Detection of hate speech and offensive language

---

1. Overview

In the modern world we have a lot of information and resources for its spreading. It could be regular news sites or social networks. The last ones are very popular in our time and it is easy to manipulate or make offensive speech. It could cause psychological harm, polarisation of society, incitement to violence and hate crimes. The primary goal of this project is an elaborate ML algorithm in order to recognise hateful speech and as a sequence to prevent negative consequences.

2. Motivation

This issue is especially important now, as we are living in a time of war. Society is under constant pressure from ongoing hostilities and daily shelling, and people's mental health is already heavily affected. As a result, the public has become more vulnerable to manipulation and incitement. That is why it is crucial to filter out hateful speech.

3. Success metrics

- Time to detection hateful speech
- F1, precision, recall, accuracy
- Moderator workload reduction

4. Requirements & Constraints

Functional requirements:

- Recognition of hate speech followed by tagging or filtering
- The system stores the history of analyzed texts for trend analysis

Non-functional requirements:

- Time detection < 200ms
- Accuracy > 90%

Constraints:

- MVP - 1 month
- Cost of MVP - 1000$

4.1 What's in-scope & out-of-scope?

In-scope:

- Baseline model development
- API for model inference
- MLFlow integration
- Model deployment

Out-of-scope:

- Integration with social networks (because it is MVP)
- Analyze images or video

5. Methodology

5.1. Problem statement

This is a supervised classification problem - hateful or non-hateful speech.

5.2. Data

It was taken from a labeled datasets from Kaggle resource. During serving, the system counts to get raw text strings.

5.3. Techniques

It will be used in the Recurrent Neural Network. Features creation - string will be transformed into tensor. Preparation of the data - it will be a batch of functions that removes noise symbols, urls, needless words (is, are, etc).

5.4. Experimentation & Validation

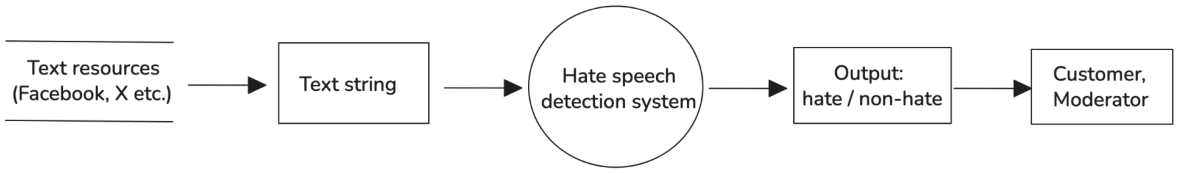
To evaluate the model offline it will be applied to the confusion matrix. Also will be plotted graphs for accuracy; recall, F1-score.

5.5. Human-in-the-loop

Basically it must be an automated system without human invasion. However users could have the possibility to mark the text manually, hateful or non-hateful. So it is needed to elaborate some tool for this.

6. Implementation

6.1. High-level design



6.2. Infra

System will be hosted on cloud, most likely Azure.

6.3. Performance (Throughput, Latency)

The system achieves performance goals through an efficient architecture, use of a lightweight model for inference, and support for horizontal scaling.

6.4. Security

Users will be authenticated by API key. It will be publicly accessible behind a firewall.

6.5. Data privacy

Customer data will not be stored permanently. All data will be retained only temporarily for processing and will be automatically deleted after a defined period.

6.6. Monitoring & Alarms

Events will be logged in the json file. Metrics: latency, timelog, inference result, input data.
It will be provided a script which monitors some threshold and alarm users.

6.7. Cost

It is quite difficult to estimate the cost of the system for that moment because I don't know all the inputs and what resources I need. Firstly I need to try to make the first model and then I can estimate roughly.

6.8. Integration points

Our system will receive upstream data via API from social media platforms or other ingestion services.
The downstream integration will be provided via an API that returns the hate speech classification result, which can be used by moderation dashboards or alerting systems.

6.9. Risks & Uncertainties

- Data drift
- False positives and negatives