TOXICITY DETECTION USING NLP

Punarva Vyas
Faculty of Computer Science
Dalhousie University, Canada
pn605302@dal.ca

Sahil Fruitwala
Faculty of Computer Science
Dalhousie University, Canada
sahil.fruitwala@dal.ca

Abhijeet Singh
Faculty of Computer Science
Dalhousie University, Canada
ab249429@dal.ca

Joel Joseph Thomas
Faculty of Computer Science
Dalhousie University, Canada
jl682450@dal.com

PROBLEM STATEMENT:

The verbal abuse, disagreement, and harassment in social media platforms have prompted people to stop expressing their thoughts and opinions. As a result, most of the social media platform enterprises are challenged to facilitate communications effectively and also being provoked to limit their service or also shut down user comments. To successfully overcome this crisis, an effective tool/service is required which would identify the toxic comments with the help NLP techniques over various machine learning models. This model helps in detecting toxicities in the comments entered by various users [1].

LITERATURE REVIEW & RELATED WORK:

Toxicity detection of text in NLP is complex due to the absence of inflection and emotions. The modern developments in NLP research have seen increasing curiosities in evaluating the negative and positive sentiment of words or phrases. Toxicity detection of text in NLP is complex due to the absence of inflection and emotions. The modern developments in NLP research have seen increasing curiosities in evaluating the negative and positive sentiment of words or phrases. Nonetheless, efficiency and exactness of outcomes are usually influenced by false sensibilities of sarcasm and this flaw is often left unnoticed and not rectified.

In sentimental Analysis, Toxicity Detection is one of the most challenging problems across various languages. For example, a project on sarcasm detection in the text was presented by Chun-Che Peng, Mohammad Lakis & Jan Wei Pan which includes various machine learning classifiers such as Naive Bayes & Kernel Support Vector Machine to comprehend and distinguish sarcastic comments from user inputed texts [2].

However, by using the Naive Bayes classifier with TF-IDF features the results obtained were inefficient. After a thorough analysis, the Naive Bayes approach did not consider contextual clues, feature types such as sentiment contrast, the order of words, etc. Both the models Naive Bayes and Kernel Support Vector Machine misclassify most of the sarcastic data. The role of the features is very essential for the system in order to perform successfully and enhance the quality of sarcasm detection [2].

Similarly, the project on Hate Detection on World Wide Web has also implemented various features and ML models such as POS and SVM respectively to predict whether the text is a hate speech or not. The prediction results of this project yielded a good accuracy score. However, the precision and recall scores were comparatively low. We will be studying and implementing the various features and models in our project to get the desired result and predictions [3].

METHODOLOGY:

DATASET AND FEATURES:

In this project, we have procured the dataset from Kaggle with the help of an API. After the dataset has been gathered, we decided to perform data processing i.e transforming the textual data into computer understandable language. Data cleaning is one of the main functions of data pre-processing to filter out useless data in order to save computational power and time. We have planned to clean the data by removing the emojis, http URLs and a few more special characters by just containing only alphanumeric values and whitespaces. Once we have completed cleaning the dataset, we have created a list of words which are ambiguous in a text document and to be replaced in their actual form instead of their contractions such as “ain’t: am not”, “didn’t: did not” & “can't: can not”. Later, we will be implying various other data processing techniques such as Lemmatization, Stopwords, etc and also create features using Bag of Words, TF-IDF, spaCy respectively with the help of standard python libraries.

Under lemmatization, we have converted the words from the document into its dictionary base form. It is suitable for the text classification problem in our project. Likewise, stopwords removal has been implemented in our project to eliminate the most common words which don't add any value to the meaning of the document while performing analysis.

After data pre-processing, we have created features using Bag of Words which is used i to get the term frequency of each word as a feature, by keeping its multiplicity and ignoring the rest i.e. it creates a dictionary of all the unique words occurring in all the documents.

Likewise, we are also creating features by applying a statistical approach TF-IDF, designed to indicate how essential a word is to a document in a corpus or a collection by creating features using the TF-IDF as a weighting factor.
Atlast, we will be creating the last set of features using spaCy which is a python software library which is used to process large volumes of text. Since spaCy uses the best and latest algorithm, it’s expected and considered to perform better with word tokenization and POS-tagging when compared with other python libraries

PROJECT PLAN & FUTURE WORK:

After creating the features from our dataset, we will be feeding all these features into various machine learning models to determine and predict the toxicity of the comment with accuracy, precision, and recall scores respectively for each model. In our project, we have concluded to use algorithms such as Support Vector Machine(SVM), Naive Bayes, Random Forest, k-nearest neighbors and a few more. We will be comparing the results from various machine learning models and select the best among them which has performed. Once the final working model has been confirmed, we will be creating the prototype of the product/service which has user front-end UI which enables the user to comment on a text and submit it. Once the text is submitted, we will be making API calls to perform backend operations. The UI will retrieve the results from the back-end and display the result to the user whether the entered text is toxic or not.

REFERENCES

[1] Kaggle. Jigsaw Unintended Bias in Toxicity Classification. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

[2] Chun-Che Peng, Mohammad Lakis, Jan Wei Pan. Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem. http://cs229.stanford.edu/proj2015/044_report.pdf

[3] William Warner. Detecting Hate on the World Wide Web. https://www.researchgate.net/publication/262363934_Detecting_hate_speech_on_the_world_wide_web