International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 622-629
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0087

Sentiment Synthesis Transforming YouTube Comments into Strategic Insights

Uma Maheswari G[1], Krishna Harni P[2], Sangeetha K[3], Zenmathy K P[4]
[1,2,3,4]Department of CSE, Kamaraj College of Engineering and Technology, Virudhunagar, India
Emails: *umamaheswaricse@kamarajengg.edu.in[1], 21ucs017@kamarajengg.edu.in[2], 21ucs012@kamarajengg.edu.in[3], 21ucs014@kamarajengg.edu.in[4]*

Abstract

Sentiment analysis in the domain of social media analytics is crucial for understanding public opinion, particularly on platforms like YouTube, where vast amounts of user-generated content are continuously uploaded. The challenge lies in efficiently processing live-streamed comments in real time while ensuring accurate classification into sentiment categories—positive, negative, and neutral. Existing approaches, such as those utilizing BERT-based models, face limitations in handling multilingual and code-mixed comments effectively, leading to reduced accuracy in sentiment classification. The proposed system enhances sentiment analysis system leveraging XML-RoBERTa, a transformer-based model trained on diverse multilingual datasets, improving classification accuracy and robustness. The system processes live YouTube comments in real time, categorizes them into sentiment classes, and visualizes the results using word clouds, pie charts, and bar charts. Additionally, it generates a detailed sentiment report that maps individual comments to their respective sentiment categories, offering a transparent and comprehensive analysis. Experimental results demonstrate that XML-RoBERTa outperforms the BERT model used in the base paper by offering better multilingual support and sentiment differentiation, validating the effectiveness of the proposed approach.

Keywords: *Real-Time sentiment analysis, YouTube Comments, XML-RoBERTa, Multilingual Classification, Data Visualization.*

1. Introduction

The rapid expansion of social media platforms, particularly YouTube, has transformed digital communication by enabling users to share opinions and engage in discussions on a global scale. However, this has also led to an overwhelming volume of user-generated content, making it difficult to manually analyse sentiments expressed in comments. Sentiment analysis plays a crucial role in understanding public opinion, identifying trends, and moderating harmful content. It enables businesses, content creators, and policymakers to gauge audience reactions, improve engagement strategies, and take necessary actions against misinformation or toxic behaviour. Additionally, sentiment analysis is valuable in tracking brand perception, analysing market trends, and detecting social issues reflected in online discussions. While significant progress has been made in automating sentiment classification using Natural Language Processing (NLP), many existing approaches struggle with handling multilingual and context-dependent expressions, particularly in languages with limited resources. [1] Current research in sentiment analysis primarily relies on transfer learning models such as BERTimbau, mDeBERTa, and GPT-based generative models, which have demonstrated strong performance for various NLP tasks. For instance, studies have shown that fine-tuning BERT-like models on domain-specific datasets improves classification accuracy. However, sentiment analysis in Portuguese YouTube comments remains a challenging task due to informal language, slang, and contextual nuances. Existing models trained on general Portuguese text often fail to capture the complexities of YouTube discourse, leading to suboptimal performance. [2] A machine learning-based method to detect hate speech on Twitter, aiming to automatically recognize offensive and hateful comments. The approach relies on specific words and patterns (unigrams and pattern-based features) that are collected from a dataset and used to train a classification model. These language patterns help the system identify different types of hate speech more effectively. To test how well this method works, researchers analyzed 2,010 tweets and achieved 87.4% accuracy in identifying whether a tweet is offensive or not (binary classification). When categorizing tweets into three groups - hateful, offensive, or clean - the model reached 78.4% accuracy (ternary classification). The findings show that using automatically collected word patterns improves hate speech detection, making this approach a useful step toward better moderation of harmful content on social media. Since the model depends on features extracted from the training set, it may struggle with generalization when applied to new datasets or other social media platforms like YouTube, where language use differs. Additionally, the approach faces data imbalance issues, as hate speech is significantly less frequent than neutral or offensive language, potentially leading the model to favour the majority class, reducing recall for hate speech detection. [3] Detecting hate speech in online social networks (OSNs) using an advanced deep learning model. The researchers introduced a multi-channel convolutional-BiLSTM network with an attention mechanism to classify hateful messages. Their model uses different word representation techniques to better understand the meaning of words in different contexts. It processes text through multiple convolutional filters with different sizes to capture various language patterns. The encoded text then passes through a two-layer BiLSTM (Bidirectional Long Short-Term Memory) network that further improves understanding by considering both past and future words in a sentence. An attention layer assigns more importance to key words, helping the model focus on the most relevant parts of the text. The final classification is done using a sigmoid activation function, which determines whether a message is hateful or not. The model was tested on three Twitter datasets and outperformed five existing models, proving its effectiveness. The study also analyzed how different factors like word representations, optimization techniques, activation functions, and batch size affect the model's performance. This study also focuses only on Twitter, so the model may not perform as well on other social media platforms with different language styles. Finally, interpretability is a challenge, as deep learning models like BiLSTMs and CNNs work as "black boxes," making it harder to understand why a specific comment was classified as hate speech. To address these challenges, our study leverages XML-RoBERTa, a transformer-based model designed for multilingual text processing. XML-RoBERTa is well-suited for analysing YouTube comments due to its ability to understand contextual relationships across different languages, including Portuguese. By fine-tuning XML-RoBERTa on annotated YouTube comment datasets, we aim to improve sentiment classification accuracy and ensure more effective analysis of online discussions. Our research also explores the impact of domain-specific training data and fine-tuning strategies on model performance, providing insights into optimizing sentiment analysis for real-world applications.

2. Literature Survey

[1] Detecting Hate Speech on Twitter With the rise of social media, people from different backgrounds communicate directly, leading to more online conflicts and hate speech. Hate speech includes aggressive or offensive language aimed at specific groups based on factors like gender, race, ethnicity, or religion. Since manually reviewing all content is impossible, automated methods are needed. This study presents a machine learning approach for detecting hate speech on Twitter. It uses unigrams (single words) and patterns from training data to train a classification model. The model was tested on 2,010 tweets, achieving 87.4% accuracy in identifying offensive vs. non-offensive tweets and 78.4% accuracy in classifying tweets as hateful, offensive, or clean. [2] Preventing the Spread of Harmful Content Social networks can spread harmful content quickly. This study introduces CONTAIN, a new method to prevent this by identifying key users spreading harmful messages and limiting their influence. The method is based on analysing network structures to detect and neutralize harmful content sources. Tests showed that CONTAIN performs better than other existing methods, like NetShield and SparseShield, by requiring fewer steps and working faster. It also scales better than similar approaches,

making it a more efficient solution for reducing the spread of harmful content. [3] A Review of Hate Speech Detection Techniques The anonymity of social media has made hate speech detection more challenging. This study reviews various methods for detecting hate speech, particularly focusing on NLP and deep learning techniques. It follows the PRISMA framework to analyse research from the past decade in sources like ACM Digital Library and Google Scholar. The study explores key topics such as how hate speech is defined, processing steps, deep learning models, and the challenges in automatic detection. It identifies gaps in current models and suggests improvements, particularly in multilingual hate speech detection and accuracy enhancement. [4] Detecting Hate Speech on Twitter Using Deep Learning The growing use of Twitter has led to a rise in hate speech, making automated detection crucial. This study uses a Deep Convolutional Neural Network (DCNN) to classify hate speech. The model represents tweets using GloVe word embeddings, capturing their meaning through deep learning layers. Tests showed strong performance, with precision, recall, and F1-scores of 0.97, 0.88, and 0.92, respectively. The study confirms that deep learning techniques, especially CNNs, outperform traditional approaches in detecting hate speech on Twitter. [5] Detecting Abusive Comments on Instagram users frequently encounter abusive comments, which can have serious psychological effects. Detecting such comments in non-English languages is particularly difficult due to a lack of labelled data. This study introduces a dataset for detecting abusive comments in Turkish, called the Abusive Turkish Comments (ATC) dataset, containing 10,528 abusive and 19,826 non-abusive comments. Several machine learning models were tested, including CNN, Naïve Bayes, SVM, Decision Tree, Random Forest, Logistic Regression, AdaBoost, and XGBoost. The CNN model with oversampling performed best, achieving a micro-averaged F1-score of 0.974 and a macro-averaged F1-score of 0.973. The study contributes to developing automated systems for detecting abusive comments in Turkish social media. [6] Improving Hate Speech Detection with Multi-Channel Deep Learning Online social networks help people communicate in real time, but they also increase the spread of hate speech and misinformation. This study presents an automated method using a deep learning model called an attentional multi-channel Convolutional-BiLSTM network. The model processes text with multiple filters that capture different language patterns, followed by a BiLSTM network for classification. Tests on three Twitter datasets showed that including attention mechanisms and multi-channel processing significantly improved accuracy compared to previous methods. [7] Detecting Offensive Language in Spanish social media Detecting offensive language in Spanish texts is challenging due to data imbalance and preprocessing difficulties. This study compares traditional machine learning models, such as SVM and Decision Trees, with newer transformer-based models trained on social media data. By applying advanced text preprocessing and domain-specific training, the study improved accuracy by 6.2% compared to previous methods, setting a new benchmark for Spanish-language offensive speech detection. [8] The Importance of Context in Hate Speech Detection Many hate speech detection models struggle because they lack context. This study examines how adding contextual information affects detection accuracy. Researchers built a dataset in Rioplatense Spanish using Twitter conversations about COVID-19. They tested state-of-the-art transformer-based models and found that including context improved performance, increasing Macro F1-scores by 4.2% for binary classification and 5.5% for multi-label classification. The study highlights the importance of context-aware hate speech detection. [9] Detecting Cyber harassment in Arabic The rise of cyber harassment has created a need for better detection systems, especially for underrepresented languages like Arabic. This study introduces a two-stage optimization method combining XGBoost and SVM with a genetic algorithm to fine-tune hyperparameters. The model was tested on the Arabic Cyberbullying Corpus (ArCybC), which includes tweets from different domains. The proposed approach achieved an 88.2% accuracy rate, demonstrating its effectiveness in detecting offensive Arabic-language tweets. [10] Sentiment Analysis of

YouTube Comments Analysing YouTube comments helps understand user behaviour, preferences, and engagement patterns. This study classifies YouTube comments into positive, negative, and neutral sentiments using NLP techniques. By identifying sentiment trends, the research helps content creators tailor their content to audience preferences, improving viewer engagement. The study highlights the importance of sentiment analysis in enhancing user experience and content strategy. [11] Hate Speech Detection in European Portuguese Detecting hate speech in underrepresented languages, such as European Portuguese, remains a challenge. This study explores the use of transfer learning models like BERTimbau, mDeBERTa, and GPT-based approaches for detecting hate speech in Portuguese social media. Tests on annotated datasets from YouTube and Twitter showed that a fine-tuned BERTimbau model performed best on YouTube comments with an F-score of 87.1%, while GPT-3.5 achieved the highest accuracy for Twitter data. The study emphasizes the importance of domain-specific pretraining and contextual prompts for improving hate speech detection in Portuguese. [12] Trends in YouTube Comment Sentiment The massive growth of user-generated content has increased interest in sentiment analysis on YouTube. This study analyses YouTube comment sentiment trends using machine learning [13-15]. Six classifiers—Naïve Bayes, SVM, Logistic Regression, Decision Tree, KNN, and Random Forest—were tested and evaluated using accuracy and F1-score metrics. The findings suggest that sentiment analysis can reveal connections between online discussions and real-world events, helping researchers understand social media trends better.

3. Proposed System

The proposed system uses XML-RoBERTa model for sentiment analysis of YouTube comments because it is a powerful transformer-based model trained on 2.5 terabytes of multilingual text from 100 different languages. This vast dataset allows XML-RoBERTa to effectively understand context, detect sentiment nuances, and handle various languages in YouTube comments. Unlike traditional machine learning approaches that require manual feature extraction, XML-RoBERTa leverages deep learning techniques to capture complex relationships between words and their meanings. It provides superior accuracy in sentiment classification, particularly when dealing with informal language, slang, and multilingual content. To fetch real-time YouTube comments, the YouTube API v3 is used, requiring an API key to access comments from a specific video given by user. Once retrieved, the comments go through a cleaning process where unnecessary characters like special characters, HTML tags, special characters and whitespaces are removed then text is converted to lowercase, and tokenization is applied (Length truncation limit to 512 tokens). After preprocessing, the comments are passed to the XML-RoBERTa model, which classifies them into three categories: positive, negative, and neutral. To make the results easy to understand, various visualizations are used. Finally, a detailed sentiment report is generated, listing each comment along with its sentiment classification. This structured report provides a clear overview of how viewers are reacting to the video, making sentiment analysis more insightful. Figure 1 depicts pipeline of the proposed approach. Tokenization Using Byte-Pair Encoding (BPE): Before XML-RoBERTa processes the input text, it applies Byte-Pair Encoding (BPE), a sub word tokenization technique. Unlike traditional tokenization, which splits text into words, BPE breaks words into smaller sub word units based on frequency. This is particularly useful for handling rare words, different word forms, and multilingual text. consider a scenario where BPE is trained on a large text dataset containing words like:

- "run" (frequent): "I run daily."
- "runner" (frequent): "She is a fast runner."
- "running" (less frequent compared to "run"): "He is running fast."

Other words ending in "ing" (like "jumping," "playing," "walking") are also common. Based on frequency, BPE learns to break "running" into sub words instead of treating it as a whole word:

- If "running" appears frequently enough as a whole word then model may keep it as "running" (a single token).
- If "run" is very frequent but "running" is not

as common then BPE splits it as "run" + "ing", Since "run" is commonly used and "ing" appears in many words, the model reuses these sub words instead of storing "running" separately

This allows the model to handle morphologically rich languages and unknown words more effectively. If a new word appears that wasn't in training, BPE can still recognize its meaning by breaking it into known sub words. This compression technique allows XML-RoBERTa to efficiently process rare words by breaking them into meaningful subunits.

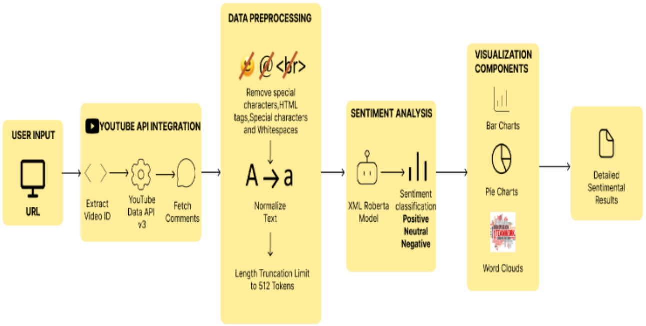


Figure 1 System Design of the Proposed Approach

3.1. Converting Tokenized Text into Numerical Representations

Once the text is tokenized, each sub word is converted into an embedding vector using a pre-trained embedding matrix. This matrix is part of XML-RoBERTa's learned parameters and maps each token to a high-dimensional vector that captures its meaning. For example: "This video is amazing!"

3.2. Output Prediction

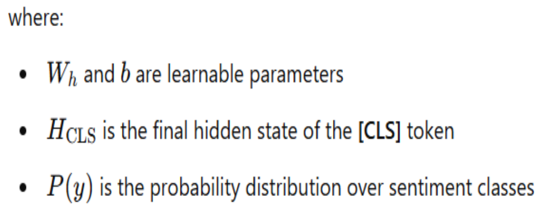
The class with the highest probability is chosen as the final sentiment label. For example, if the softmax probabilities are [0.8, 0.1, 0.1], the model classifies the comment as positive (since 0.8 is the highest). If the softmax probabilities are [0.1, 0.1, 0.8], the model classifies the comment as negative (since 0.8 is the highest). If the softmax probabilities are [0.2, 0.6, 0.2], the model classifies the comment as Neutral (since 0.6 is the highest) Tokenized as → ["This", "video", "is", "amaz", "ing", "!"] Converted to embeddings → [E1, E2, E3, E4, E5, E6] (where E represents a high-dimensional vector) These embeddings are then passed through multiple transformer layers.

3.3. Processing Through Transformer Layers

XML-RoBERTa is based on the Transformer architecture, which consists of self-attention and feedforward layers.

- **Self-Attention Mechanism:** Each token attends to every other token in the sentence using scaled dot-product attention. The model computes query (Q), key (K), and value (V) vectors for each token and determines the importance of each word in the context. This allows XML-RoBERTa to capture long-range dependencies between words, even in long sentences. Example: In the sentence "The movie was not bad", self-attention helps understand that "not" modifies "bad", making the sentiment positive rather than negative.
- **Feedforward Networks (FFN):** Each token representation is passed through a fully connected feedforward network to extract deeper features. It applies non-linearity (ReLU/GELU activation) to improve the model's expressiveness. These steps are repeated for multiple transformer layers, refining the representations at each stage.
- **Sentiment Classification Using the Final Layer:** After passing through transformer layers, the [CLS] token (which represents the whole sentence) is used for classification. The [CLS] (classification) token is a special token added at the beginning of every input text in transformer-based models like XML-RoBERTa, BERT, and RoBERTa. It acts as a summary representation of the entire sentence and is used for classification tasks. This token's output is passed to a fully connected layer (classification head) that computes probabilities for each sentiment class (positive, negative, neutral). The model applies the softmax activation function, ensuring that the output sums to 1.

$$P(y) = \text{softmax}(W_h \cdot H_{\text{CLS}} + b)$$



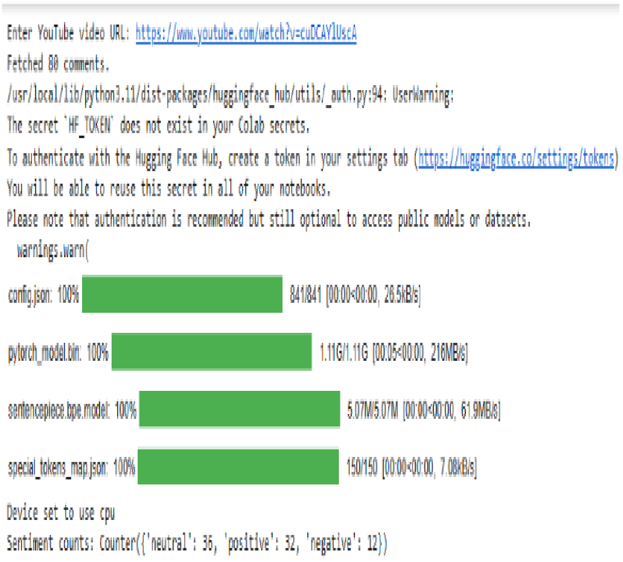
where:

- W_h and b are learnable parameters
- H_{CLS} is the final hidden state of the **[CLS]** token
- $P(y)$ is the probability distribution over sentiment classes

Figure 3 and Figure 4 gives a clear picture of how people feel about a YouTube video based on their comments. The comments have been analyzed using the XML-RoBERTa model and grouped into three categories: neutral, positive, and negative. Positive (Green): A large number of comments are supportive, appreciative, or enthusiastic about the video. Neutral (Gray): Most comments fall into this category, meaning they are either factual, balanced, or don't strongly express emotions. Negative (Red): A smaller portion of comments express criticism or negative opinions.

4. Result

The sentiment analysis of YouTube comments was done using the XML-RoBERTa model, which categorized the extracted comments into positive, neutral, and negative sentiments. The model successfully loaded all required components, including config.json, pytorch_model.bin, sentencepiece.bpe.model, and special_tokens_map.json. The device used for inference is CPU. This is depicted in Figure 2.



```
Enter YouTube video URL: https://www.youtube.com/watch?v=cuDCAYlUscA
Fetched 80 comments.
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens)
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%  841/841 [00:00<00:00, 26.5kB/s]
pytorch_model.bin: 100%  1.11G/1.11G [00:05<00:00, 216MB/s]
sentencepiece.bpe.model: 100%  5.07M/5.07M [00:00<00:00, 61.9MB/s]
special_tokens_map.json: 100%  150/150 [00:00<00:00, 7.08kB/s]
Device set to use cpu
Sentiment counts: Counter({'neutral': 36, 'positive': 32, 'negative': 12})
```

Figure 2 Model Initialization

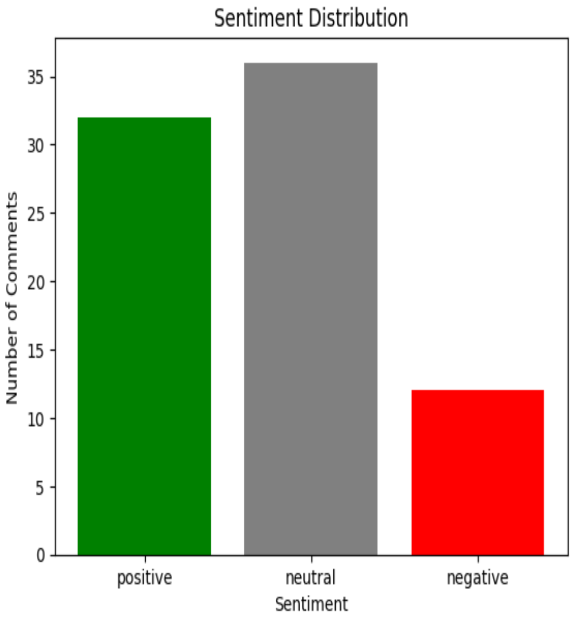


Figure 3 Bar Chart

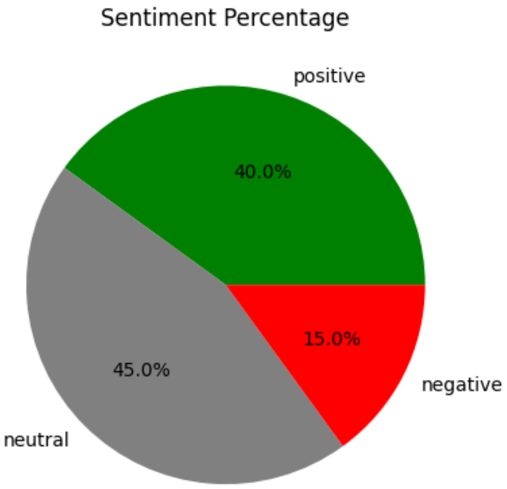


Figure 4 Pie Chart

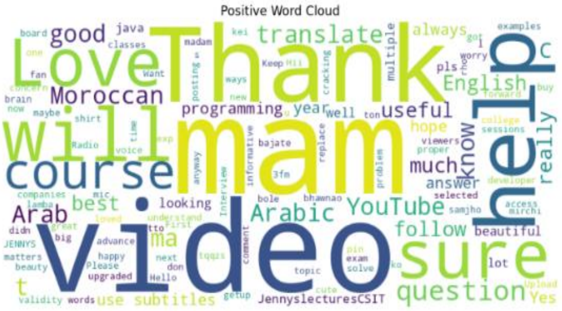


Figure 5 Positive Word Cloud

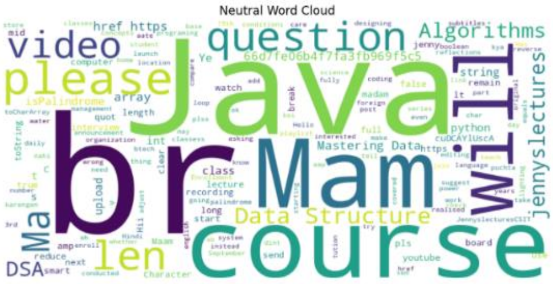


Figure 6 **Neutral Word Cloud**

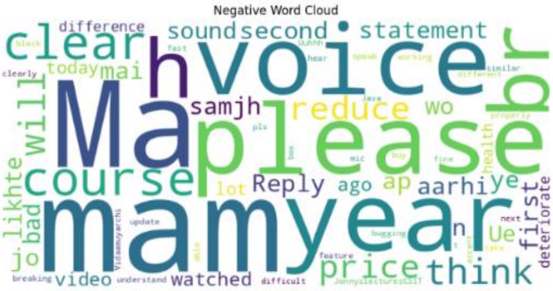
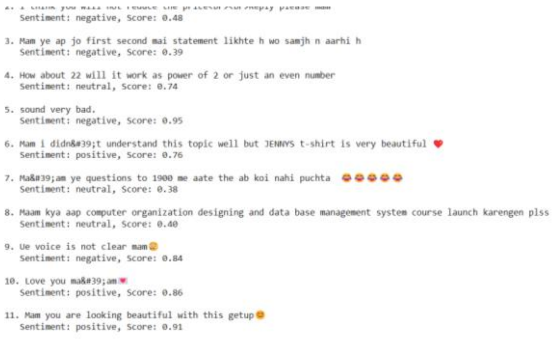


Figure 7 **Negative Word Cloud**

Figure 5, 6 and 7 visually represent the most frequently used words in comments, with larger words appearing more often. By comparing these word clouds, we can better understand audience feedback, identify strengths, and pinpoint areas for improvement.



```
2. I think you will not reduce the price[illegible]Reply please mam
   Sentiment: negative, Score: 0.48

3. Mam ye ap jo first second mai statement likhte h wo samjh n aarhi h
   Sentiment: negative, Score: 0.39

4. How about 22 will it work as power of 2 or just an even number
   Sentiment: neutral, Score: 0.74

5. sound very bad.
   Sentiment: negative, Score: 0.95

6. Mam i didn&#39;t understand this topic well but JENNYS t-shirt is very beautiful ❤️
   Sentiment: positive, Score: 0.76

7. Ma&#39;am ye questions to 1900 me aate the ab koi nahi puchta  😂😂😂😂😂
   Sentiment: neutral, Score: 0.38

8. Maam kya aap computer organization designing and data base management system course launch karengen plss
   Sentiment: neutral, Score: 0.40

9. Ue voice is not clear mam😕
   Sentiment: negative, Score: 0.84

10. Love you ma&#39;am💗
   Sentiment: positive, Score: 0.86

11. Mam you are looking beautiful with this getup😊
   Sentiment: positive, Score: 0.91
```

Figure 8 **Detailed Sentiment Report**

Figure 8 presents a detailed sentiment deport that showcases individual YouTube comments along with their sentiment classification and confidence scores. Each comment is analysed using XML-RoBERTa, which assigns one of three sentiment labels: positive, neutral, or negative. The confidence score (ranging from 0 to 1) indicates how certain the model is about its classification.

Conclusion

This project successfully automates sentiment analysis of YouTube video comments using XML-RoBERTa, providing valuable insights into audience reactions. By extracting and categorizing comments into positive, neutral, and negative sentiments, it enables a deeper understanding of viewer opinions. The bar and pie charts effectively visualize sentiment distribution, while the word clouds highlight frequently used words within each sentiment category. This approach helps content creators and researchers identify audience preferences, common discussion points, and areas for improvement. Future enhancements could include real-time monitoring, multilingual support, and topic-based sentiment classification for even richer insights.

References

[1]. Gil Ramos, Fernando Batista, Ricardo Ribeiro, (2024) "Leveraging Transfer Learning for Hate Speech Detection in Portuguese Social Media Posts", IEEE Access.

[2]. Mohd Fazil, Shakir Khan, (2023)" Attentional Multi-Channel Convolution with Bidirectional LSTM Cell Toward Hate Speech Prediction", IEEE Access.

[3]. Md Saroar Jahan, Mourad Oussalah, (2023) "A systematic review of hate speech automatic detection using natural language processing", Neurocomputing.

[4]. Pradeep Kumar Roy, Asis Kumar Tripathy, (2020) "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network", IEEE Access.

[5]. Habibe Karayigit, Çigdem Inan Acı, Ali Akdaglı, (2021), "Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods", Expert Systems with Applications.

[6]. José María Molero, (2023) "Offensive Language Detection in Spanish Social Media: Testing from Bag-of-Words to Transformers Models", IEEE Access.

[7]. Juan Manuel Pérez, Franco Luque, (2023) "Assessing the impact of contextual

information in hate speech detection", IEEE Access.

[8]. Fatima Shannaq, Bassam Hammo, (2022) "Offensive Language Detection in Arabic Social Networks Using Evolutionary-Based Classifiers Learned from Fine-Tuned Embeddings", IEEE Access.

[9]. Ch. Kesava Manikanta, A. Gowtham, (2023)" YouTube Comment Analysis Using Machine Learning", (IJRD).

[10]. Ritika Singh, Ayushka Tiwari, (2021) "YouTube Comments Sentiment Analysis", International Journal of Scientific Research in Engineering and Management.

[11]. Hajime Watanabe, Mondher Bouazizi, (2018) "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection", IEEE Access.

[12]. Elena Simona Apostol, (2024) "CONTAIN: A community-based algorithm for network immunization", Engineering Science and Technology.

[13]. Songxin Tan, Zixing Shen, (2024) "Relationship Between Cognitive Features and Social Media Engagement: An Analysis of YouTube Science Videos", IEEE Transactions on Engineering Management.

[14]. Ha Eun Jang, Seung Ho Kim, Jong Seok Jeon, (2024) "Visual Attributes of Thumbnails in Predicting YouTube Brand Channel Views in the Marketing Digitalization Era", IEEE Transactions on Computational Social Systems.

[15]. Gabriele Etta, Matteo Cinelli, (2024) "A Topology-Based Approach for Predicting Toxic Outcomes on Twitter and YouTube", IEEE Transactions on Network Science and Engineering.