Toxicity Detection in Twitter Posts

CSCI 6509: Natural Language Processing
Project Statement

B00812651: Naina Nijher
B00808427: Nirav Solanki

Problem Statement

Due to the exponential increase in use of the internet across the world, the influence of the internet over the population has also increased manifold. The social media has a tremendous impact politics and businesses. The effects of the internet, however, have not all been positive. It has led to increase in cyber bullying and online harassment, an invasion in the privacy.

Platforms such as Facebook, Twitter and Instagram have users across the world spending an average of 2 hours per day liking, sharing, reposting, updating and tweeting. As social media brings together people from different cultures, faiths and educational background, keeping the domain healthy is of utmost importance, so that no one gets hurt.

To do so it is important to identify toxic content present in social media. For our project we have chosen to identify toxic tweets present in a user's tweeter profile. Our aim to identify tweets that can be categorized as obscene, threat, insult or identify hate.

Possible Approaches for the Project

We have decided to follow a supervised learning approach to build a model and solve this issue. To begin, we needed a labelled dataset to train our model. A dataset comprising of Wikipedia comments, that had been classified into categories such as obscene, threat, insult or identify hate, is available on Kaggle. To build the Classification Model, we will be using this dataset [1].

The second step is to perform the data pre-processing steps starting with tokenizing the sentence into words, which is followed by removing the stop words. The objective is to remove non-alphabet characters also, hence, improving the accuracy of the trained model. The next step would be to perform stemming and lemmatization on each word. The plan is to build the model in python and majority of the data pre-processing steps would be performed using the NLTK library [2].

We have not finalised the specific supervised algorithm that we would be using to build the model yet. Some of the few algorithms that we are still exploring to implement are:

- Naive Bayes Algorithm
- Support Vector Machine
- Random Forest Tree

If time permits, we plan on implementing a few algorithms to test and compare the accuracy of these models for such a problem. The models will be tested using the Test dataset, also provided on Kaggle.

Once we have a model with acceptable accuracy, we will use it classify tweets from live tweeter feed. Using Tweepy API, the tweets will be fetched associated to a user profile [3] and will be fed to the model. The tweets will be segregated under the afore mentioned categories.
As a conclusion to the project, two visualizations will be presented – one of them will show a comparison between the number of tweets that were toxic against the number of tweets that weren't, and the second will visualize the classification of the tweets under the various categories.

Milestones: Project Plan

Tasks	Status
Data Collection and Study of dataset	Completed
Data Cleaning and Pre-processing	Completed
Studying the various algorithms through which the project can be implemented	To be finalised by 15th March, 2019
Model Implementation	16th March, 2019 – 20th March, 2019
Model Testing	21st March, 2019 – 24th March, 2019
Creating Visualizations	25th March, 2019 – 28th March, 2019
Report and Presentation	To be completed by 2nd April, 2019

References

[1]"Toxic Comment Classification Challenge | Kaggle", *Kaggle.com*, 2019. [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data. [Accessed: 12- Mar- 2019]

[2]"Text Preprocessing in Python: Steps, Tools, and Examples", *Medium*, 2019. [Online]. Available: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908. [Accessed: 12- Mar- 2019]

[3]"API Reference — tweepy 3.5.0 documentation", *Docs.tweepy.org*, 2019. [Online]. Available: http://docs.tweepy.org/en/v3.5.0/api.html. [Accessed: 12- Mar- 2019]