Toxic Comment Classification Using Neural Networks

Emmanuel Otubo
Mathematics and Statistics
Auburn University
eeo0010@auburn.edu

Ukamaka Nnyaba
Mathematics and Statistics
Auburn University
uvn0001@auburn.edu

Chinedu Eleh
Mathematics and Statistics
Auburn University
cae0027@auburn.edu

Ekene Aguegboh
Applied Economics and Rural Sociology
Auburn University
esa0013@auburn.edu

Abstract

Machine reading comprehension, question answering, sentiment analysis, and named entity recognition are essential tasks in natural language processing. In a past Kaggle competition titled 'Toxic Comment Classification Challenge', competitors are challenged to build a multi-headed model that is capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. We will be using a dataset of comments from Wikipedia's talk page edits. Improvements to the current model will hopefully help online discussion become more productive and respectful. In this project, our goal is to create a neural network classifier for sequence classification to build a model that is capable of detecting different types of toxicity as mentioned above. The dataset can be found here `https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data`. We hope to compare our result to the baseline model which used a NB-SVM (Naive Bayes-Support Vector Machine) to create a strong baseline for the Toxic Comment Classification Challenge competition. This baseline model can be found here `https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline`

1 Introduction

A fundamental feature that has set humans apart as a species is entails the ability to cultivate sophisticated forms of communication. Over time, communication skills amongst humans have evolved and advanced into more fiddly arena of virtual communication. In the post 2020 era, human interactions and activities become increasingly virtual/social-media based to the extent that lots of people across different generation have had to readjust their lives to this new norm. Unlike in the past, a person can access millions of people around the world just by a single click. While this is amazingly wonderful in many respects, it comes with certain social costs. One of such costs is the abuse of social medial to disparage other users. Among other things, such abuse have taken the forms of cyber-attacks and cyber-bullying, leading toxicity in the social media space. Therefore, every online impression made exposes users to threats, abuse and harrassment, which makes that many people quit self expression and seldom seek different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

The focus of this project is to analyse the use of toxic words in the social media. We do this by focusing on one of the primary sources of abuse, which entails the use of sentimental words to constitute nuisance in the social media space.

1.1 Objective

The objective of this project is to conduct sentiment analysis on a dataset of comments from Wikipedia's talk page edits. We do this by building a multi-label classification model using neural networks to detect different types of toxicity like threats, obscenity, insults, and identity-based hate.

This objective is relevant as it provides a basis for comparing our results with already existing approach such as the Naive-bayes learning. The accuracy of our result would serve as threshold to measure the performance neural network in conducting sentiment analysis.

2 Data Description / EDA

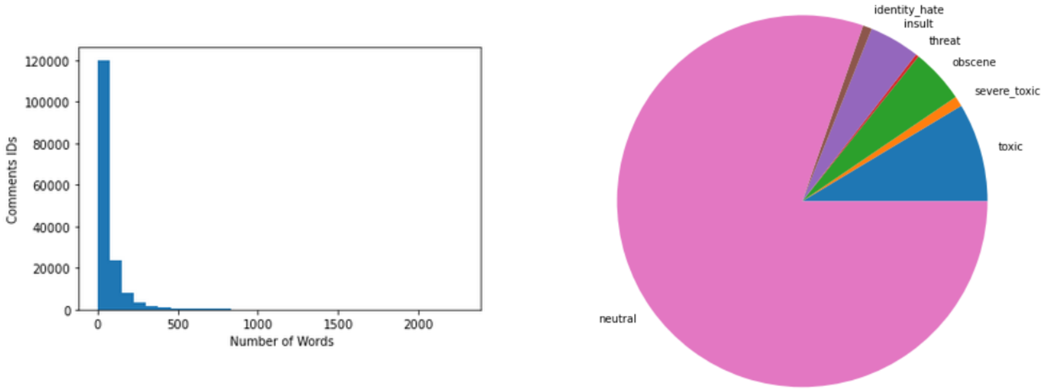


Figure 1: Features visualization

The data for this project was collected from the Wikipedia's talk page edit. The data constitutes of a structured dataset made up of 159,572 rows and 7 columns. The first column is an ID number, which uniquely identifies each observation. The second column is a comment-text, which contains the text that we plan to classify.

Columns three through eight contains six labels. These labels includes: ”toxic”, ”severe-toxic”, ”obscene”, ”threat”, ”insult”, and ”identity-hate.” To give the dataset some sense of balance, we include an artificial label titled: ”neutral.”

Therefore, each observation contains an ID number, a comment-text, and the six labels, where the entry if 1 if the observation is classified as that label and 0 if the observation is not classified as that label. There are rare instances where the observation are classified under more than one label.

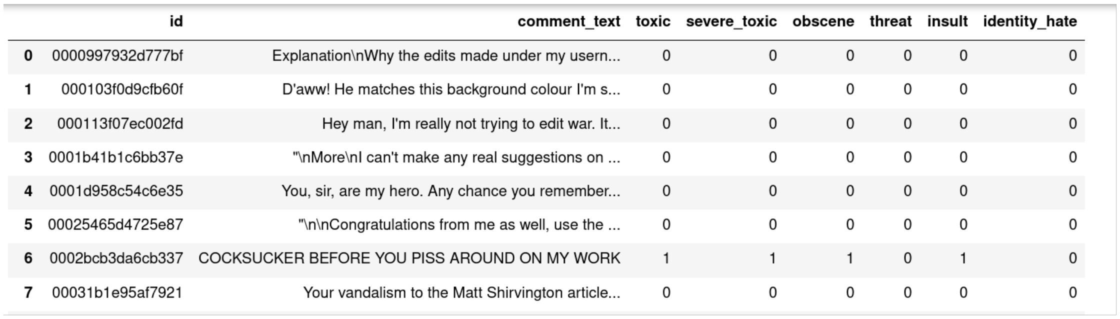
See the above, a histogram and pie chart that further describe the data. The histogram indicates the number of words in each comment. Accordingly, we find that the longest comment has 2,273 words.

The data set is highly noisy. Characters such as \n, $, #, =, *, ;̂ , ?, !, /] are rampant in the comments. In processing the data, we split the comment into words by these characters \n, $, #, =, *, ;̂ , ?, !, /], in addition to splitting by spaces.

The bar chat on the other hand shows a distribution of the seven classes of words including the neutral class that we created. As the chart suggest, the neutral class has the highest representation (80%) in the comments, which typically reflects real life scenario where non-sentimental words typically populates usage by people in all forms of communication. The neutral section is followed is by the word ”toxic” at 25%. Then we have insult and obscene at 7% and 8% respectively. Then the words, severe-toxic, identity hate and threat had the lowest percentages at....

2.1 Data Analysis

The data set appears as a multi-label problem. That is, there are comments with more than one label. An example is the comment with *id 0002bcb3da6cb337* as shown below. It is classified as *toxic, severe_toxic, obscene* and *insult*, an evidence of *multi-collinearity*, as shown in the correlation plot. To account for this in the model, we use the *power set* method to split the data into disjoint sub-classes. This results into a multi-classification problem with 41 unique classes.



	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	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0

Figure 2: Typical Comments

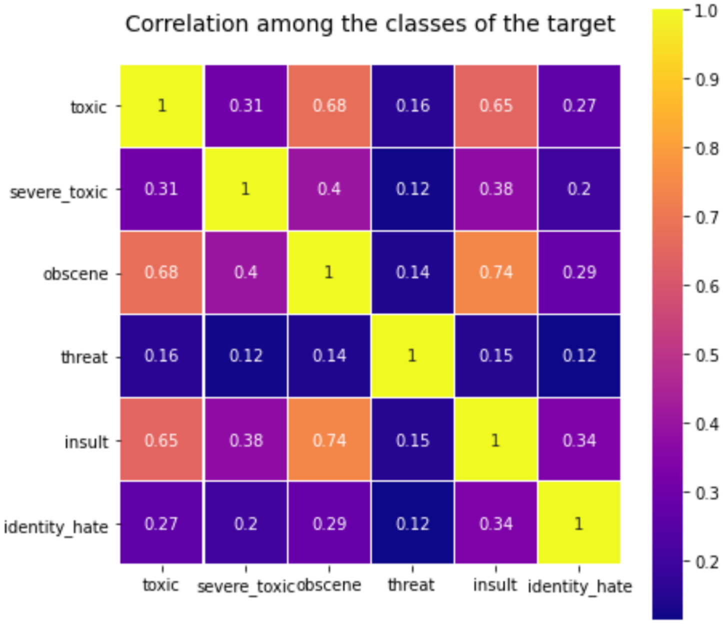
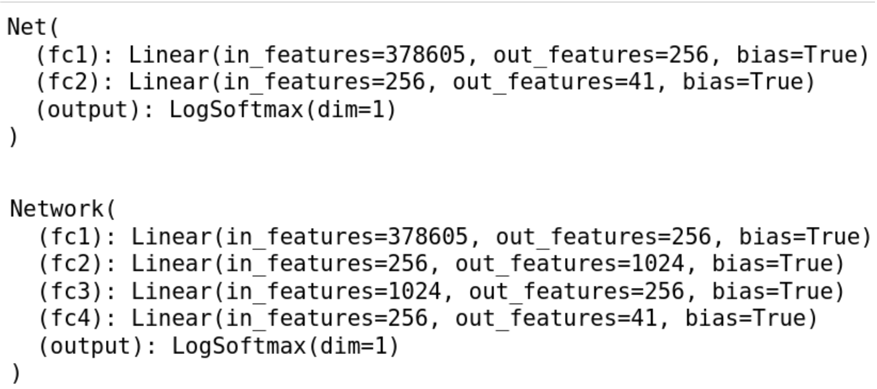


Figure 3: Correlation plot

As evident in the figure above, we see that certain words exhibit pairwise correlation. They include obscene and toxic, insult and toxic, and insult and obscene. To address this problem of high correlation, we employ the Power Set method. This technique allows us to split the set of label combinations in order to generate unique labels. By extension, we also generate unique independent classes, which are useful for setting up the multi-class classification model with target labels and new classes from the Power Set method. A case in point in our analysis is that instead of classifying into the original problem into seven classes, it if reformulated into a 41 level classification problem.

2.2 Neural Networks Architecture

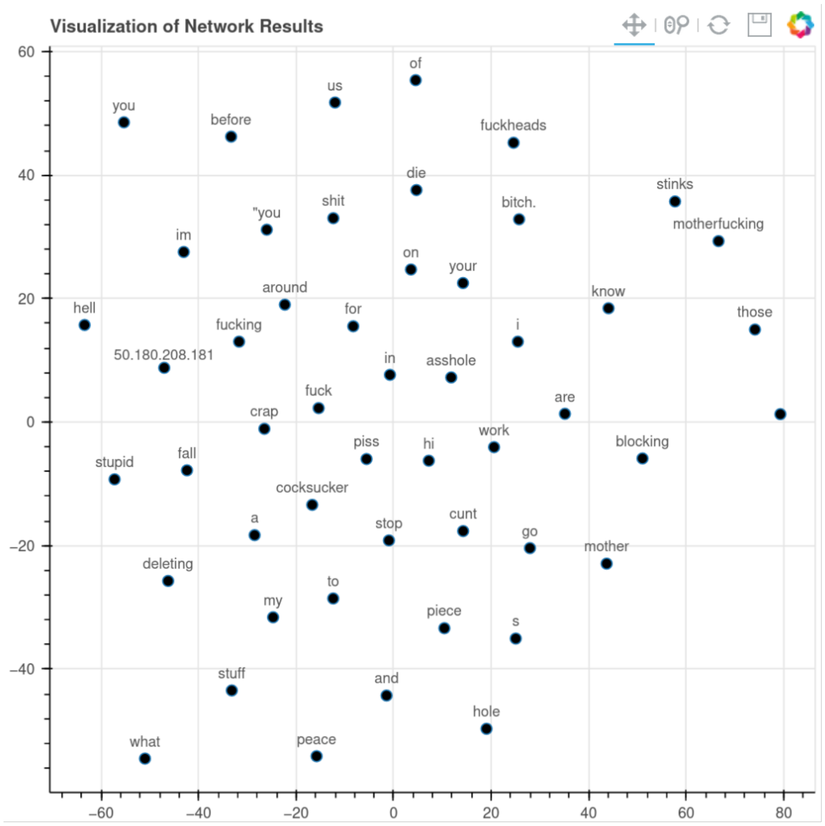
An input to the network will be a vector of numbers. We create word embedding through *word2vector*, which converts comments to vectors of 0's and 1's. This results into a huge *corpus*, a 378605 input nodes to the network. Training and tuning the network evidently became problematic.



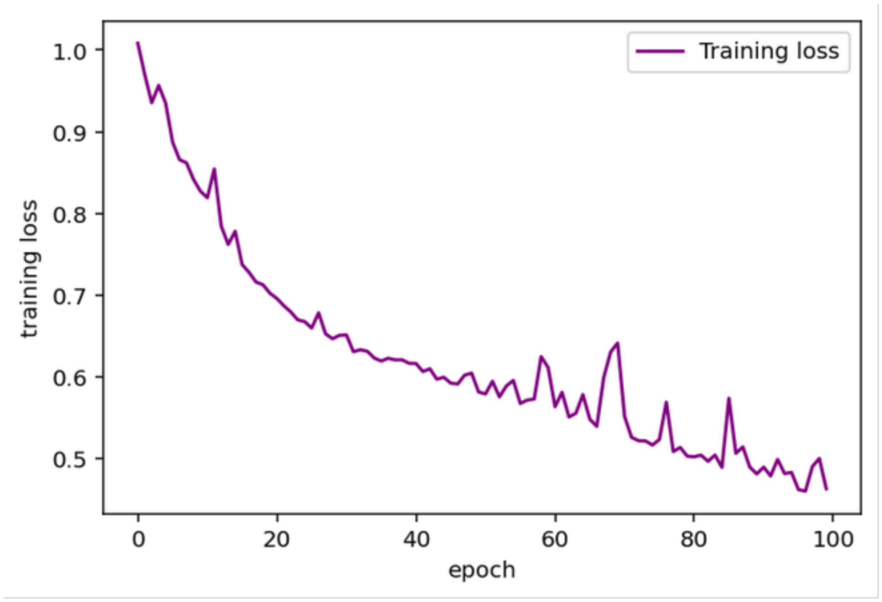
```
Net(
  (fc1): Linear(in_features=378605, out_features=256, bias=True)
  (fc2): Linear(in_features=256, out_features=41, bias=True)
  (output): LogSoftmax(dim=1)
)


Network(
  (fc1): Linear(in_features=378605, out_features=256, bias=True)
  (fc2): Linear(in_features=256, out_features=1024, bias=True)
  (fc3): Linear(in_features=1024, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=41, bias=True)
  (output): LogSoftmax(dim=1)
)
```

To alleviate this, we used 10% of the data set to train a network with the architecture only two fully connected layers `fc1` and `fc2`, with one hidden layer of 256 nodes and a ReLU activations. This took about 3 days to train on a GeForce GTX 1060 6GB. We then used this network weights and biases as pretrained model to our main model, turning off backpropagation on these pretrained weights. It is network with 4 fully connected layers `fc1, fc2, fc3` and `fc4`, two hidden layers and ReLU activations in between the layers, as shown below. The network speed was impressive.

We could have used standardized transfer learning packages, like *BERT* to do this. However, our goal was to use this medium to explore and understand the working principles of such high performing state of the art models.



The above plot shows a word vector space of the pretrained model. This provides an insight into what the weights are learning. By reason of interpretation, words that are closest posses the same sentiment value. In the middle of the diagram, toxic comments seems to be highly concentrated because they have the same level of sentiment. The left and right of the diagram, we find that less sentiment words seems to be sparsely populated e.g.: you would see words like ”you”, ”what”, ”deleting” ”know” and ”those” on the left and right extremes of the diagram. To achieve the purpose of training the model, we simply filter out the words that lack sentiment.



We had an accuracy of 89% on this model, as compared to the 82% that was achieved by the winning team, with *naive bayes* on the *kaggle* competition. Also, we see an exponential decay in training loss. The above plot indicates the training loss of our model as we implemented it over several epochs. The loss started from one and continued to diminish until we attained levels near/below 0.5. The outcome of our training is typical of a neural network, and the overall performance exceeds that of the Naiyes Bayes approach implemented in the Kaggle Competition.

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.