



Using Sentiment Analysis to Detect Hate Speech

By: Varun Chilukuri, Rahul Nair, Anirudh Poruri, Shreya Shete

Problem

- Millions of people use social media platforms like Youtube and Twitter every day
- Such platforms oftentimes have hateful and offensive language, which can influence impressionable youth.



Problem (cont.)

Project Goals:

- To identify hateful/offensive comments so they can be removed on these platforms
- Develop a user-friendly tool that utilizes our model's results to classify comments

What makes this challenging?

- Hate speech can be subtle and heavily context-dependent
- Language is constantly evolving and new phrases are being coined all the time
- We need to balance sensitivity and specificity—overly aggressive models may result in high false positives, impacting free speech and causing user dissatisfaction.


Word Cloud

“Neither Hate Speech nor Offensive Language”:



Dataset

Hate Speech and Offensive Language Dataset from Kaggle:



+

Create

Home

Competitions

Datasets

Models

< > Code

Discussions

Learn

More

Search

ANDRII SAMOSHYN · UPDATED 4 YEARS AGO

142

New Notebook

Download (1 MB)

Hate Speech and Offensive Language Dataset

research hate-speech detection

Data Card

Code (36)

Discussion (4)

Suggestions (0)

About Dataset

Dataset using Twitter data, is 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's important to note that this dataset contains text that can be considered racist, sexist, homophobic, or generally offensive.

Usability

9.41

License


[CC0: Public Domain](#)

Expected update frequency

Never

Tags

Text

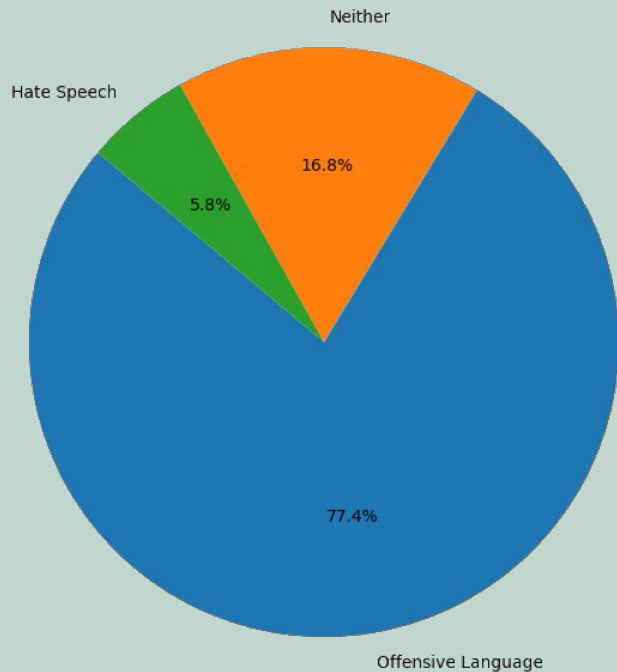


Approach / Methods

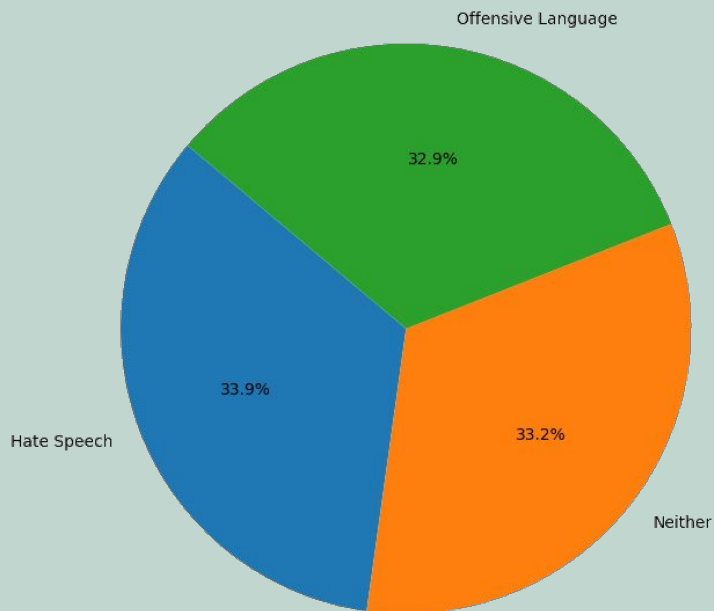
- Libraries: Tensorflow, Keras, Numpy, Pandas, Nltk, Sklearn, matplotlib
- Data preprocessing (for Tweet column)
 - balancing dataset by upsampling hate speech & downsampling offensive speech
 - removing punctuation & stop words
 - splitting text into words
 - tokenizing, converting to integers
 - padding sequences to have same length
- Bi-Directional LSTM Model
 - Input: 1D Sequence of Padded Tokens
 - Output: Classifies as Hate Speech and/or Offensive Language, or Neither
- Training & Testing Model
- Plotting Accuracy and Loss for Training & Validation Data over Epochs
- Calculating Accuracy Precision, Recall, and F1 Score for Test Data

Class Distribution After Upsampling and Downsampling

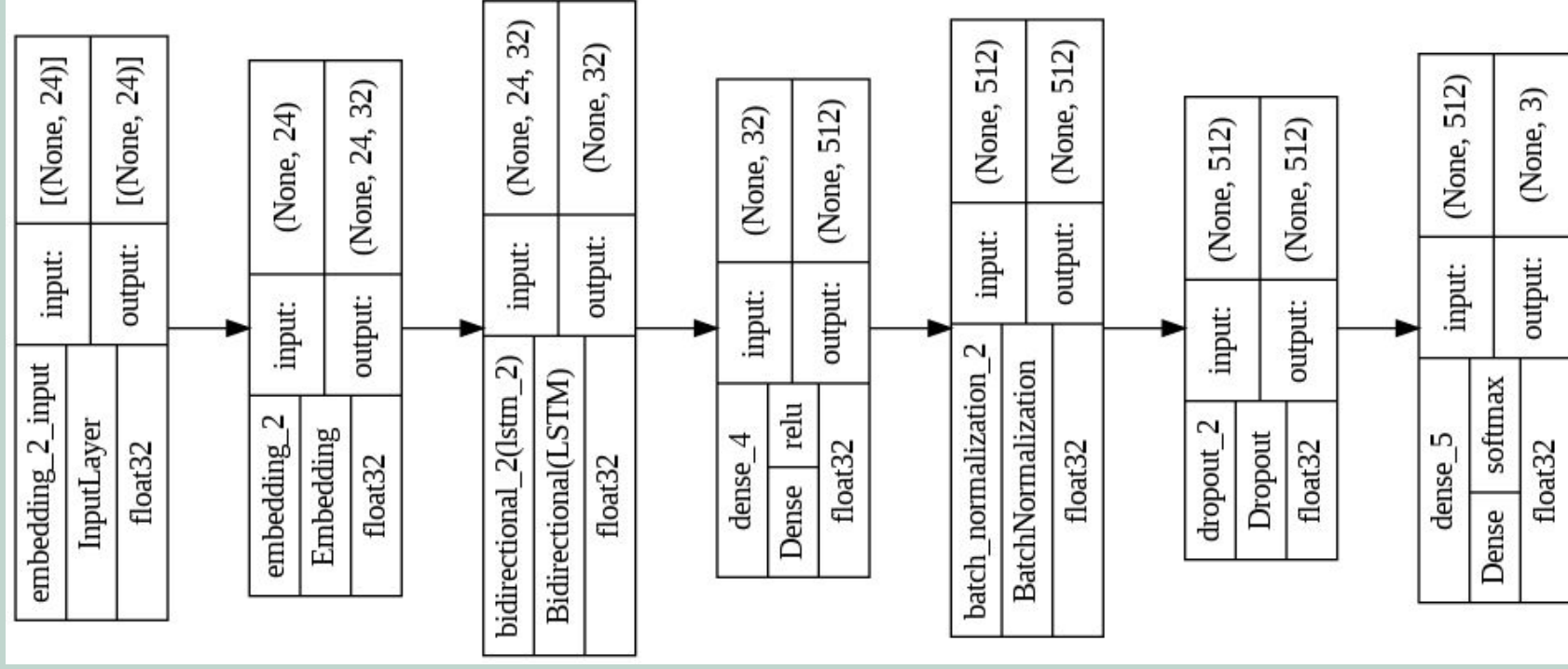
Distribution of Tweet Classes



Distribution of Tweet Classes



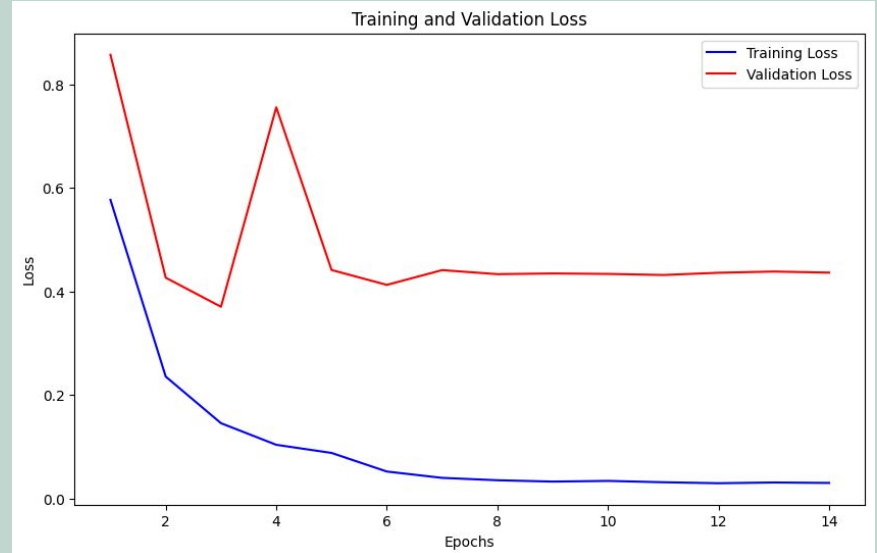
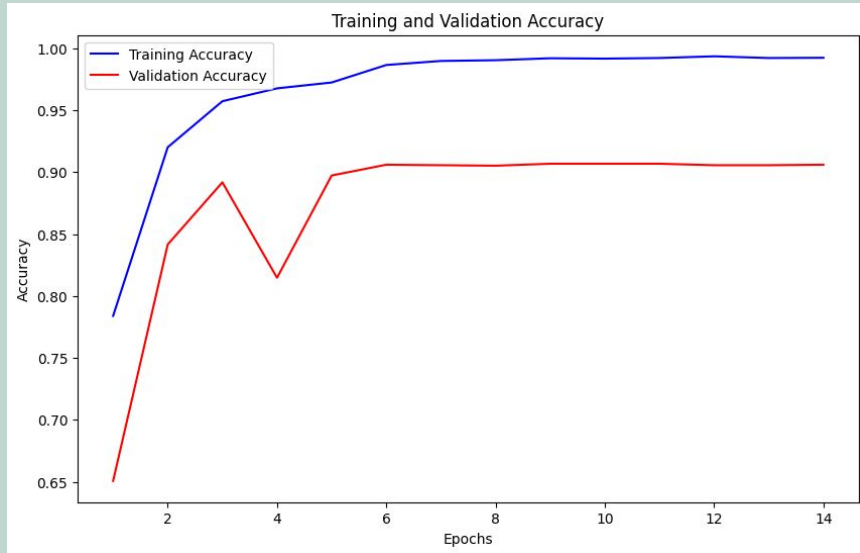
Model Architecture



Results

Accuracy on test data: 0.9067562222480774
F1 Score: 0.9007254838943481
Precision: 0.9020119905471802
Recall: 0.8994792103767395

- We achieved a validation accuracy of nearly 91%
- Precision, Recall, F1 score of nearly 90% → algorithm returns more relevant results

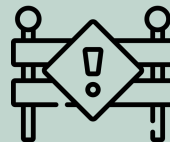


Discussion



Success Criteria

- The model accurately identifies hate speech, offensive language, and neutral speech
- We achieved a high accuracy and built two user-friendly tools that allow us to analyze individual comments or an entire YouTube video section



Roadblocks

- Initial dataset improperly labeled hate speech values as “neutral”
- New dataset had more nuance—it differentiated between “offensive language” and “hate speech”

Applications

Text Classifier

you are a smelly idiot



Classify Text

Entered Text: **you are a smelly idiot**

Offensive Percentage: **49.67**

Predicted Category: **Offensive Language**

Hate Speech Detection Results

Video Name: **Offering People \$100,000 To Quit Their Job**



Looking through top 100 comments

Comment Classifications:



Breakdown:

Not Offensive: 80.0%
Leaning Towards Offensive: 12.0%
Offensive: 7.000000000000001%
Very Offensive: 0.0%
Hate Speech: 1.0%

Our Solution

Limitations

- Profanity and true hate speech is classified well
- Friendly and supportive text is classified well
- Strong real-world applications

Future Work

- Gather more data or improve on size of data with more augmentation
- Improve Accuracy of model in general
- Classify mildly offensive text more accurately
- Provide a more detailed text classification rather than just “offensive language”, “hate speech”, or “neither”