Data Science Intern at Data Glacier

Project: Hate Speech Detection using Transformers (Deep Learning)

Week 7: Deliverables

Name: OBIDA ALHAMOUD

University: MANISA CELAL BAYAR UNIVERSITY

Email: obaida.ismail.alhamoud@gmail.com

Country: Turkey

Specialization: Data Science

Batch Code: LISUM35

Date: 17 Aug 2024

Submitted to: Data Glacier

1.Project Lifecycle & Deadlines

Weeks	Date	plan

Weeks 07	May 18, 2022	Problem Understanding Research hate speech detection techniques. Analyze the problem and define the scope.
Weeks 08	May 25, 2022	Data Cleaning and Normalization. Preprocess the tweets by removing noise (e.g., URLs, special characters). Handle missing data, if any. Normalize the text data.
Weeks 09	June 1, 2022	Representation Learning
Weeks 10	June 8, 2022	Model Building & Training
Weeks 11	June 14, 2022	Performance Evaluation & Reporting
Weeks 12	June 21, 2022	Model Deployment & Inference
Weeks 13	June 30, 2022	Documentation & Submission

2. Problem Description

- **Objective:**
  - Develop a model to detect hate speech in tweets using deep learning techniques, specifically Transformers.
- **Definition of Hate Speech:**
  - Any communication that attacks or uses derogatory or discriminatory language against a person or group based on religion, ethnicity, nationality, race, color, ancestry, sex, or other identity factors.
- **Data Source:**
  - A labeled dataset of tweets where `label` is 0 or 1 (0 for non-hate speech, 1 for hate speech), and `text_format` contains the original tweets.

3. Business Understanding

- **Importance of Hate Speech Detection:**
  - Ensures safer online communities by identifying and mitigating hate speech.

  - Supports social media platforms in enforcing policies against harmful content.
- **Potential Applications:**
  - Content moderation on social media platforms. ○ Automated reporting of harmful content. ○ Enhancing user experience by filtering out hate speech.

4. What type of data do we have:

A dataset contains 3 features:

1. Id
2. Label
3. Text

Id feature datatype is integer and it contains the tweet Id.

Label is an integer of 0 and 1 and it represents if the text is negative or positive.

Text is a string feature ant it contains the text of tweet.

5. Approaches to clean the data:

Using libraries like pandas and re could help us to clean and normalize the dataset

6. Problems:

we need to remove special characters and remove all the unnecessary things like:

1. Punctuation
2. URLs
3. @tags

Punctuation: it is important to remove the punctuation because is not important.
We remove that using regular expressions.
URLs: because we are working on hate speech detection app, we need to give only the text.
@tags: we remove @tags using regular expressions

EDA

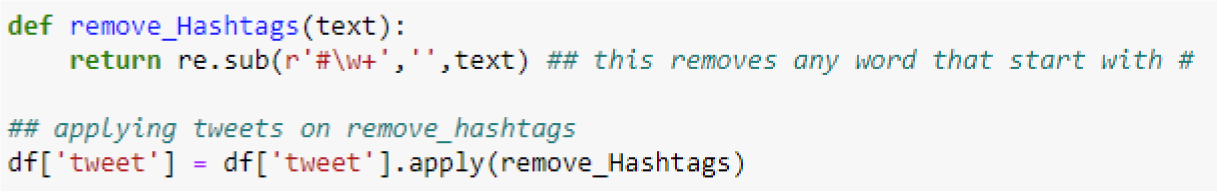
1.Data cleaning

To prepare out dataset I performed some operations on dataset and these operations are:

- Removing tags
- Removing Hashtags
- Removing links
- Convert to lowercase
- Remove emojis and symbols

To remove tags and Hashtags we use regex library.

It’s a library that we can perform some operations on text

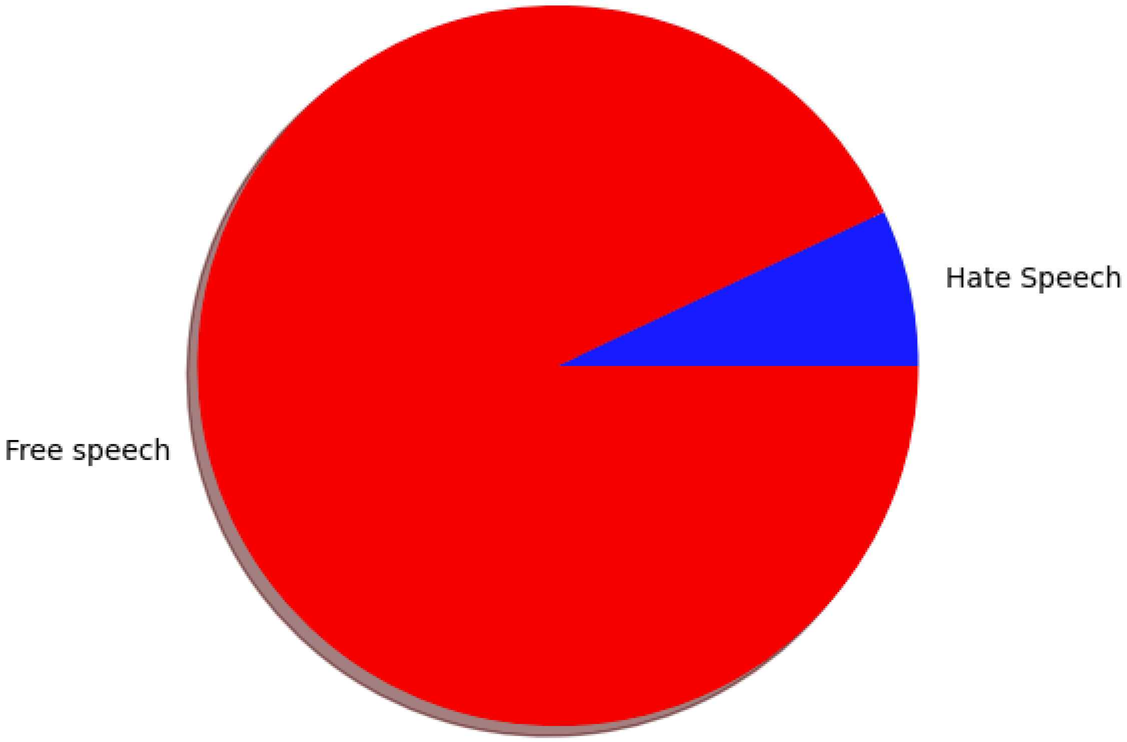


```
def remove_Hashtags(text):
    return re.sub(r'#\w+','',text) ## this removes any word that start with #

## applying tweets on remove_hashtags
df['tweet'] = df['tweet'].apply(remove_Hashtags)
```

2.Data Visualizing

To have a nice look on the dataset and see the relationship between the features to plot the features on screen, so we can use matplotlib library.

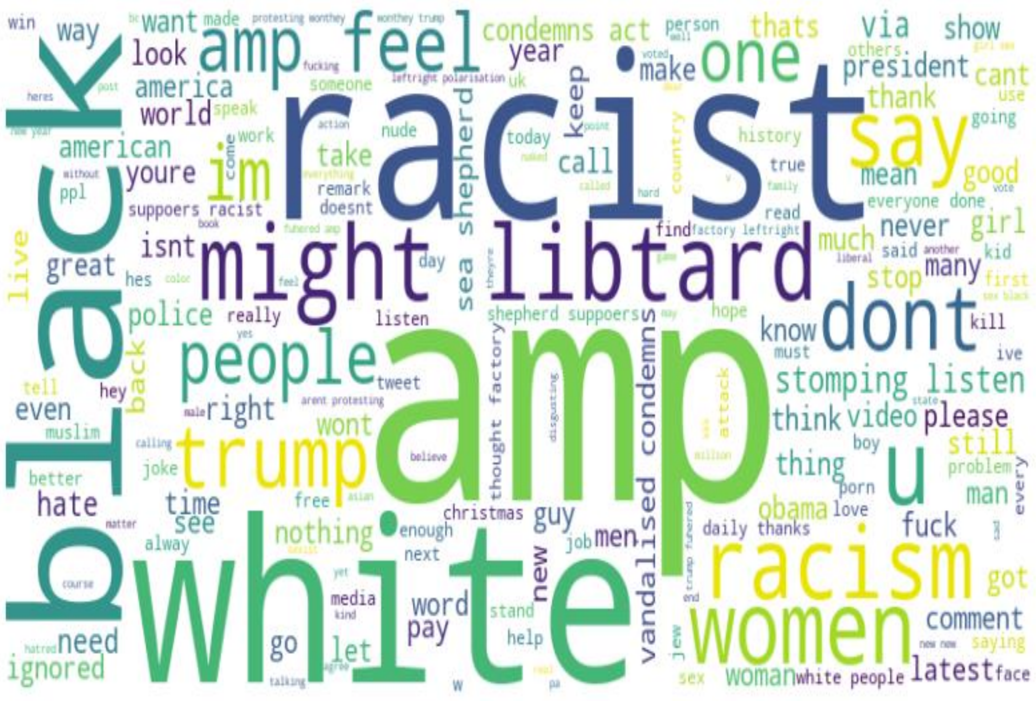


As we can see the text that has been labelled as ‘Hate Speech’ is a small percentage of our dataset

Word cloud:

A **word cloud** (or **tag cloud**) is a visual representation of text data where the size of each word indicates its frequency or importance in the text. Words that appear more frequently are displayed in larger, bolder fonts, while less frequent words appear smaller. This makes word clouds a popular tool for quickly identifying the most prominent terms in a body of text, such as a document, tweet, or collection of comments.

These are the most frequently displayed negative words in the dataset



3.Feature Extraction

Here we must vectorize our text to make feature so that we can measure how strong the word is and where is it frequently used.

TF IDF Vectorizer is a good choise to do that.