PROJECT PROPOSAL

TOXICITY DETECTION IN AUDIO

PURPOSE: This project aims to detect toxic speech in video game communication using audio. The starting point is the TAPAD dataset, which contains cuss word recordings. The goal is to train a machine-learning model to classify new audio sentences as toxic or non-toxic.

An important aspect to think about are the different accents (Phonemes) and the quality of the audios.

Inspiration: Videogames when people are toxic to other players. Working in similar fields will help in censoring hate speech and strict action can be taken against such people.

ACHIEVABILE: The project requires a dataset containing both toxic and clean speech. I found TAPAD dataset on github, but it provides toxic audio samples only, additional clean speech data is needed. These datasets can be found on Kaggle and also on github. I will need to ensure that the datasets are balanced.

The project will be re-implementing an existing system, that flags toxic content. I plan to use different datasets, so there may be a need to pre-process some audio files before the actual model is implemented. I plan on using Random-Forest and CNN for machine learning models.

RELEVANCE: The project is about detecting hate speech in audio. It will be making use of SFTs and Spectrograms to process the audio files.

I have saved some research papers that talk about implementing similar systems.

<https://link.springer.com/chapter/10.1007/978-3-031-64064-3_21>

<https://link.springer.com/article/10.1007/s10772-024-10116-6>

<https://ojs.aaai.org/index.php/ICWSM/article/view/14955/14805>

TIMEBOUND: The steps for the project to be completed are:

1. Collect datasets
2. Process the audio. Trim silent ends, remove noise, etc.
3. Extract useful features
4. Train the machine learning model for Random Forst and CNN
5. Develop the system to take an audio file as input, and classify whether it is normal speech or hate speech -> may need to process the audio file
6. Model evaluation to check whether the model is robust.

MESURABLE: The application can be measured by its accuracy. I hope to attain a minimum of 85% accuracy. I will need to run several grid-params to obtain the best hyper-parameters. The aim is to have lowest number of false values while making sure to not over-fit.