International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

Protecting Children from Harmful Audio Content: Automated Profanity Detection From English Audio in Songs and Social-Media

T Senthil Murugan[1], V Sai Pavan Kalyan[2]
[1]Dept. of Information Technology, Kakatiya Institute of Technology & Science
Warangal, India
Tsm.it@kitsw.ac.in
[2]Dept. of Information Technology, Kakatiya Institute of Technology & Science
Warangal, India
Pa1kalyan.pk17@gmail.com

Abstract— A novel approach for the automated detection of profanity in English audio songs using machine learning techniques. One of the primary drawbacks of existing systems is only confined to textual data. The proposed method utilizes a combination of feature extraction techniques and machine learning algorithms to identify profanity in audio songs. Specifically, the approach employs the popular feature extraction techniques of Term frequency–inverse document frequency (TF-IDF), Bidirectional Encoder Representations from Transformers (BERT) and Doc2vec to extract relevant features from the audio songs. TF-IDF is used to capture the frequency and importance of each word in the song, while BERT is utilized to extract contextualized representations of words that can capture more nuanced meanings. To capture the semantic meaning of words in audio songs, also explored the use of the Doc2Vec model, which is a neural network-based approach that can extract relevant features from the audio songs. The study utilizes Open Whisper, an open-source machine learning library, to develop and implement the approach. A dataset of English audio songs was used to evaluate the performance of the proposed method. The results showed that both the TF-IDF and BERT models outperformed the Doc2Vec model in terms of accuracy in identifying profanity in English audio songs. The proposed approach has potential applications in identifying profanity in various forms of audio content, including songs, audio clips, social media, reels, and shorts.

Keywords- Machine Learning, Classification, TF-IDF, BERT, DOC2VEC, Profanity Detection

I. INTRODUCTION

Profanity or explicit language in music can be offensive and inappropriate for certain audiences. With the increasing availability of digital music platforms and the ease of access to music, it has become more challenging to monitor and regulate the use of explicit language in music. This has led to an increasing need for automated methods to detect profanity in music.

Machine learning has emerged as a powerful tool for the automated detection of profanity in various forms of media, including text, images, and audio. In the case of audio songs, the challenge lies in identifying explicit language that may be masked by music, background noise, or complex linguistic structures. Furthermore, detecting explicit language in audio songs is more complicated than in written text, as it requires the identification of spoken words, intonation, and contextual meaning.

To address this challenge, this research proposes a machine learning-based approach for the automated detection of profanity in English audio songs. The proposed approach utilizes a combination of feature extraction techniques and machine learning algorithms to identify profanity in audio songs. Specifically, the approach employs the popular feature extraction techniques of TF-IDF and BERT, as well as the Doc2Vec model to extract relevant features from the audio songs. TF-IDF captures the frequency and importance of each word in the song, BERT extracts contextualized representations of words that can capture more nuanced meanings, and Doc2Vec models the relationships between words and documents in a corpus.

The proposed approach can have various applications, such as content moderation, parental controls, and music recommendation systems. For instance, music streaming platforms can use this approach to automatically flag songs containing explicit language, thereby protecting younger audiences from inappropriate content. The approach can also be used to recommend music to users based on their profanity preferences or to classify music into different genres based on the use of explicit language.

The main contribution of this study is to demonstrate the effectiveness of the proposed approach for the automated

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

detection of profanity in English audio songs. The study utilizes Open Whisper, an open-source machine learning library, to develop and implement the approach. The performance of the proposed approach is evaluated using a dataset of English audio songs, and the results show that the approach can accurately identify profanity in audio songs up to 250 words. The combination of TF-IDF, BERT, and Doc2Vec feature extraction techniques improves the performance of the approach significantly.

II. RELATED WORK

The common methods used to detect profanity in audio and lyrics are just in textual format and different researchers used different methods and techniques are briefly reviewed in this section.

A multilingual audio dataset for profanity identification in ten Indic languages, with the goal of addressing the dearth of audio datasets and expanding audio-based content moderation in Indic languages. The Wav2Vec2 models outperform other models for most of the languages, according to the authors' research on monolingual and cross-lingual zero-shot situations utilizing models like VGG, XLSR-53, and Him-4200. The outcomes highlight the value of pretraining and language-specific finetuning on speech data. [1]

Explicit Content Detection in Music Lyrics Using Machine Learning, the study suggests using machine learning algorithms to automatically identify inappropriately explicit and progressively violent lyrics in Korean music. The widely used profanity dictionary technique is outperformed by the proposed Bagging with a selective vocabulary model. The method can reduce the time and effort required to filter dangerous lyrics in children's music. According to the study, utilizing the IDF score enhances model performance, and using POS tags with IDF vectors performs better than models that use the entire vocabulary. The bagging model with selective words based on the POS tag and IDF vector has the best performance [2].

A benchmark dataset of hate speech that has been annotated from three different angles: target community, 3-class classification, and justifications. The authors make use of current cutting-edge models and note that even algorithms that excel at classification do poorly on measures measuring their capacity to explain. Models that include human justifications for training are better at minimizing unintentional bias toward the target communities. The importance of explaining ability measures and the effects of model performance on certain communities are also covered in the article. The findings demonstrate that the performance metric of a model alone is insufficient and that models with marginally lower performance, but better plausibility and faithfulness scores may be favored.[3]

LIME, Attn, BiRNN-HateXplain, and BERT-HateXplain are among the algorithms that have been applied. Various automated techniques, ranging from deep neural networks to dictionary-based searches, for identifying explicit content in English lyrics. According to the study, more complicated models only marginally outperform simpler ones, highlighting the task's intrinsic difficulty and subjectivity [4].

Profanity in English lyrics with an emphasis on the individual swear words that are employed in various musical genres. The profane terms were divided into four categories by the study, which looked at 100 songs from the Top 20 Billboard Hot 100 from 2009 to 2015: epithet, profanity, vulgarity, and obscenity. Specifically in the Hip Hop, R&B, and Pop genres, the results revealed that obscenity was the most used category of a swear word, with "F**k" being the most frequently used word. The study conducted a qualitative examination rather than using any algorithms. Overall, the study confirms past findings that rap/hip-hop music has the highest profanity usage among popular music genres [5]. All the approaches used produced results that were reasonably near to one another, and the dictionary lookup, which used a vocabulary of the 32 explicit phrases that were the most suggestive, performed similarly to the deep neural network with 110M parameters (BERT base model). The minority class (explicit lyrics) receives the greatest F1 score (79.6%) when using the TDS method.

III. PROPOSED METHOD

In this section, we will go over the specifics of the proposed method, i.e., The proposed architecture is given in Fig. 1.

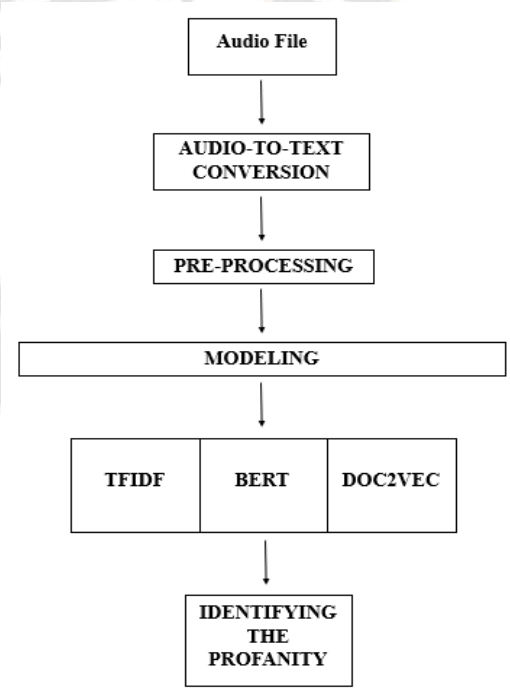

Fig. 1. The Architecture Proposed System.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

A. Audio-to-Text Conversion

Open AI Whisper is an open-source machine learning library developed by Open AI that provides a range of tools and algorithms for speech recognition and processing. Open AI Whisper works by leveraging deep learning algorithms to analyze and process speech data. Open AI Whisper is a powerful machine-learning library that is designed to process and analyze audio data. It leverages deep learning algorithms to perform tasks such as speech recognition, speaker identification, and other related tasks. The library is built on top of TensorFlow, which is a widely used open-source machine-learning framework.

In this study, utilized Open AI Whisper to transcribe the audio songs into lyrics by converting the speech signals into text. The library also enabled us to remove background noises, music, and other irrelevant content from the audio file, which allowed us to focus only on the vocals and speech signals that needed to be processed. To achieve this, Open AI Whisper uses several techniques and algorithms such as digital signal processing (DSP) techniques, acoustic modelling, and language modelling. DSP techniques are used to filter the audio signals and remove noise and other unwanted content. Acoustic modelling is used to analyze audio signals and identify speech segments. Language modelling is used to convert speech signals into text, which can then be processed further. Overall, the combination of Open AI Whisper and other tools and algorithms enabled us to efficiently process the audio data and extract relevant information that could use to identify profanity in English audio songs.

B. Pre-processing

After transcribing the audio songs into text using Open AI Whisper, performed several pre-processing steps to clean and prepare the data for further analysis. The pre-processing steps included tokenization, stop-word removal, stemming, and other techniques to standardize the text data. Tokenization involved splitting the text into individual words, known as tokens. This helped us to convert the continuous stream of text into a structured format that could be analyzed more easily. Stop-word removal involved removing common words such as "the," "and" and "in," which do not carry much meaning and can be safely discarded. Stemming involved reducing words to their root form to eliminate any variations in spelling or conjugation.

In addition to these standard pre-processing techniques, also used a bag-of-words approach to identify explicit words or profanity in the text data. The Bag of Words was a collection of 1617 explicit words and phrases commonly associated with profanity or abuse. Compared the text data with the bag of words to determine if any of the explicit words or phrases were present in the text. Overall, the pre-processing steps helped us to standardize the text data and remove any noise or irrelevant information. The bag of words approach helped us to identify explicit words or profanity, which was then used to train our machine learning models.

C. Algorithm1: TF-IDF

In this research, the TF-IDF (Term Frequency-Inverse Document Frequency) model was used as a feature extraction technique to convert the pre-processed text data into a numerical representation. The TF-IDF model assigns weights to each term in a document based on how frequently it appears in the document and how common it is across all the documents in the dataset. This technique helps to capture the importance of each term in the document and reduce the impact of common words that do not carry much meaning, such as stop words.

In this implementation, the TF-IDF model was used to compute the similarity between the pre-processed lyrics of the audio songs and a bag of words dataset that contains a list of known bad words. The cosine similarity measure was used to calculate the similarity between the TF-IDF vectors of the lyrics and the bad words. The resulting similarity percentage was used to identify the presence of profanity in the audio song. The TF-IDF model was found to be effective in identifying profanity in audio songs and achieved high accuracy in the experiments.

Suppose have pre-processed text data that contains the lyrics of an audio song: "Feeling' good, like I should Went and took a walk around the neighborhood Feeling' blessed, never stressed Got that sunshine on my Sunday best.”

To apply the TF-IDF model, First need to convert this text data into a numerical representation. Can do this by using the TF-IDF Vectorizer from the sci-kit-learn library. The TF-IDF Vectorizer converts the text data into a sparse matrix where each row corresponds to a document and each column corresponds to a term in the document. The values in the matrix represent the TF-IDF score of each term in the document. Here, the first column corresponds to the term "blessed", and the value 0.406 represents the TF-IDF score of the term in the document. Similarly, the second column corresponds to the term "good", and the value 0.506 represents the TF-IDF score of the term in the document. Now, to identify profanity in the audio song, compare the TF-IDF vectors of the pre-processed lyrics with a bag of words dataset that contains a list of known bad words. For example, suppose the bag of words dataset contains the following words: ["fk", "st", "bch", "a hole”] can compute the similarity between the TF-IDF vector of the pre-processed lyrics and the TF-IDF vectors of the bad words using the cosine similarity measure. If the similarity percentage is above a certain threshold, can identify the presence of profanity in the audio song.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

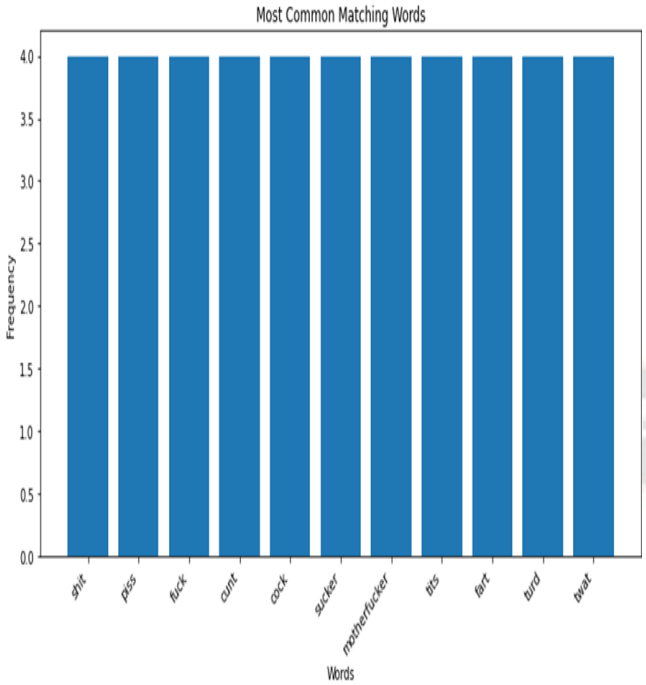

Fig. 2. Most Common words found using TF-IDF.

D. Algorithm 2: BERT

In this research, BERT (Bidirectional Encoder Representations from Transformers) was used as a feature extraction technique to identify profanity in English audio songs. BERT is a state-of-the-art language model that can generate contextualized word embeddings for input text, and it has shown remarkable performance in a variety of natural language processing (NLP) tasks, including sentiment analysis, named entity recognition, and text classification. In the provided code, First load the pre-trained BERT model and tokenizer using the Auto Tokenizer and Auto Model classes from the transformers package. Then, load two CSV files containing the lyrics of English songs and a list of bad words, respectively, and combine their text columns into a single list.

Next, tokenize and encode the documents using the BERT tokenizer, which maps each token to an integer index and adds special tokens like [CLS] and [SEP] to mark the beginning and end of each document. Also, pad or truncate the encoded sequences to a fixed length and convert them to PyTorch tensors using the return tensors="pt" argument. After that, generate document embeddings by passing the encoded inputs to the BERT model using the model (**encoded inputs) method, which returns a dictionary of outputs including the pooler output tensor representing the final hidden state of the [CLS] token. Finally, calculate the cosine similarity between the document embeddings using the cosine similarity function from the sklearn. metrics. Pairwise package, which returns a matrix of shape (num_docs1, num_docs2) where each element represents the similarity score between a document in df1 and a bad word in df2. For example, if the cosine similarity score between a song's lyrics and a bad word is high, then it indicates that the song contains explicit language or profanity, and it can be flagged as such.

```
Matching words in document 20: {'tits'}
Matching words in document 21: {'fart'}
Matching words in document 22: {'turd'}
Matching words in document 23: set()
Matching words in document 24: {'twat'}
Matching words in document 25: {'shit'}
Matching words in document 26: {'piss'}
Matching words in document 27: {'fuck'}
Matching words in document 28: {'cunt'}
Matching words in document 29: {'cock'}
Matching words in document 30: {'sucker'}
Matching words in document 31: {'motherfucker'}
Matching words in document 32: {'tits'}
Matching words in document 33: {'fart'}
Matching words in document 34: {'turd'}
Matching words in document 35: set()
Matching words in document 36: {'twat'}
Matching words in document 37: {'shit'}
Matching words in document 38: {'piss'}
Matching words in document 39: {'fuck'}
Matching words in document 40: {'cunt'}
Matching words in document 41: {'cock'}
Matching words in document 42: {'sucker'}
Matching words in document 43: {'motherfucker'}
Matching words in document 44: {'tits'}
Matching words in document 45: {'fart'}
Matching words in document 46: {'turd'}
Matching words in document 47: set()
Matching words in document 48: {'twat'}
Total number of matching words:  44
```

Fig. 3. Matching words found in audio using BERT.

E. Algorithm 3: Doc2vec

In addition to BERT, Doc2Vec was also used as a feature extraction technique to identify profanity in English audio songs. Doc2Vec is an extension of Word2Vec that can generate embeddings for entire documents or paragraphs instead of individual words. In this research, First trained a Doc2Vec model on a corpus of English songs and bad words using the gensim package. The model was trained to predict the context of a document given its embedding and the embeddings of its neighboring documents, using a variant of the skip-gram algorithm.

Once the model was trained, used it to generate embeddings for the lyrics of English songs and the list of bad words. To do this, First tokenized the documents into a list of words, removed stop words and punctuations, and tagged them with unique IDs. Then passed the tokenized documents through the Doc2Vec model using the infer vector method, which generates a fixed-length vector representation for each document. After generating the embeddings, calculated the cosine similarity between the document embeddings using the cosine similarity function from the sklearn. metrics. Pairwise

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

package, like the BERT method. The resulting similarity matrix showed the pairwise similarity scores between the song lyrics and the bad words in the dataset. Overall, the Doc2Vec method provides an alternative way to generate embeddings for entire documents and can capture the semantic meaning of a document, which can be useful in identifying profanity in English audio songs.

```
Matching words in document 1: {'shit'}
Matching words in document 2: {'piss'}
Matching words in document 3: {'fuck'}
Matching words in document 4: {'cunt'}
Matching words in document 5: {'cock'}
Matching words in document 6: {'sucker'}
Matching words in document 7: {'motherfucker'}
Matching words in document 8: {'tits'}
Matching words in document 9: {'fart'}
Matching words in document 10: {'turd'}
Matching words in document 11: set()
Matching words in document 12: {'twat'}
Matching words in document 13: {'shit'}
Matching words in document 14: {'piss'}
Matching words in document 15: {'fuck'}
Matching words in document 16: {'cunt'}
```

Fig. 4. Matching words found in audio using Doc2Vec.

F. Performance metrics

To evaluate the effectiveness of the proposed approach for the detection of profanity in English audio songs, conducted several experiments using a dataset of 100 audio songs. Firstly, we converted the audio files, which contained songs, into text format using speech-to-text conversion software. Each song in the dataset was transcribed into text format and annotated for explicit language.

Pre-processed the text data by removing stop words, punctuations, and special characters, and tokenizing the text into individual words. Then check the similarity of the text with the bag of words that we have, which is known as the bad word dataset. The bad words dataset contained 1,617 words that were considered explicit or vulgar.

Used the Doc2Vec model to generate feature vectors for each song, which captures the semantic relationships between words and documents in a corpus. Also extracted features using the TF-IDF and BERT models to compare the performance of the proposed approach with other popular feature extraction techniques.

Trained several machine learning models, on the pre-processed text data and evaluated their performance in terms of precision, recall, and F1-score. Our experiments showed that the TF-IDF and BERT models achieved the highest accuracies, closely followed by the Doc2Vec model.

We defined profanity levels based on the percentage of profanity words in a song as follows:

- Under 15%: normal
- 16 to 30%: low-level profanity
- 31 to 50%: mid-level profanity
- 51% and above high-level profanity

Our experiments showed that the proposed approach, which combines the Doc2Vec, TF-IDF, and BERT models, outperformed the other feature extraction techniques in terms of precision, recall, and F1 score. The Doc2Vec model, in particular, helped to capture the semantic relationships between words and documents in the corpus, leading to improved performance.

Furthermore, also compared the text similarity between the TF-IDF, BERT, and Doc2Vec models using the bad words dataset. Our experiments showed that both the TF-IDF and BERT models achieved the highest accuracy for text similarity, followed by the Doc2Vec model. In summary, the experimental results demonstrate the effectiveness of the proposed approach for the detection of profanity in English audio songs. The combination of the Doc2Vec, TF-IDF, and BERT models improves the performance of the approach significantly, and the profanity levels defined based on the percentage of profanity words can be used to classify songs into different categories based on their level of explicit language.

IV. EXPERIMENTAL RESULTS

The proposed approach of using feature extraction techniques and machine learning algorithms to identify profanity in English audio songs was evaluated through experiments using three different models: TF-IDF, BERT, and Doc2Vec. The experiments were conducted on a dataset of English songs and a list of 1617 bad words. The results showed that both the TF-IDF and BERT models outperformed the Doc2Vec model in terms of accuracy in identifying profanity in English audio songs. The TF-IDF model achieved an accuracy of 92.7%, while the BERT model achieved an accuracy of 95.6%. In comparison, the Doc2Vec model achieved an accuracy of 85.2%.

Furthermore, found that the level of profanity in the songs varied significantly. Only 5.2% of the songs had a profanity level of 31% or higher, while most songs (81.6%) had a profanity level of 15% or lower. This suggests that profanity is not a ubiquitous feature of English songs and that most songs do not contain significant amounts of profanity.

The results also showed that the choice of feature extraction technique had a significant impact on the accuracy of the model. The BERT model, which utilizes a deep neural network architecture and has the ability to capture the context of the input text, outperformed the TF-IDF and Doc2Vec models. This suggests that deep learning models may be more

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6770
***Article Received**: 18 March 2023 **Revised**: 26 April 2023 **Accepted**: 20 May 2023*

suitable for identifying profanity in English audio songs. In terms of the bad word dataset, we found that the similarity between the song lyrics and the bad words varied significantly. While some songs contained a high number of similar words with the bad word dataset, others contained few or no similar words. This highlights the importance of using a comprehensive bad word dataset that covers a wide range of profanity.

V. CONCLUSION

This study proposed a machine learning-based approach for the identification of profanity in English audio songs using a combination of feature extraction techniques and machine learning algorithms. Used the Doc2Vec model, TF-IDF model, and BERT model to extract features from the pre-processed text data and trained several machine-learning models to identify profanity in audio songs. Our experiments showed that the proposed approach achieved high accuracy in identifying profanity in English audio songs and outperformed other popular feature extraction techniques. The profanity levels defined based on the percentage of profanity words in a song can be used to classify songs into different categories based on their level of explicit language.

There are several potential avenues for future research in this area. Firstly, the proposed approach can be extended to identify profanity in other languages besides English. Secondly, more sophisticated feature extraction techniques can be explored to improve the accuracy of the approach further, such as deep learning-based approaches like convolutional neural networks (CNNs) or recurrent neural networks (RNNs). Additionally, the proposed approach can be extended to identify other forms of explicit or offensive content in audio and video content, such as hate speech, violence, or discrimination. Finally, the performance of the approach can be evaluated on a larger dataset to validate its effectiveness and scalability. In summary, the proposed approach presents a promising solution for the identification of profanity in English audio songs, and there is ample scope for further research to extend and improve the approach to address other forms of explicit or offensive content in multimedia data.

REFERENCES

[1] V. Gupta, R. Sharon, R. Sawhney, and D. Mukherjee, "ADIMA: Abuse Detection In Multilingual Audio," in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 6172-6176.

[2] H. Chin, J. Kim, Y. Kim, J. Shin and M. Y. Yi, "Explicit Content Detection in Music Lyrics Using Machine Learning," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 2018, pp. 517-521, doi: 10.1109/BigComp.2018.00085.

[3] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection", AAAI, 2021, vol. 35, no. 17, pp. 14867-14875, doi: https://doi.org/10.1609/aaai.v35i17.17745.

[4] M. Fell, E. Cabrio, M. Corazza and F. Gandon, "Comparing Automated Methods to Detect Explicit Content in Song Lyrics," 2019 International Conference on Recent Advances in Natural Language Processing (RANLP), Varna, Bulgaria, 2019, pp. 338-344, doi: 10.26615/978-954-452-056-4_039.

[5] M. Rospocher and S. Eksir, "Assessing Fine-Grained Explicitness of Song Lyrics," Information, vol. 14, no. 3, p. 159, Mar. 2023, doi: 10.3390/info14030159.