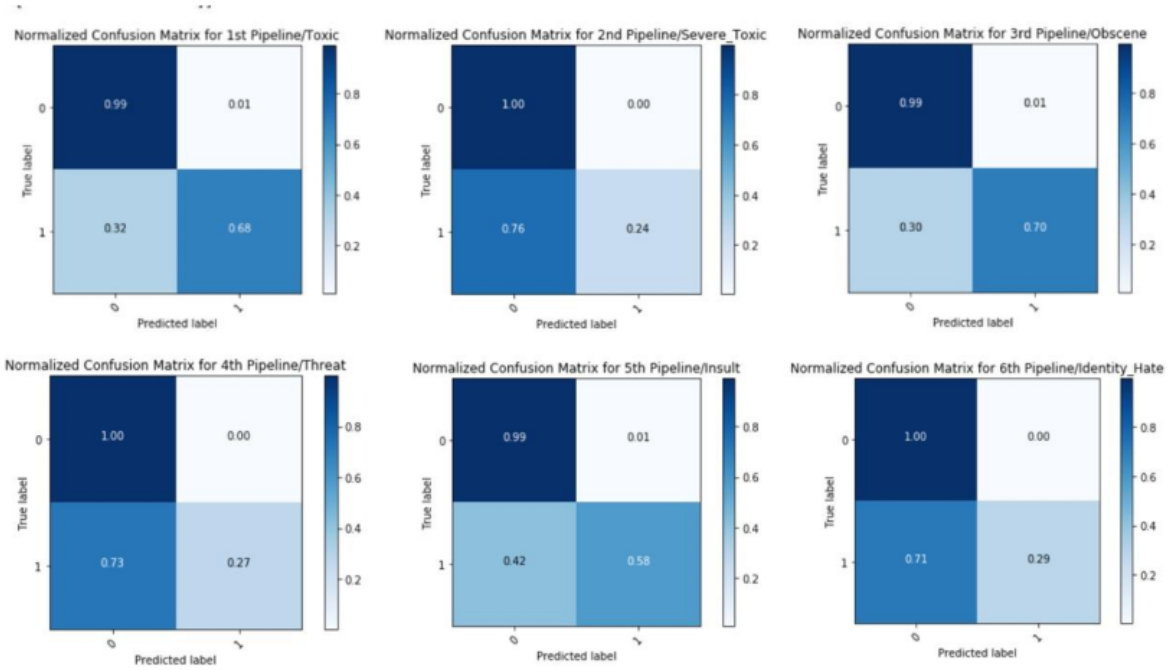


Fig. 6. Evaluation of LSTM algorithm using confusion metrics.

The results showed that LSTM has an almost 20% increase in True Positive Rate (TRP) which means that among all the identified toxic comments LSTM has 20% more sensitivity in correct assignment of the comments. Furthemore there was an increase in Recall and F1 score when using LSTM.

2.2 A Machine Learning Approach to Comment Toxicity Classification by Navoneel Chakrabarty

The authors solved the problem of multi-label classification rather than binary and further classified toxic comments as hate, toxic, severelytoxic and so on. Two approaches were used for 3 labels each out of 6 labels. The authors used Bag of words using Word count vectorizer and then the tf-idf transformer which is finally used to train. Support Vector Machine (SVM) classifier with 100 as maximum iterations and C as 1.0. Whereas for the other three labels, decision tree classifier was used instead.

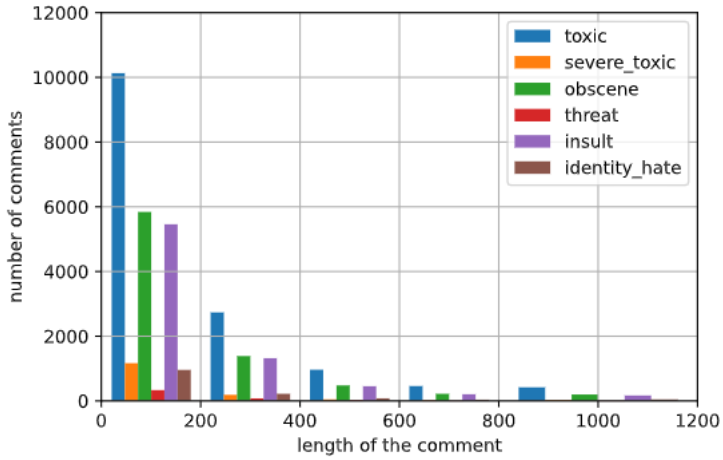


The paper claims that this apporach performs better than LSTM and Naive Bayes and the results validates this claim. We want to validate this claim on binary classification.

3 Project Description and goals

4 Dataset Specification

Our dataset had 159571 comments, with each comment labelled as toxic, severe toxic, obscene, threat, insult, identity hate or none of 'em. The length of each comment varied from 10 words to 1200 words. To make it easier for us to train the model we chose comments with word count less than 600 as 132327 comments were under 600 words, and the number seemed good enough for the purposes of this project.



Due to various constraints we decided to go with a binary classifier rather than a multilabel classifier. For this purpose, we created added a new column, istoxic which labels itself 1 if the comment is toxic and 0 if it isn't. The dataset contained a lot of punctuations and numbers which didn't really contributed to the overall

sentiment or toxicity of the comment and hence were removed.
Afterwards we first lemmetized, and then stemmed each comment and removed any stop words present.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

5 Methods, experiments and explanation, schematics

6 Implementation details

After the data is read, we select only comments with less than 600 words in it, and label each comment as toxic or not depending on the preexisting labels. nltk's wordnet module is used for stemming and lemmetizeing each comment after removing all the punctuations and numbers along with any other redundant utf-8 characters. After all the comments are cleaned, we use Tfidf word embedding to vectorize our comments so that we can use 'em for training our model. TfidfVectorizer is used as is from the scikit-learn module. The test to train split ratio is 3:1. Then we trained our model on various models, nameley SVM, Logistic Regression, Naive Bayes and decision trees. We trained the model on a laptop with i5-8520U and 16 gigs of RAM. We also trained a multi-label model using Binary Relevence from scikit-multilearn module which is built on top of scikit-learn eco-system and provides multi label classification. After training, we decided to deploy our model for demo purposes. Since we used python for our project, we decided to go with a simple FastAPI server. We created a pipeline with two stages, the first one vectorizing any input text and at the stage the actual fitting or prediction occurs. To save the model we pickle the pipeline object using python's in-built pickle library and write the raw binary data onto a file. Afterwards, the file is un-pickled at the server-side and a single string array is given to it for prediction. The API takes a single string as it's payload and returns the predicted outcome as it's response. The user has to create a POST request with the comment as the payload and we return whether the comment is toxic or not in the response body. Lastly we deployed the whole thing in a heroku server. You can take a look at the API at this link :- https://iml-tox-det.herokuapp.com/docs (please keep in mind that we are using a free server, it might take some time to start). We also went on and created a frontend website that uses the api: https://adoring-bell-52f360.netlify.app (the name is generated by netlify)

7 Observations

7.1 Binary Classifier SVM

Confusion Matrix:

```
array([[29258,   189],
       [ 1263,  2372]], dtype=int64)
```

Other metrics

recall	precision	f1 score
0.926	0.653	0.766
0.926	0.653	0.766

7.2 Multilabel Classifier SVM

7.3 Logistic Regression

Confusion Matrix:

```
array([[29304,   143],
       [ 1425,  2210]], dtype=int64)
```

Other metrics

recall	precision	f1 score
0.939	0.608	0.738
0.939	0.608	0.738

7.4 Naive Bayes

Confusion Matrix:

```
array([[29432,    15],
       [ 2646,   989]], dtype=int64)
```

Other metrics

recall	precision	f1 score
0.985	0.272	0.426
0.985	0.272	0.426

7.5 Decision Trees

Confusion Matrix:

```
array([[28591,   856],
       [ 1055,  2580]], dtype=int64)
```

Other metrics

recall	precision	f1 score
0.751	0.71	0.73
0.751	0.71	0.73

8 Conclusions and future directions

9 Examples of citations, figures, tables, references

9.1 Citations

Citations use `natbib`. The documentation may be found at

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing (Kour and Saabne, 2014a; Hadash et al., 2018) but other people thought something else (Kour and Saabne, 2014b). Many people have speculated that if we knew exactly why Kour and Saabne (2014b) thought this...

9.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

9.3 Tables

See awesome Table 1.

The documentation for `booktabs` ('Publication quality tables in LaTeX') is available from:

https://www.ctan.org/pkg/booktabs

9.4 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014a.

Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014b. doi:10.1109/SOCPAR.2014.7008025.

[1] Sample of the first footnote.