



Data Science Intern at Data Glacier

Project: Hate Speech Detection using Transformers (Deep Learning)

Week 13: Final Report

Name: Harshad S. Chormare

University: University of Mumbai

Email: chormareharshad@gmail.com

Country: India

Specialization: Data Science

Batch Code: LISUM19

Date: 30 May 2023

Submitted to: Data Glacier

Table of Contents:

1. Project Plan	3
2. Problem Statement.....	3
3. Data Collection	3
4. Data Preprocessing	4
4.1. Text Cleaning.....	4
4.1.1. Lower Case.....	4
4.1.2. Remove Punctuation.....	4
4.1.3. Remove URL.....	4
4.1.4. Remove @tags.....	4
4.1.5. Remove Special Characters.....	4
4.2. Preprocessing Operations	4
4.2.1. Tokenization.....	4
4.2.2. Remove Stop Words.....	5
4.2.3. Lemmatization.....	5
4.2.4. WorldCloud	5
4.3. Feature Extraction.....	6
4.3.1. TF-IDF Model	6
4.4. Split the Dataset into Train and Test	6
5. Build the Model	6
5.1. CNN with LSTM	6
6. Result Evaluation	7
6.1. Evaluation Criteria.....	7
7. Application Design	8
7.1. Turning ML Model Into Flask Framework	9
7.2. Running Procedure	10
8. Conclusion	11
Reference.....	12

1. Project Plan

Weeks	Date	Plan
Weeks 07	April 19, 2023	Problem Statement, Data Collection, Data Report
Weeks 08	April 26, 2023	Data Preprocessing (Text Cleaning)
Weeks 09	May 2, 2023	Data Preprocessing (Preprocessing Operation + Feature Extraction)
Weeks 10	May 9,2023	Building the Model
Weeks 11	May 16,2023	Model Result Evaluation
Weeks 12	May 23,2023	Build ML APP Based on Flask
Weeks 13	May 30,2023	Final Submission (Report + Code + Presentation + Application)

2. Problem Statement

The term hate speech is understood as any type of verbal, written or behavioural communication that attacks or uses derogatory or discriminatory language against a person or group based on what they are, in other words, based on their religion, ethnicity, nationality, race, color, ancestry, sex or another identity factor. In this problem, we will take you through a hate speech detection model with Machine Learning and Python.

Hate Speech Detection is a task of sentiment classification. So, for training, a model that can classify hate speech from a certain piece of text can be achieved by training it on a data that is used to classify sentiments. So, for the task of hate speech detection model, we will use the Twitter tweets to identify tweets containing Hate speech.

3. Data Collection

The Data is about Twitter hate Speech taken from Kaggle which contains the 3 number of features and 31962 number of observations. Dataset using Twitter data, it was used to research hate-speech detection. The text is classified as: hate-speech, offensive language, and neither. Due to the nature of the study, it is important to note that this dataset contains text that can be considered racist, sexist, homophobic, or offensive.

Table 1: Data Information

Total number of observations	31962
Total number of files	1
Total number of features	3
Base format of the file	csv
Size of the data	2.95 MB

4. Data Preprocessing

In part, we explain the data preprocessing approach that we apply in the text data.

4.1 Text Cleaning

First, we clean our text because it was so messy data.

4.1.1 Lowercase

Converting a word to lower case (NLP -> nlp). Words like Racism and racism mean the same but when not converted to the lower case those two are represented as two different words in the vector space model (resulting in more dimensions). Therefore, we convert all text word into lower case letter.

4.1.2 Remove Punctuation

It is important to remove the Punctuation because is not important. Therefore, we remove that Punctuation to do that we use regular expression.

4.1.3 Remove URLs

In this part, we remove URLs because we are working on hate speech application which detect the hate and free speech and to get the output, we need to give only text not URLs therefore, we remove the URLs because we need only clean text input.

4.1.4 Remove @tags

In this part, we remove @tags which used when we mentioned someone So, it's doesn't concern to our application therefore, we remove @tags by using regular expressions.

4.1.5 Remove Special Characters

Remove Special Characters is essentially the following set of symbols [!"#\$%&'()*+,-./:;<=>?@[^_`{|}~] which basically don't have meaning. Therefore, we remove that kind of symbols because we don't need that. To remove we use python isalnum method.

4.2 Preprocessing Operation

In this part, we implement the preprocessing operation

4.2.1 Tokenization

Tokenization is breaking the raw text into small chunks. Our text data is into paragraph so to convert into work tokenize we use nltk word_tokenize library. These tokens help in