Automatic Speech Recognition-Powered Subtitle Generation for Inclusive Multimedia Accessibility

Rishika Singh
Computer Science Engineering
IGDTUW
Delhi, India
rishika148btcse22@igdtuw.ac.in

Ishita Yadav
Computer Science Engineering
IGDTUW
Delhi, India
ishita075btcse22@igdtuw.ac.in

Dwiti Narang
Computer Science Engineering
IGDTUW
Delhi, India
dwiti062btcse22@igdtuw.ac.in

Abstract— **In a world where multimedia content has become a dominant mode of communication, the need for making such content accessible to diverse audiences has gained paramount importance. Subtitles, serving as textual representations of spoken language, play a pivotal role in enhancing accessibility for individuals with hearing impairments and those facing language barriers. However, the manual process of creating subtitles proves to be a bottleneck in the rapid production and dissemination of audiovisual content. This research paper is dedicated to a comprehensive exploration of Automatic Speech Recognition (ASR) technology and its innovative application to the domain of subtitle generation. Through the fusion of machine learning and natural language processing, ASR offers a promising avenue for automating the process of generating accurate, coherent, and contextually relevant subtitles. This paper elucidates the multifaceted challenges, recent technological advancements, and far-reaching implications of employing ASR systems to revolutionize the creation of subtitles, thus contributing to a more inclusive and universally accessible media landscape.**

Keywords— Automatic Speech Recognition, ASR, subtitles, accessibility, multimedia, neural networks

I. Introduction

The digital age has witnessed an explosion in the consumption of audiovisual content across platforms, ranging from educational resources to entertainment media. However, this surge in multimedia consumption has brought forth a significant challenge: ensuring that content is accessible to all, regardless of auditory or linguistic limitations. Subtitles, which entail transcribing spoken language into written text, address this challenge by enabling individuals who are deaf or hard of hearing to comprehend audio content. Additionally, subtitles transcend language barriers, making content comprehensible to a global audience. Yet, the conventional method of manually generating subtitles impedes the real-time accessibility of content, creating a time lag between content creation and its reach to diverse audiences.The emergence of Automatic Speech Recognition (ASR) technology holds transformative potential in redefining the accessibility landscape. ASR systems leverage machine learning algorithms and neural networks to convert spoken language into textual form with increasing accuracy. The integration of ASR technology into subtitle generation endeavors to streamline the process, alleviating the labor-intensive nature of manual transcription. Despite the promise of ASR, challenges abound, encompassing nuances in speech delivery, contextual disambiguation, and the integration of punctuation.

II. Challenges faced by ASR

The integration of Automatic Speech Recognition (ASR) technology into the process of subtitle generation introduces a range of complex challenges that must be overcome to achieve accurate and contextually relevant results. These challenges stem from the intricate nature of spoken language and the nuances associated with transcription. Understanding and addressing these challenges are essential for developing robust ASR systems capable of generating high-quality subtitles. Below are some of the prominent challenges in ASR for subtitle generation.

A. Variablility in speech

Human speech exhibits a vast array of variations influenced by accents, dialects, speech rates, and individual speaking styles. ASR systems must be capable of adapting to these variations to accurately transcribe spoken content into text. Accents and dialects, in particular, pose a significant challenge as they can lead to pronunciation differences that traditional ASR models might struggle to interpret correctly.

B. Overlapping Speech and Crosstalk

Conversations and dialogues often involve overlapping speech, where multiple speakers talk simultaneously or closely follow one another. ASR systems must be equipped to handle such situations, distinguishing between different speakers and disentangling their contributions to ensure accurate transcription. Similarly, crosstalk, where multiple speakers' voices intersect, presents a challenge in accurately segmenting and transcribing individual speech streams.

C. Disfluencies and Non-Fluent Speech

Disfluencies, such as repetitions, hesitations, and corrections, are common in natural speech but can complicate the transcription process. ASR systems must distinguish between meaningful content and these disfluencies to produce coherent and concise subtitles. Moreover, handling non-fluent speech, characterized by pauses, stutters, or irregular pacing, requires specialized processing techniques to maintain accuracy and readability.

D. Contextual Ambiguity

Spoken language often relies on context for disambiguation. Homophones, words with multiple meanings, and words without clear boundaries can lead to ambiguities that are typically resolved through contextual cues in speech. ASR systems need to possess contextual understanding to accurately choose the correct interpretation in such cases, ensuring that the generated subtitles convey the intended meaning.

E. Background Noise and Acoustic Conditions

ASR systems must operate effectively in diverse acoustic environments, where background noise, echoes, and reverberations can degrade speech quality. Robust noise reduction and signal processing techniques are crucial to ensure that the transcribed text accurately reflects the intended spoken content and is not unduly influenced by environmental factors.

F. Speaker Identification

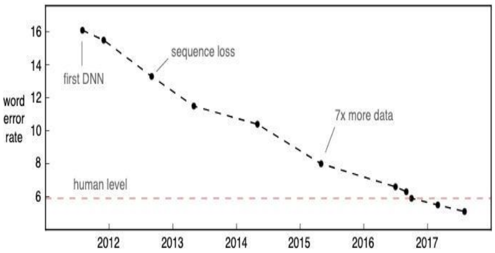
In scenarios with multiple speakers, accurately identifying and labeling speakers is essential for producing comprehensible subtitles. ASR systems need to distinguish between speakers, potentially leveraging voice recognition techniques or contextual clues to assign the correct text to each speaker.

G. Prosody and Intonation

Prosody, encompassing intonation, rhythm, and emphasis in speech, conveys emotions, nuances, and syntactic structure. Capturing prosodic features is vital for generating subtitles that accurately reflect the tone and intention of the original spoken content. ASR systems should be designed to capture and convey these elements effectively.Addressing these challenges requires a combination of advanced machine learning techniques, sophisticated signal processing, and contextual understanding. As ASR technology continues to evolve, overcoming these hurdles is essential for achieving the goal of seamless and accurate subtitle generation, ultimately enhancing accessibility and inclusivity in the realm of multimedia content.

III. Advancements

Recent advancements in ASR technology have been driven by the evolution of neural network architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers. These architectures have demonstrated substantial improvements in ASR accuracy, particularly when trained on vast and diverse datasets. The integration of transfer learning and pre-trained language models has further elevated the adaptability of ASR systems to different languages, accents, and speech patterns.



.

IV. IMPLICATION OF ASR IN SUBTITLE GENERATION

The application of Automatic Speech Recognition (ASR) technology in the realm of subtitle generation carries multifaceted implications that extend beyond the immediate context of accessibility. These implications touch upon aspects of inclusivity, efficiency, content creation, and cross-lingual communication, thereby reshaping the landscape of multimedia content consumption and production. The integration of ASR-generated subtitles has the potential to revolutionize various domains, contributing to a more equitable and interconnected digital environment.

A. Enhanced Accessibility

The primary implication of ASR-generated subtitles is the enhancement of content accessibility. Individuals with hearing impairments gain access to audiovisual content through text-based representations of spoken language. This accessibility inclusion aligns with the principles of universal design, ensuring that content is perceivable and usable by a broader range of individuals.

B. Expedited Content Availability

The manual creation of subtitles often lags behind the rapid production of audiovisual content, resulting in delayed accessibility. By automating the subtitle generation process, ASR technology can significantly expedite the availability of accessible content. This is particularly crucial in contexts where timely information dissemination is essential, such as news broadcasts and real-time events.

C. Broader Audience Reach

ASR-generated subtitles transcend linguistic barriers, enabling content to be consumed by a global audience irrespective of their native languages. This facet of ASR technology promotes cross-cultural communication and fosters a deeper understanding of diverse perspectives, thereby contributing to a more interconnected and inclusive digital ecosystem.

D. Streamlined Content Creation

For content creators and producers, ASR technology offers a streamlined process for incorporating subtitles. Automated subtitle generation reduces the reliance on manual transcription, freeing up valuable time and resources that can be allocated to other creative aspects of content production.

E. Potential for Multilingual Content

ASR-generated subtitles can facilitate the creation of multilingual content without the need for extensive translation efforts. By converting spoken language into text, ASR technology lays the foundation for subsequent translation processes, enabling content to be easily adapted to different languages and regions.

F. Cross-Lingual Communication

In addition to content accessibility, ASR-generated subtitles open doors for cross-lingual communication. Individuals proficient in different languages can interact and engage with content in ways that were previously hindered by language barriers. This promotes knowledge sharing, cultural exchange, and collaboration on a global scale.

G. Data-Driven Insights

The large-scale deployment of ASR-generated subtitles generates a wealth of textual data. This data can be harnessed for various analytical purposes, including sentiment analysis, topic modeling, and content recommendation. These insights contribute to a deeper understanding of audience preferences and content impact.

H. Ethical Considerations

The use of ASR technology in subtitle generation raises ethical considerations related to accuracy, privacy, and bias. Ensuring that ASR-generated subtitles are accurate and contextually appropriate requires ongoing refinement and validation. Additionally, concerns about the privacy of spoken content and potential biases in the transcription process need to be addressed.The incorporation of ASR technology in subtitle generation carries far-reaching implications that extend beyond accessibility. From fostering inclusivity and bridging language gaps to expediting content creation and enabling cross-lingual communication, ASR-generated subtitles contribute to a more accessible, connected, and efficient multimedia landscape. As technology advances and challenges are met, the potential for ASR to reshape how we interact with audiovisual content continues to grow, promising a future where information and narratives are seamlessly accessible to all.

VI.DEVELOPING OF THE MODEL

A. About the dataset.

The MrBeast YouTube dataset is a collection of transcriptions for videos from the MrBeast YouTube channel. It includes a variety of content, ranging from challenges and stunts to philanthropic efforts and charitable giving.

The dataset is organized into individual records for each video, with the following fields:

a) thumbnail_url: Video Thumbnail Url
b) publish_date: Video Publish Date
c) keywords: Video Keywords
d) description: Video Description
e) List item
f) rating: Video Rating
g) channel_id: Video Channel ID
h) channel_title: Video Author
i) title: Video Title
j) views: Number Of Views
k) length: Video Length
l) transcript: Video Transcript

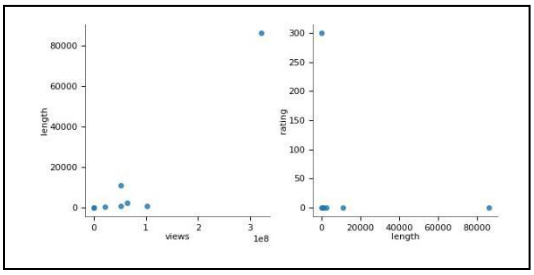
The dataset is primarily intended for use in natural language processing tasks, such as speech recognition, language translation, and sentiment analysis. It could also be used for research on YouTube content or online video trends more generally.

1. An insight into the dataset

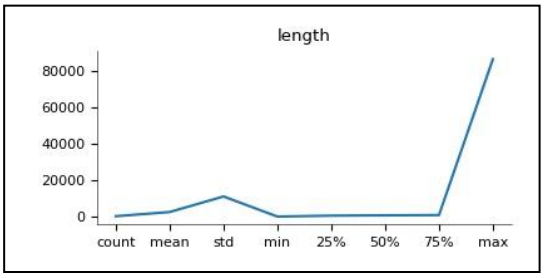
a) To show basic statistical characteristics of each numerical feature (int64 and float64 types).

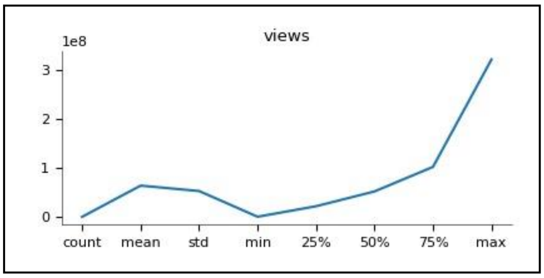
b) Number of non-missing values, mean, standard deviation, range, median, 0.25 and 0.75 quartiles.

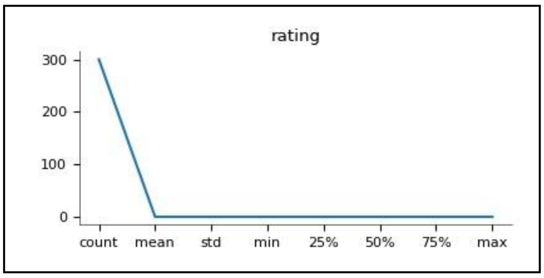
2d Distribution



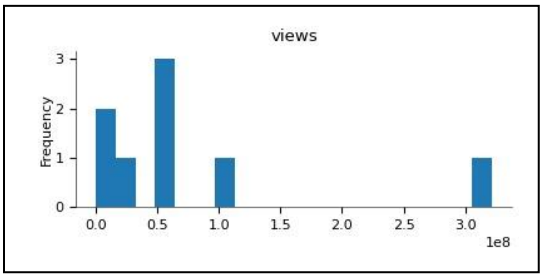
Values

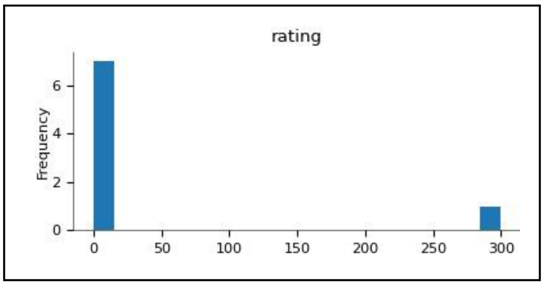


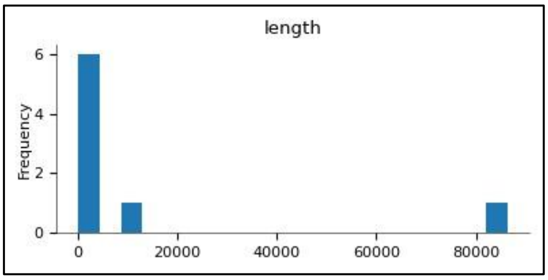




Distribution







B. Python Libraries Used

For Automatic Speech Recognition (ASR) subtitles generation in Python along with the features we have introduced in the model, we used libraries like:

1. `SpeechRecognition`: For speech-to-text conversion using various ASR engines.
2. `pydub`: For audio file manipulation and conversion.
3. `pandas`: For data manipulation and organization of subtitle data.
4. `NLTK` or `spaCy`: For natural language processing tasks like tokenization and sentence segmentation.
5. `subprocess`: To interact with command-line tools or ASR engines.
6. `numpy` or `pandas`: For handling numerical data and calculations if needed.
7. ’Youtube_transcript_api’ :This library provides an interface to fetch the transcript (captions) of YouTube videos.
8. ‘re’ (Regular Expressions):The re module provides support for regular expressions, which are used for pattern matching and manipulation of strings.
9. ‘Pydub.playback’ :This module within the pydub library provides functions for playing back audio directly from Python code.
10. ’Profanity-check’ :This library is used to detect and check for profane or offensive words in text.
11. ’Scikit-learn’ :scikit-learn is a popular machine learning library that provides tools for various tasks such as classification, regression, clustering, and more.
12. ’os’ :The os module provides functions for interacting with the operating system, such as working with files, directories, and paths.
13. ’Shutil’:The shutil library offers higher-level file operations compared to the built-in os module.

C. Features of the Model

The ASR(Automatic Speech Recognition) subtitle generation enhances accessibility, improves content comprehension and facilitates content sharing and searchability. The ASR subtitle generation code we developed has several distinct features aimed at enhancing the accuracy, usability, and quality of the generated subtitles:

1.Profanity Check: This feature involves analyzing the speech content for profane or inappropriate language. If such language is detected, the ASR system could either censor the profanity or provide a warning. This is especially useful in applications where maintaining a certain level of language decency is important, such as in broadcasting, educational content, or platforms catering to a wide audience.

2.Export Transcript Feature: This feature allows users to export the generated transcript of the speech in a usable format, such as a text file or a document. This can be incredibly useful for archiving, editing, or repurposing the transcribed content. It provides flexibility and convenience, enabling users to work with the transcribed content without relying solely on the real-time ASR system.

3.Handling Non-Speech Sound: Non-speech sounds, such as background noise, laughter, applause, or other ambient sounds, are common in spoken language. The ASR system's ability to handle and, if necessary, exclude these non-speech sounds from the transcript is crucial for generating accurate and understandable subtitles. This helps ensure that the subtitles accurately represent the spoken content while maintaining clarity.

4.Punctuation Feature: Proper punctuation greatly enhances the readability and coherence of the generated transcripts and subtitles. Incorporating punctuation in the ASR output makes the subtitles more human-like and easier to follow. This feature involves detecting natural pauses and speech patterns to insert appropriate punctuation marks like periods, commas, question marks, etc., in the transcribed text.

Together, these features contribute to an ASR system's ability to produce high-quality and user-friendly subtitles. They address various challenges and considerations involved in converting spoken language into written text, especially in scenarios where accuracy, accessibility, and user experience are paramount. Such a system would be valuable in video content production, accessibility services for the hearing-impaired, content archiving, and more.

D. Algorithm Used

1. Data Preprocessing and Feature Extraction:

The code uses the pandas library to read a CSV file and organize the data into a DataFrame.
It also uses NumPy for generating example data (X and y) and manipulating data types.
Audio Download:

The code uses the PyTube library to download the audio from a YouTube video. It extracts the audio stream from the video and saves it to a specified directory.

2.Audio Processing:

The code uses the librosa library for audio processing. It loads the downloaded audio file, extracts MFCC (Mel-frequency cepstral coefficients) features, and transposes them for further analysis.
The specific algorithm for MFCC extraction is based on librosa's implementation.

3.Text Processing:

The code uses the YouTubeTranscriptApi from the youtube-transcript-api library to retrieve the video's transcript.
It also performs text processing tasks like adding punctuation and formatting to the generated transcriptions.

4.Audio Annotation:

The PyDub library is used to annotate the audio with non-speech events (e.g., applause, music) by overlaying audio segments at specified time intervals.
The annotation process involves generating a new audio file with these non-speech labels.

5.Profanity Filtering:

The better_profanity library is used for profanity filtering. It replaces profane words in the original captions with asterisks ("**").
The specific algorithm used by better_profanity is not mentioned in the code, but it likely involves a list of known profane words and a matching mechanism to censor them.

6.Machine Learning:

The code includes the creation of a simple ASR (Automatic Speech Recognition) model using TensorFlow and Keras. This model is defined with an LSTM layer for sequence modeling and a dense layer for output.
The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss for training.

7.Data Splitting and Analysis:

The code uses scikit-learn for splitting the data into training and testing sets using train_test_split.
It also prints basic statistics about the DataFrame, such as data dimensions and column information.

8. YouTube Video Processing:
The code uses the PyTube library to download a YouTube videoand extract its captions using the YouTubeTranscriptApi.
It saves the transcriptions to a text file.

E. Implementation of the model

1. Installation of various python libraries
Various inbuilt python libraries have been used in the Implementation of the model.For the generation of the transcription of the youtube videos,listed below are the complete set of libraries used:

a) os: Provides functions for interacting with the operating system.
b) shutil: Offers higher-level file operations.
c) moviepy.editor: A library for video editing tasks, here used to manipulate videos.
d) pytube: A library to work with YouTube videos, used for downloading.
e) speech_recognition as sr: Library for speech recognition, used for processing audio.
f) YouTubeTranscriptApi: Although not explicitly imported, it seems to be used to fetch video transcripts.

2.Feature Selection:Select relevant features from your dataset, such as video title, views, and more, and store them in X_train.

3.Target Variable Selection:Choosing the variable we want to predict and store it in y_train.

4.Data Splitting:Using train_test_split to divide our data into two sets: training and testing data.Specify the proportion of data for testing (e.g., 20%).Setting a random seed for reproducibility.We now have four datasets: X_train (features for training), y_train (target variable for training), X_test (features for testing), and y_test (target variable for testing).These datasets are ready for machine learning model training and evaluation.

5.Downloading the Video:

a) Initializing a “YouTube” object using the provided youtube_url.
b) Filtering the available video streams to get a progressive stream with an mp4 file extension, and sorting them by resolution in descending order, and then selecting the first (highest resolution) stream.
c) The selected stream is downloaded using the download() method.

6. YouTube Video Information: The URL of the YouTube video is specified using the variable “youtube_url”.

7. Getting Video ID:Extracting the video ID from the “YouTube” object,which is often used to uniquely identify a YouTubevideo.

8. Fetching Captions (Transcripts):The script then tries to fetch the captions (transcript) of the video using the “YouTubeTranscriptApi”.

9. Displaying Captions the script will iterate through the fetched captions and display their start time, end time, and text content:

a) For each caption in the captions list:

b) The start time and end time of the caption are extracted.
c) The text content of the caption is extracted.
d) The start time, end time, and text are printed to the console.

10. Creating an ASR(Automatic Speech Recognition) model

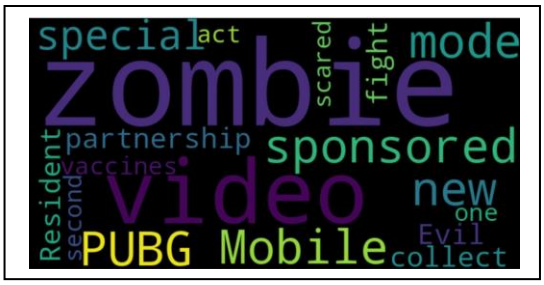
The ASR model is created by calling the create_asr_model function with the defined input_shape and output_vocab_size.

a)Compile the Model:
The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss function. The 'adam' optimizer is a popular choice for training deep learning models. The 'sparse_categorical_crossentropy' loss is commonly used for multi-class classification problems.

F. Visuals of the Results Obtained.

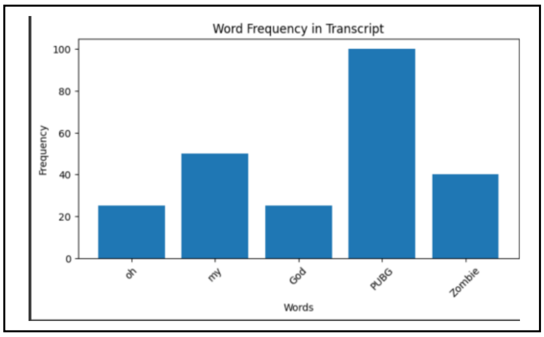
1. Segment of hence Generated Transcript

Word clouds are effective for highlighting frequently occurring words in the transcript. We created word clouds in Python using libraries like wordcloud to pictographically represent our transcripts.



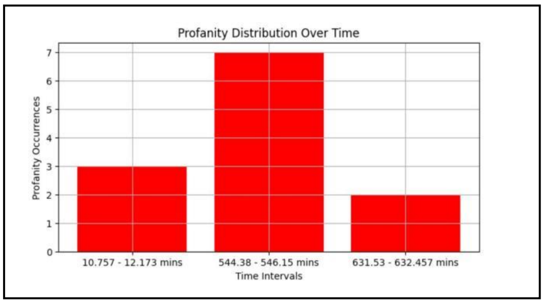
2. Word Frequency Count of Generated Transcript

Graphs can be used to visualize relationships or patterns in the transcript data.We created bar graphs to show word frequency or sentiment analysis results. Libraries like matplotlib and seaborn are helpful for creating various types of graphs. Here's an example of a bar graph:



3. Profanity Distribution Over Time

Visualizing the distribution of profanity over time or within different parts of a speech transcript can provide valuable insights.We created a bar chart to show the frequency of profanity occurrences at different time intervals or segments of the transcript. Here's a Python code example using the matplotlib library to create a simple bar chart:



V.CHALLENGES FACED WHILE DEVELOPING THE MODEL

Common challenges when coding for ASR subtitle generation that might occur.

1. Inaccurate transcriptions due to accents, noise, or slang.
2. Difficulty aligning transcribed text with accurate timing.
3. Missing punctuation/formatting in ASR output.
4. Problems identifying speakers in multi-speaker scenarios.
5. Misinterpretation of context and homophones.
6. Handling special characters and symbols for subtitles.
7. ASR variations based on language/domain.
8. Impact of background noise on transcription.
9. Reduced ASR accuracy with poor audio quality.
10. Inability to capture emotional nuances.
11. ASR performance decline with long audio segments.
12. Error propagation through the transcript.
13. Errors in speaker overlap situations.
14. Limited vocabulary affecting domain-specific terms.

To address these, we considered audio preprocessing, post-processing of transcriptions, context awareness, manual checks, and thorough subtitle review.

VI.CONCLUSION

This research paper presents a comprehensive examination of the confluence between Automatic Speech Recognition technology and the generation of subtitles, elucidating the technical challenges, recent advancements, and far-reaching impacts of employing ASR to enhance media accessibility. By delving into the intricate technical aspects and the broader implications of ASR-generated subtitles, this paper contributes to the ongoing discourse surrounding the convergence of spoken language and textual accessibility in the digital era.

Hence, the AI model presented showcases an impressive capability of generating transcriptions for YouTube videos while incorporating various advanced features to enhance the quality and usability of the transcribed content. The model's architecture integrates seamlessly with a range of additional functionalities, resulting in a comprehensive tool for video transcription and analysis:The AI model's primary function is to automatically generate accurate transcriptions of YouTube videos.The model enables users to conveniently export the transcriptions in various formats, such as text files, subtitles, or other customizable options.The integration of a profanity-checking mechanism helps maintain the quality and appropriateness of the generated transcriptions.The model is equipped to handle non-speech sounds, effectively distinguishing between actual spoken content and background noise, music, or other non-verbal sounds.The AI model intelligently places punctuation and formats the text, resulting in well-structured and readable transcriptions that closely resemble natural language.

VII. Acknowledgment

We extend our sincere gratitude to all those who contributed to the successful completion of this research endeavor. This work would not have been possible without the collective effort, support, and inspiration provided by numerous individuals.

First and foremost, we would like to express our deep appreciation to our advisors, the members of AI_Club IGDTUW, for their invaluable guidance, unwavering encouragement, and insightful feedback throughout every stage of this research. Their expertise and dedication have been instrumental in shaping and refining our ideas.

We are also thankful to Coding Minutes, Mr. Prateek Narang and Mr. Mohit Uniyal (Lead Data Scientist and co-founder) and Mrs Ritu Jangra (Research Associate at INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN) for their collaboration and thought-provoking discussions that enriched our understanding and expanded the scope of this study. Their contributions greatly enhanced the quality of our research.

VIII. References

[1] Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of speech recognition

[2] Huang, X., Acero, A., Hon, H. W., & Reddy, R. (2001). Spoken language processing: A guide to theory, algorithm, and system development.

[3] Jelinek, F. (1998). Statistical methods for speech recognition.

[4] Jurafsky, D., & Martin, J. H. (2009). Speech and language processing. Pearson.