AI-driven Transformation of Hate Speech in Video Content

Clara Chappuis [1] Mehdi Hajoub [1] Renuka Singh Virk [1]

Abstract

In this miniproject, our goal is to detect hate speech in videos and censor it to promote safer online environments. Initially, we transcribe the audio of each video into text. We then employ a fine-tuned pre-trained BERT model to classify the video content as either hate speech or non-hate speech. This model is fine-tuned using the *Dynamically-Generated-Hate-Speech-Dataset* [1].

The primary challenge of this project is to convert harmful sentences into non-harmful ones. To address this, we fine-tune the Meta-Llama-3-8B model to generate non-harmful content from harmful content. Despite the relatively small size of the ChatGPT generated dataset we use for this task, the performance of the LLM is still adequate (see Section 4).

Keywords: hate speech, videos, classification, LLM, lipsync

1. Introduction

Disclaimer: In this report (especially in Sections 3.3.1 and 4) some offensive and hateful material can be found; however it is unavoidable given the nature of this project.

In today's digital landscape, the internet is increasingly accessible, especially to young individuals. This accessibility underscores the importance of prioritizing the creation of safe online environments and promoting inclusivity. In our mini project we aim to contribute to it by tackling hate speech so that it does not reach the younger audience.

The UN Strategy and Plan of Action on Hate Speech defines hate speech as: "*any kind of communication in speech, writing or behaviour, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, colour, descent, gender or other identity factor.*" [2]

The main challenge of our project is that we are able to detect only some types of hate speech, specifically those that manifest as degrading opinions about minority groups or as slurs, insults and stereotypes. Additionally, to test our implementations, we found only a few videos that met our requirements to obtain good results (see Section 3.4):

- Good image resolution
- Good sound quality (not too noisy)
- Protagonist has to be facing camera
- Lips have to be steady and facing camera

2. Related Work

Hate speech detection has gained popularity in many recent researches. We review the project *HateMM: A Multi-Modal Dataset for Hate Video Classification* [3] which is one of the few that directed the detection of hateful content in videos, instead of text media. Their research focuses on various models for hate speech classification, namely BERT, ViT and MFCC.

They create the data set by collecting approximately 43 hours of videos coming from BitChute (a video-hosting platform with low content moderation) that are then manually classified as hate or non-hate; they use all three modalities of the videos (text, audio and video) to classify them. The results obtained show that text-based model performs well when the transcript is clean, the audio-based model when there is shouting or aggression, and the vision-based model can detect hateful content when offensive activities or the target of the abuse are present in the video. [3] In addition, the best performances are obtained with the multi-modal models, in particular with the BERT $\odot$ ViT $\odot$ MFCC which performs the best among all other models (Fig. 1).

We decide to focus our mini-project on text classification,

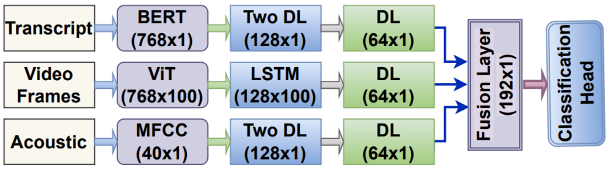

Figure 1. Schematic of the multi-modal model. DL: Dense Layer

[1] Group 19.

EE-559: *Deep Learning*, 2024.

by extracting the text from the videos and using a BERT model to perform the classification.

3. Methods

3.1. Audio transcription

The first step of our pipeline is to extract the text from the videos. We thus start by extracting the audio from the video using the `moviepy` library [4]. From there we extract the text using the `speech_recognition` library [5].

3.2. Classification

After extracting the text from the videos, we need to assess whether it qualifies as hate speech. We use a language representation model called BERT. [6]. As its name suggests, BERT (Bidirectional Encoder Representations from Transformers) is an encoder-based architecture specifically designed for processing sequences of data, in particular words. BERT's strength lies in its use of encoders, which are built on the Transformer architecture. The attention layers in Transformers allow the model to weight the importance of different words in a sequence, enabling BERT to perform well in tasks involving natural language understanding. This model's output is then concatenated with a fully connected layers with a classification layer at the end.

This model is pre-trained and available on the hugging face library. We then finetune it on our hate-speech dataset. This allows to give the model a finer sensitivity to hate speech.

Due to memory constraints, we opt to use only a subset of the dataset, which is still large enough to achieve satisfactory performance.

3.2.1. Dataset description

The dataset we employ to fine-tune the BERT classifier is the *Dynamically-Generated-Hate-Speech-Dataset* [1]. It comprises 41'000 sentences categorized as either hateful or non-hateful. The dataset is balanced, with a slightly higher number of hateful sentences compared to non-hateful ones (see Fig. 2).

The dataset includes instances of hate speech targeting various minority groups based on factors such as race, religion, gender, sexual orientation, and more (see Fig. 3). We use this dataset because of its heterogeneity in the targeted minority groups, as well as its diversity in the language used, i.e. not only slurs.

3.3. Generation of transformed content

Once hate speech sentences have been extracted, we want to generate similar but more inclusive and non harmful content. For example, if one sentence reads ”I hate women”, we would want to transform it into something similar to ”all genders are equal and I have no specific hate for any of them”. Our approach is to use the pre-trained llama3 model [7] and fine-tune it on a dataset containing hate speech and the corresponding politically correct sentence.

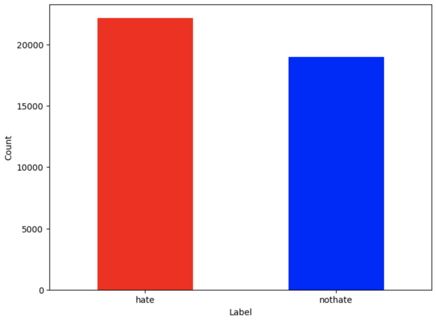

Figure 2. Count of hateful and non hateful sentences

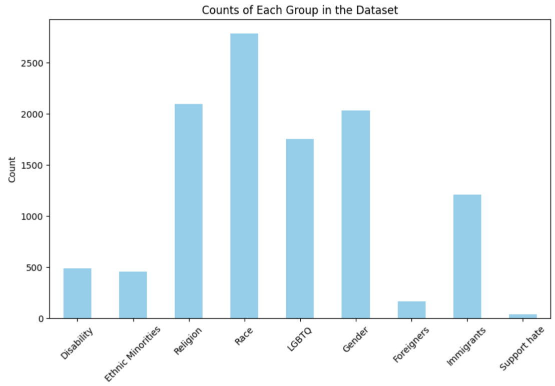

Figure 3. Minority groups representation in the dataset

3.3.1. Dataset description

We use a dataset manually created with the help of ChatGPT. It includes various types of sentences along with their corresponding corrected sentence. Some examples that can be found in the dataset are shown in Table 1.

Original sentence	Corrected sentence
She's too old to be attractive	Everyone has their own unique beauty at any age
Fucking Asians are inferior	Everyone deserves to be treated
Women can't make tough decisions	Decision-making skills are not determined by gender

Table 1. Some pairs of sentences found in the dataset

This dataset, comprising 357 pairs of phrases, is relatively small and it represents one of our primary challenges. Generating this dataset involves creating hate speech examples, a task that is often restricted by model safeguards. To address this problem, we need to make sure that the purpose of the request is clear in our prompts to ChatGPT.

3.4. Lipsync

The last step we implement is the synchronization of the lips and text. We use the Gooey.AI API Platform [8].

4. Validation

Using BERT to classify content as hatespeech or not can be very effective if a large and diverse enough dataset is used to fine tune the model.
We use the following hyperparameters : Batch size : 32, Learning rate : 1e-5, Num epochs : 10.

In our case, we obtain good results which allow us to detect efficiently the hate speech. The scores of our BERT classifier are illustrated in Table 2. As previously mentioned, the main challenge of this miniproject is the generation of non harmful content. Our language model (LLM) is fine-tuned using the unsloth library [9] and the PyTorch framework, focusing on quantized versions of Llama 3 to optimize memory usage. The process begins with loading pre-trained models, such as ”unsloth/llama-2-7b-bnb-4bit” and others, using 4-bit quantization. This setup is essential for handling large sequence lengths efficiently. Some examples of the performance of the fine-tuned Llama3 model are shown in Table 3.

	precision	recall	f1-score	support
non-hate	0.79	0.82	0.80	494
hate	0.82	0.80	0.81	535
weighted scores	0.805	0.81	0.805	-

Table 2. Results of the BERT classifier

To test our implementation we use some videos from the dataset HateMM [3]. We observe that our model is able to detect hate speech from the selected videos and the LLM can generate a politically correct sentence. To show a concrete example of our model performances, we gave as input the following video containing some hate speech: original video link and it returned this modified video transformed video link. For the lipsync, we choose to use the voice of the opposite sex to clearly indicate that the video is AI-generated. (see the Ethics section 5).

Original sentence	Transformed sentence
Women belong at home, not at work	Everyone should have the opportunity to pursue their career goals without discrimination
Black people are all thieves	Everyone deserves to be treated with respect and fairness
Fuck all religions	Respect for all religions is important

Table 3. Performance of the fine tuned LLM

5. Ethics

It is important to address some ethic concerns about this project.
The dataset created to fine tune the llama3 model requires to either ask ChatGPT to generate hate speech, or to manually add it. In either case the execution is questionable, and it is debatable whether or not such methods should be used.
In addition, our project raises the following question: is it acceptable to alter someone’s statements without their knowledge, if the purpose is to censure the hate speech? We consider that in our case it is important to be transparent towards what we do. Our approach is believed to protect younger people from violence and discrimination, we thus believe that as long as we make it clear that we alter someone’s sayings, the censorship is justified to maintain the balance between protection and freedom.

6. Conclusion

The different steps implemented in this project can be useful to foster safer online spaces, however their collective application might not always be prudent. For example, detecting hate speech is a promising tool to censor online audio or video content, given that the detection is good enough. However, it could become dangerous to also accidentally censor controversial opinions. Therefore, if the hate speech classification tool was to fall into the hands of, for instance, a government or other authority with intentions to censor its population, this tool would make censorship even more potent and effortless.
The generation of more balanced politically correct content is the most challenging task, as depending on the context, the transformed sentence should vary quite a lot. As mentioned earlier, changing one’s words in order to censor hate speech might be worse than simply censoring and indicating it. Indeed, once people’s voiced opinions are altered online, it becomes very difficult to know who said what. This tool might thus not be implementable.
The lip-syncing tool raises similar concerns. If this tool is to improve to the point where it is impossible to distinguish between an original video and its transformed content, this could pose a substantial risk. Therefore, we advocate for restricting the use of this tool to comedic purposes or in film production to streamline the production process, thereby minimizing the number of takes.

References

[1] B. Vidgen, T. Thrush, Z. Waseem, and D. Kiela, "Learning from the worst: Dynamically generated datasets to improve online hate detection," 2021.

[2] U. Nations, "Understanding hate speech," n.d.

[3] M. Das, R. Raj, P. Saha, B. Mathew, M. Gupta, and A. Mukherjee, "Hatemm: A multi-modal dataset for hate video classification," 2023.

[4] Zulko, "Moviepy: a python library for video editing, which can be used for basic operations (like cutting, concatenations, title insertions, video compositing (a.k.a. non-linear editing), video processing, and creation of custom effects)," 2023.

[5] A. Cannon, "speech_recognition."

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[7] AI@Meta, "Llama 3 model card," 2024.

[8] GOOEY.AI, "Lipsync with tts." https://gooey.ai/lipsync-maker/api/, 2023. Accessed: May, 2024.

[9] unsloth, "Unsloth for finetuning llamma 3 git hub," n.d.