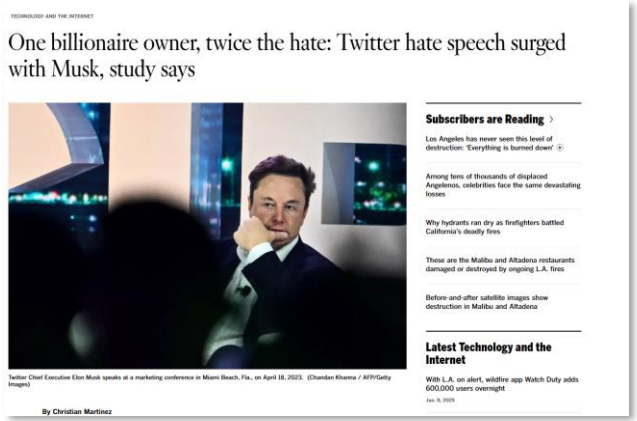
Offensive Speech Recognition

23rd January, 2023

“Moderation is a propaganda word for censorship”

Elon Musk, interview with Don Lemon, March 2024

Big tech companies are reducing their focus on moderation leading to increase of hate speech


TECHNOLOGY AND THE INTERNET

One billionaire owner, twice the hate: Twitter hate speech surged with Musk, study says

Twitter Chief Executive Elon Musk speaks at a marketing conference in Miami Beach, Fla., on April 18, 2023. (Chandan Khanna / AFP/Getty Images)

By Christian Martinez

Subscribers are Reading

Los Angeles has never seen this level of destruction: 'Everything is burned down'

Among tens of thousands of displaced Angelenos, celebrities face the same devastating losses

Why hydrants ran dry as firefighters battled California's deadly fires

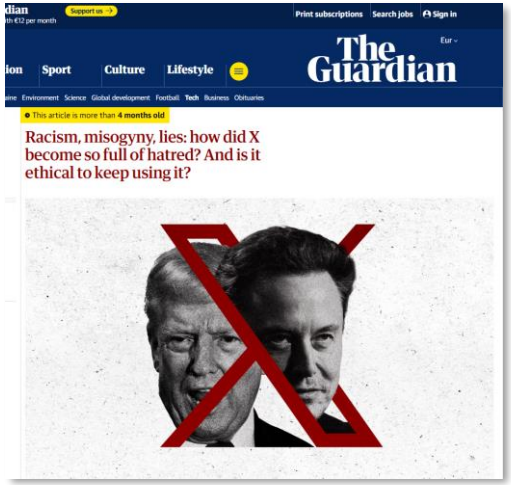
These are the Malibu and Altadena restaurants damaged or destroyed by ongoing L.A. fires

Before-and-after satellite images show destruction in Malibu and Altadena

Latest Technology and the Internet

With L.A. on alert, wildfire app Watch Duty adds 600,000 users overnight

Jan. 8, 2025


Support us

Print subscriptions | Search jobs | Sign in

The Guardian

Sport | Culture | Lifestyle

This article is more than 4 months old

Racism, misogyny, lies: how did X become so full of hatred? And is it ethical to keep using it?


Meta's new hate speech guidelines permit users to say LGBTQ people are mentally ill

Changes to its hate speech guidelines were among broader policy shifts Meta made to its moderation practices.

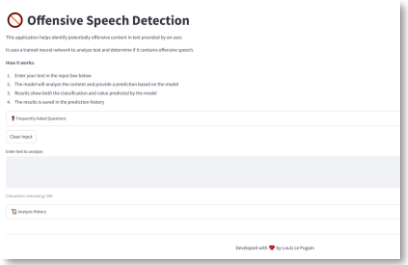
Objectives : build a fully deployed offensive speech detection model

Target

Build a model to detect offensive speech

Results

Deployed model with 73% accuracy and 68% recall



4 steps to get a deployed model

	Description	Technology
1 Dataset	• Choose dataset to use in the training	Hugging Face
2 Model	• Build preliminary models to define the best design	colab
3 Fine-Tuning	• Fine-tune the model to find best parameters • Register prediction and preprocessing models	mlflow
4 Deployment	• Make the model available through an API • Create an app with Streamlit to use it	FastAPI Streamlit

OLID dataset used for training, with 13.2k tweets, of which 33.2% are offensive

Source

- Catalog of abusive language data (PLoS 2020)
- Paper : *Vidgen B, Derczynski L (2020) Directions in abusive language training data, a systematic review: Garbage in, garbage out*
- Catalogue of datasets in different languages
- 25 languages available
- 59 datasets in English

Dataset used

- Name : The Offensive Language Identification Dataset (OLID)
- Paper : *Predicting the Type and Target of Offensive Posts in Social Media (2019)*
- Available data : Annotated tweets flagged offensive or not offensive
- Size : 13,240 tweets, of which 4,400 flagged as offensive (33.2% of total)
- Available on Hugging Face

tweet	subtask_a
@USER She should ask a few native Americans wh...	OFF
@USER @USER Go home you’re drunk!!! @USER #MAG...	OFF
Amazon is investigating Chinese employees who ...	NOT
@USER Someone should'veTaken" this piece of sh...	OFF
@USER @USER Obama wanted liberals & illega...	NOT

A neural network with GRU layers were chosen for deployment

	Description	Preprocessing	Results
GRU	• Gated recurrent units neural networks • 1 Embedding layer • 1 GRU layers with 64 units • Max Pooling / Dropout / Dense final layers	• Text cleaning : Punctuation and symbols removal, conversion to lowercase • Lemmatization and Stop Words removal • Encoding and Padding	• F1 Score : 0.63 • Recall : 0.66 • Accuracy : 0.74 Chosen model
LSTM	• Long short-term memory neural network • 1 Embedding layer • 2 LSTM layers with 64 units • Max Pooling / Dropout / Dense final layers	• Text cleaning : Punctuation and symbols removal, conversion to lowercase • Lemmatization and Stop Words removal • Encoding and Padding	• F1 Score : 0.61 • Recall : 0.58 • Accuracy : 0.75
BERT	• Use of BERT pre-trained model from 2018 • Use of preprocess and encoder BERT layers • Dropout and Dense final layers	• Text cleaning : Punctuation and symbols removal, conversion to lowercase • Lemmatization and Stop Words removal • BERT pre-trained preprocessed layer using directly text	• F1 Score : 0.59 • Recall : 0.57 • Accuracy : 0.73 • 3h training time

The model registered in MLFlow reached 68% recall and 73% accuracy

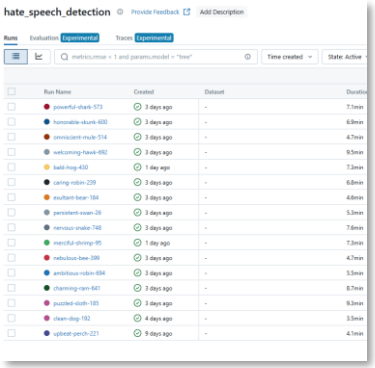
MLFlow Experiments

mlflow

Overview

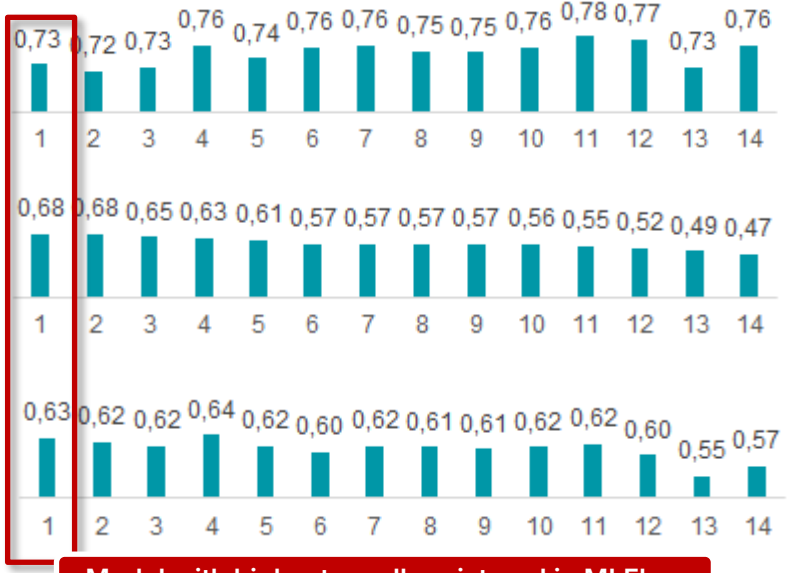
- Change the number of units of each layers
- Add intermediate layers
- Increase or decrease the number or epochs for training

Experiments



Results

Accuracy



Recall

F1 Score

Model with highest recall registered in MLFlow

Additional preprocessing model was registered on MLFlow

mlflow

Raw Tweet

- Model input
- Example : *"@USER @USER Go home you're drunk!!! @USER #MAGA #Trump2020 🇺🇸🇺🇸 URL"*

Preprocessing

- Punctuation, and symbols removal
- Conversion to lowercase
- Lemmatization
- Stop Words Removal

Vectorization

- Text conversion to integer
- Padding of each sentence

Prediction Layers

- Embedding layers
- Prediction based on GRU layers

Scope of prediction model

Not in the previous model. Additional preprocessing model registered on MLFlow

The model were deployed through a Streamlit app use an API build with Fast API

API

FastAPI

Description

- FastAPI API deployed on Hugging Face
- 2 endpoints using MLFlow models :
  - /preprocessing
  - /predict using the preprocessing model
- Prediction output
  - Prediction : “Offensive” or “Not Offensive”
  - Probability : Number between 0 and 1, the higher, the more offensive

Overview

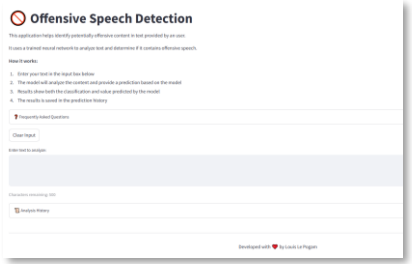
Detection App

Streamlit

Description

- Streamlit app deployed on Hugging Face
- Text entered as an input from the user
- Output from the model shown after validation
- Various explanation and FAQ
- History of research saved on a S3 bucket

Overview



Demonstration

Let's open the app and see how it works!

Next Steps : Further improve the model and the app

	Improvements
Model	• Train the model on bigger dataset to improve the performance • Use remote training to use more ambitious models • Leverage on improvements in text detection and pre-trained model
App Performance	• Improve app performance to insure quicker prediction • Additional costs required to have access to better server
App UI	• Handle better exception and potential errors • Add additional endpoint and features (e.g. batch predict from a csv file)

Thanks!