FSP Finder

Foul speech pattern (FSP) detector and automatic censoring tool

Members: Jared Able, Duncan Clark, Elzbieta Polak, Shuo Yan

Hugging Face Spaces
https://huggingface.co/spaces/dac202/fsp-finder

GitHub Repository
https://github.com/dclark202/auto-censoring

Erdos Institute Deep Learning Bootcamp (Summer 2025)

FSP Finder

AI Powered explicit content detector and automatic censoring tool

Project description

Radio stations and other public broadcasters must screen audio for airplay. Airing explicit content (curse words, drug references, violent or sexuall explicit material) can lead to costly FCC fines. Edited versions of tracks exist, but not for all media.

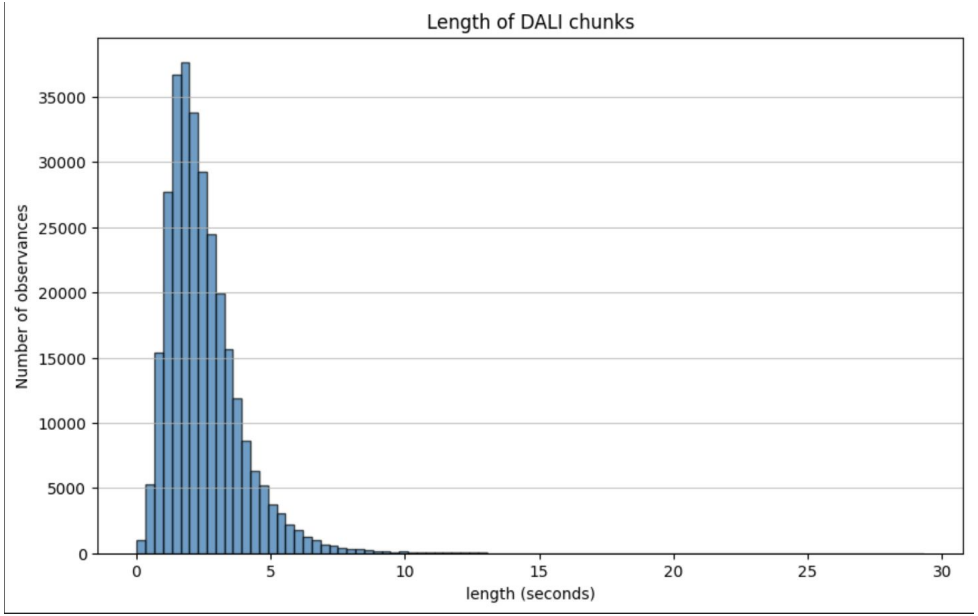
This projects creates an easy to use tool to automate the process of searching for explicit content and censoring tracks. We use a fine tuned automatic speech recognition model (OpenAI whisper) to create a vocals transcript with word timestamps for an input file. Key phrases are searched for within the transcript and the vocals are automatically censored at those times.

The tool is packaged with a web interface made in Gradio. The user can upload single files or process songs in batches, and is presented with the full transcript of each track along with a link to the official lyrics to the song (via genius.com, if possible) with a correctness score of the machine created transcript.

An entire album can be processed in a matter of minutes, edited files are presented to the user with the original file's metadata.

DALI dataset

Meseguer-Brocal, Cohen-Hadria, and Peeters (2019)
Available at https://zenodo.org/records/2577915



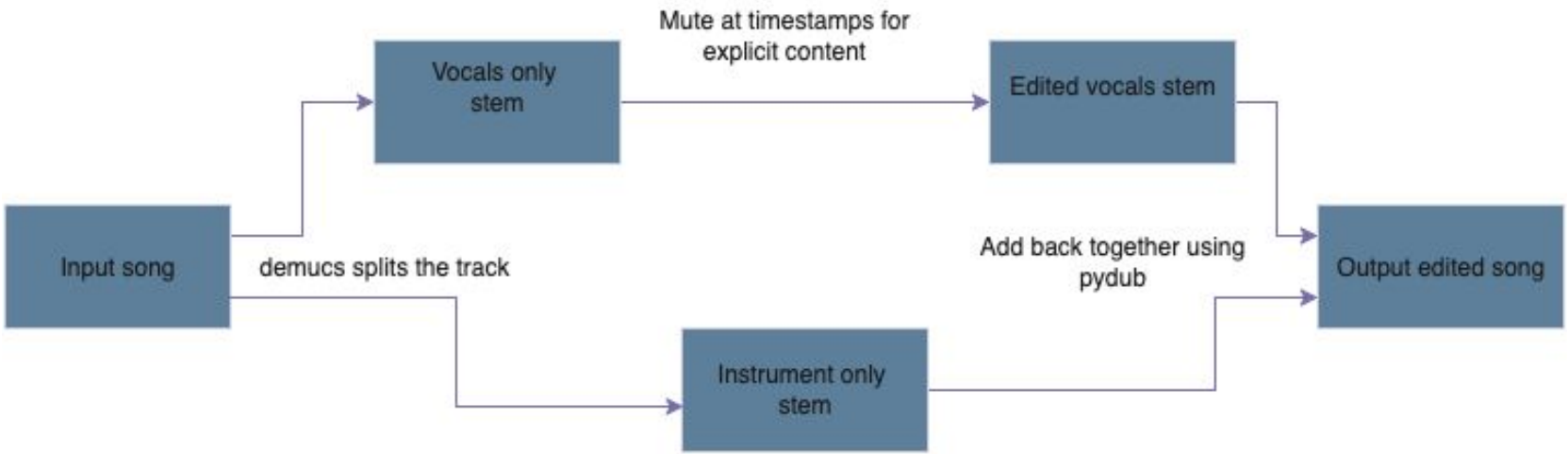
Dataset information

- Over 5000 tracks from a wide variety of genres
- Vocals stems extracted and tracks split into short “line-level” audio clips
- Extensive cleaning of transcripts was needed
- Roughly 295,000 total segments
- Each segment was run through a toxicity filter, those which were marked as toxic were selected for training (roughly 2,000 total) to bias the model towards high sensitivity on explicit content

Note: whisper’s max processing time is 30 seconds

Processing pipeline

- demucs stems are saved as .wav files for best quality
- metadata is automatically transferred to edited song from original track
- FSP Finder works with most standard audio formats (.mp3, .wav, .m4a, etc.)



Dataset preparation

Source: DALI dataset (2019)
https://zenodo.org/records/2577915

*DALI contained many misspellings as well as words with spaces placed in between syllables (e.g., whe ther) or concatenated together (e.g., andalways) making the data unsuitable for training.

Step 1 / Transcripts are extracted from DALI dataset and corresponding music files obtained via the youtube video links in DALI

Step 2 / Each track has its vocals stem extracted by demucs and is then broken into pieces as identified by “lines” in DALI. These are 3–5s long on average. Roughly 295,000 total segments are created.

Step 3 / The line-level transcripts are cleaned* and a 70/15/15 train/val/test split is created at this point, keeping segments from the same song contained within the same group.

Step 4 / Line segments are run through a fine-tuned toxicity filter to determine if they qualify as “explicit” (i.e., contain curse words, reference drugs, are sexually crude, etc.). Only those marked as toxic are used for training, to bias the model for sensitivity on explicit content.

Training specifications

Speech transcription models considered

- OpenAI Whisper base, medium.en, large-v3 https://github.com/openai/whisper
- Wav2Vec2* (https://huggingface.co/docs/transformers/en/model_doc/wav2vec2)
- Facebook s2t-small-librispeech-asr (https://huggingface.co/facebook/s2t-small-librispeech-asr)

OpenAI **whisper-medium.en** (“.en” = english only) selected based on performance on training and validation sets.

Total parameters: ~769 million

Trainable parameters: (~7% of total)

- LoRA config to train the q_proj and v_proj attention head
- LM layer (proj_out)

Training method:

- Linear warmup learning rate scheduler with to max rate of 6.25e-6
- MER (match error rate) used as performance metric on validation set. Training halted after three consecutive rounds of non-increasing MER

Whisper-timestamped used to transcribe full songs with word timestamps with fine-tuned model

*Wav2Vec2 thrown out after initial testing as its performance we deemed to be subpar

Training results

- **WER** (word error rate) is a correctness measurement for text transcription models.
- **MER** (match error rate) is a normalized variant with range between 0 and 1.
- Lower score is better: **0.0 for either means the predicted text is a perfect match with the labels**

Performance on testing set

Model	**WER**	**MER**
OpenAI whisper-medium.en baseline	0.64305	0.58424
OpenAI whisper-medium.en fine-tuned (LM layer only)	0.59760	0.53377
OpenAI whisper-medium.en fine-tuned (LM layer + LoRA)	**0.52992**	**0.48113**
Facebook s2t-small-librispeech-asr baseline	0.92058	0.83982
Facebook s2t-small-librispeech-asr fine-tuned (LoRA only)	0.62333	0.58148

Terminology:

- LM layer is the final, fully connected layer of the transformer (also called `proj_out`).

- **LoRA** (Low-rank approximation) is an approximation tool for training the attention head of the transformer.

Note: WER and MER only tell part of the story. By training on explicit content, we aim to maximize sensitivity in detecting curse words (and other foul speech).

Gradio web interface

Available at https://huggingface.co/spaces/dac202/fsp-finder

- Easy to use interface for creating high quality edited files of .mp3 files
- Process files one at a time or in batches

This interface can also be run locally following the instructions in the GitHub repo.

Note: running this app in any reasonable amount of time will require a CUDA enabled GPU with minimum 12 GB of VRAM

Ideas for future implementation

- Option for the user to add their own words to the list of words to be censored by highlighting additional words in the full transcript provided by the model.
- Method for censoring "explicit sounds", i.e., non-vocals noises that may be offensive (gun shots, sexually explicit sounds, etc.).
- Use of a language model to detect context-dependent explicit words (i.e., references to specific drugs or firearms) within lines marked as explicit.

FSP Finder

Foul speech pattern (FSP) detector and automatic censoring tool

Hugging Face Spaces https://huggingface.co/spaces/dac202/fsp-finder
GitHub Repository https://github.com/dclark202/auto-censoring

Project video https://youtu.be/csp4E_csyco

```
Credits
  -   Man with hat ............. Duncan Clark
  -   Man with sunglasses ...... Duncan Clark
  -   1st voiceover ............. Elzbieta Polak
  -   2nd voiceover ............. Shuo Yan
  -   Disclaimer voiceover ..... Jared Able
  -   Videographer.............. Amanda Luff
```