Data Glacier

Your Deep Learning Partner

Twitter Hate Speech Detection using Transformers:
Exploratory Data Analysis

Group name: VerbalVigilantes
Name: Zohra Bouchamaoui
Email: Zohra.Bouchamaoui@outlook.com

Country: United Kingdom
Company: Data Glacier

Batch Code: LISUM21
Specialisation: NLP

Agenda

Problem Description

Data Description

Data Cleaning and Transformation

Exploratory Data Analysis (EDA)

Key Insights and Findings

Model Building and Evaluation

Model Hyperparameter Tuning

Conclusion

Recommendations (1 & 2)

Problem Description

Introduction:

- Hate speech is defined as any form of derogatory or discriminatory communication targeting individuals or groups base on attributes.
- Objectives: Design a machine learning model to accurately detect hate speech.

Approach:

- Hate speech detection as a form of sentiment classification.
- Utilising a commonly used sentiment classification dataset for training.
- Twitter data is used for hate speech detection in this project.

Data Description

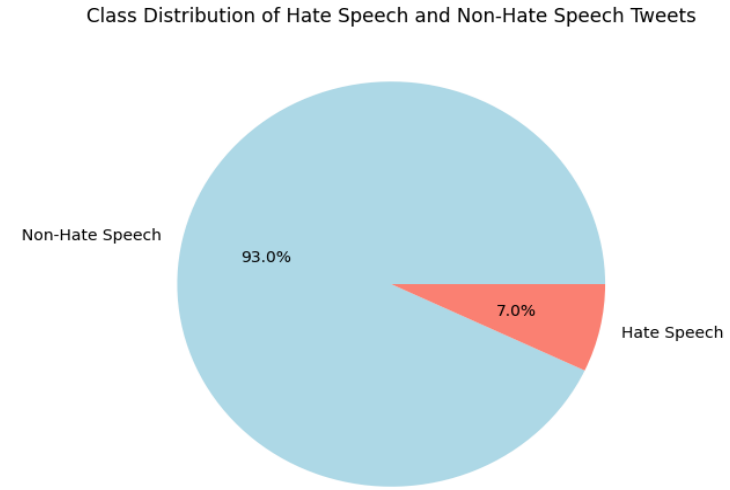
- Twitter dataset collected by Rahul Agarwal on Kaggle[1] , containing labeled tweets for hate speech detection.
- Datasets consists of two files train and test.
- Train dataset consists of tweets with labels indicating hate speech or non-hate speech (using 0 as non-hate speech and 1 as hate speech).
- Both train and test sets have unique identifiers (id) for each tweet and have a tweet column containing the text content of the tweets.

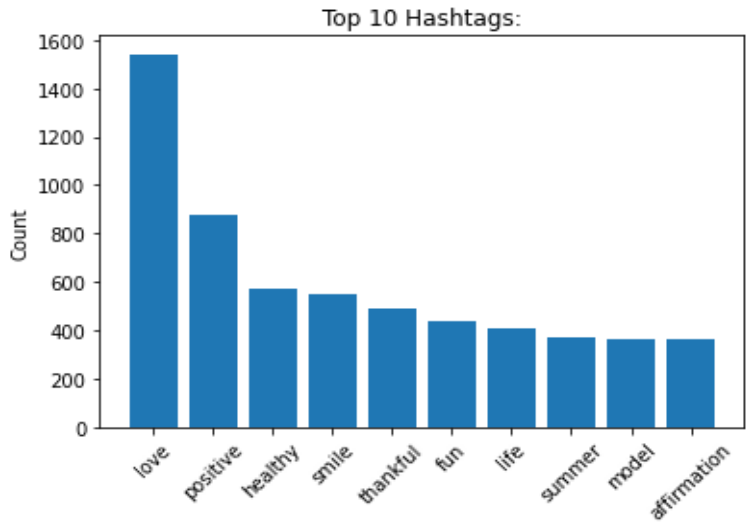
1. https://www.kaggle.com/datasets/vkrahul/twitter-hate-speech?select=train_E6oV3lV.csv

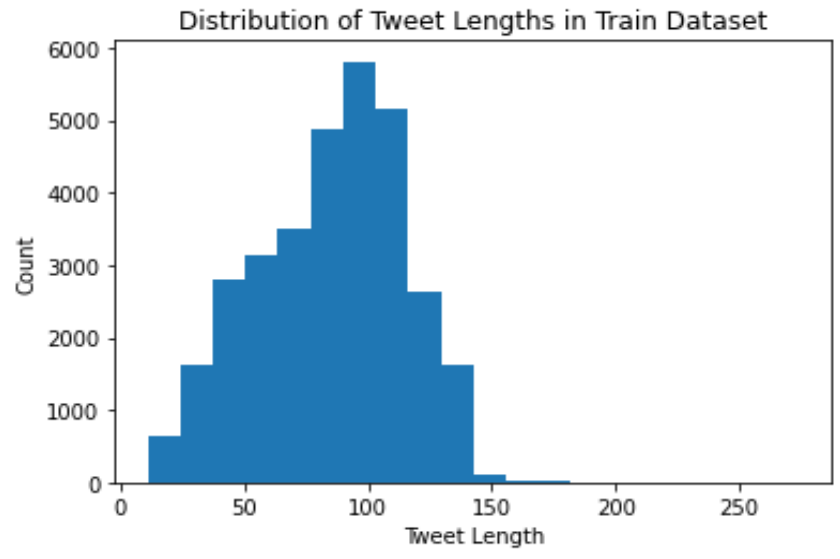
Data Cleaning and Transformation

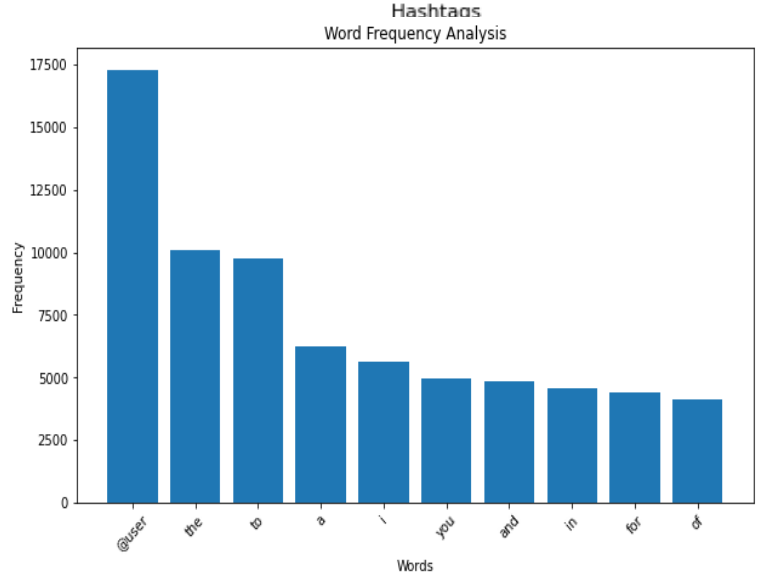
- Data preprocessing techniques applied to improve data quality
- Highlights of the cleaning process:
    - Removal of URLs and special characters
    - Lowercasing the text
    - Handling missing values, if any

Exploratory Data Analysis (EDA)









Data Glacier

Key Insights and Findings

- Dataset exhibits class imbalance with a larger number of non-hate speech tweets.
- Shorter tweets tend to have more negative sentiment.
- Common words include, *'@user'* (freq. 17,291), *'the'* (10,065), *'to'* (9,768), *'a'* (6,261), *'i'* (5,655).
- Top hashtags include, *'love'*, *'positive'*, *'healthy'*, *'smile'*, *'thankful'*, *'fun'*, *'life'*, *'summer'*, *'model'*, *'affirmation'*.
- Topic modelling showed 5 major topics within the tweets:
  - **Topic 1:** day, happy, new, love, father
  - **Topic 2:** love, life, time, bull, smile
  - **Topic 3:** like, amp, need, people, don
  - **Topic 4:** user, thankful, positive, amp, just
  - **Topic 5:** user, good, amp, gt, music

Model Building and Evaluation

- Hyperparameter tuning was described as a process to optimize model performance.
- We used techniques like GridSearchCV to find the best hyperparameters and improve hate speech detection performance.
- Results after hyperparameter tuning:

SVM	
Accuracy	0.9576
Precision	0.8685
Recall	0.4781
F1-Score	0.6167

RNN	
Accuracy	0.9521
Precision	0.6704
Recall	0.6469
F1-Score	0.6584

XGBoost	
Accuracy	0.9510
Precision	0.7375
Recall	0.4868
F1-Score	0.5865

Model Hyperparameter Tuning

- We used three machine learning models for hate speech detection: Support Vector Machine (SVM), Recurrent Neural Network with Word Embeddings, and XGBoost with Bag-of-Words.
- Results before hyperparameter tuning:

SVM	
Accuracy	0.9557
Precision	0.8035
Recall	0.5022
F1-Score	0.6181

RNN	
Accuracy	0.9587
Precision	0.7874
Recall	0.5767
F1-Score	0.6658

XGBoost	
Accuracy	0.9513
Precision	0.7457
Recall	0.4824
F1-Score	0. 5858

Conclusion

- Experimented with three models: SVM with TF-IDF, RNN with word embeddings, and XGBoost with Bag-of-Words for hate speech classification.
- Before hyperparameter tuning, SVM achieved 95.8% accuracy, RNN 95.2%, and XGBoost 95.1% accuracy.
- After tuning, SVM improved to 95.6% accuracy, RNN to 95.9%, and XGBoost showed minimal improvement to 95.1% accuracy.
- SVM and RNN exhibited balanced precision and recall for class 1, with F1-score of 0.67.
- RNN outperformed SVM in accuracy and F1-score after tuning due to its ability to capture sequential patterns and dependencies in text.
- RNN with word embeddings showed the best performance in hate speech classification after hyperparameter tuning.

Recommendations (1)

- Developing strategies to monitor and moderate specific hashtags associated with hate speech.
- Implementing user reporting mechanisms to identify and address hate speech instances.
- We recommend further exploring advanced NLP techniques to enhance hate speech detection accuracy, e.g., lemmazation, stemming, and part-of-speech tagging.
- Collecting and utilizing more extensive datasets can potentially improve the model's generalization.
- The hate speech detection model can be applied to other platforms or domains to address online toxicity effectively.

Recommendations (2)

- Implement advanced word embeddings models such as Word2Vec, GloVe, or BERT to capture contextual information and semantic meaning of words in hate speech context.
- Consider employing transformer-based models like GPT-3 or RoBERTa, which have demonstrated remarkable performance in various NLP tasks.
- Explore transfer learning techniques to leverage pre-trained language models on large text corpora and fine-tune them on the hate speech detection task.
- Combine predictions from multiple hate speech detection models, leveraging the strengths of different algorithms, to create an ensemble model.

Thank You