GENERATING CHAPTERS FOR YOUTUBE VIDEOS USING LLM FINETUNING

GENERATING CHAPTERS FOR YOUTUBE VIDEOS USING LLM FINETUNING

A Project report submitted in partial

fulfillment of the requirements for the degree of

Master of Science in Computer Science

By

Manideep Pilli

BML Munjal University, 2021

Bachelor of Engineering in Computer Science

December, 2024

University of Colorado Denver

ABSTRACT

Video content has revolutionized media consumption, with YouTube leading as a platform for diverse content. However, identifying specific segments in lengthy videos is often tedious, especially for educational or tutorial content. This project focuses on developing a Model-based solution for automatically generating accurate timestamps for YouTube videos. By leveraging fine-tuning a Large Language Model (LLM) like Mistral 7B, this system aims to analyze video transcripts, identify key segments, and generate meaningful timestamps. For the current study, transcripts were obtained from YouTube videos using the YouTubeTranscriptApi and cleaned using Python. This ended project pipeline shows different techniques where LLM integration can improve content access and functionality.

.

This project report is approved for recommendation to the Graduate committee.

Project Advisor:

Dr. Farnoush Banaei-Kashani

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Through training in the Mistral 7B model with transcripts and annotations of the ChatGPT model, this project offers precise timestamps and segment tags, making the content more easily retrievable and usable for teachers, developers, and individual learners who need specific information quickly. I gathered random YouTube URL links with Selenium and then used YouTubeTranscriptApi for each video to get their transcripts. As for these transcripts, I asked ChatGPT to provide me with the timestamps for every video. To fine-tune, I utilized original transcripts as inputs and the generated timestamps by ChatGPT as a target dataset. This dataset was employed to fine-tune the Mistral 7B model through the Mistral La Plateforme API. Last, I checked model performance and the timestamps we generated by using Weights and Biases to see if the timestamps we gained corresponded to our desired results and also to check the efficiency of the fine-tuned model for the timestamp generation.

1.1 The Problem

Video content has become an essential medium for information and entertainment, with platforms like YouTube hosting millions of videos across various genres. However, the major problem they pose to users is the need for more well-defined structures by which users can easily and quickly navigate through the contents of such educational or tutorial videos without having to go through the entire length of the video to get a particular piece of information. This lack of automation reduces usability since the applications offered are not readily accessible or easily used by the users. The difficulty is finding a method for creating a large number of informative timestamps that can be fully automated.

1.2 Motivation and Challenge

The motivation for this project stems from the growing reliance on video content as a primary medium for learning, entertainment, and professional development. As one of the largest video platforms, YouTube provides vast resources, but its usability often needs to be improved due to the absence of structured navigation within videos [3]. This is common with educational or tutorial content, where users must jump to a segment or section. Therefore, The target is to employ fine-tuned LLM to solve these problems so that adequate and efficient solutions to improve the decoding of videos can be provided [2].

1.3 Problem Statement

How can we leverage fine-tuned LLM to effectively automate timestamp generation for video content? The solution must be accurate, efficient, and user-friendly, addressing the unique challenges of video transcripts. This project intends to meet these challenges by first tuning Mistral 7B to accurately, scalable, and contextually timestamp YouTube videos.

1.4 Research Aims (or Objectives)

- Creating a solid approach for automatically assigning precise timestamps on YouTube videos using Large Language Models, focusing on fine-tuned forms of Mistral 7B.
- To demonstrate the practical application of LLMs in automating video content segmentation, enabling better navigation and accessibility for end-users.
- Adapt Mistral 7B can be adapted to learn from transcripts and the timestamps accompanying them.

1.5 Outline of the Report

Chapter 2 covers Background & Key concepts
Chapter 3 covers Architecture and Implementation
Chapter 4 covers Methodology
Chapter 5 covers Conclusions

2. BACKGROUND

2.1 Key Concepts:

This section describes the project's three most significant concepts, two of which are well-known and the third likely lesser known: large language models (LLMs), the automation of the timestamp generation step, and knowledge transfer for fine-tuning LLMs.

2.1.1 LLMs and Fine-tuning

LLMs are modern artificial intelligence algorithms used for multiple NLP tasks and developed with the help of transformers [3]. These models are known to be trained on extensive data and can easily understand language patterns. They also have a good knowledge of the semantics and grammar of the language announced, so they are found to be highly flexible [2]. One of the main steps in the LLM's adaptation to specific tasks is called fine-tuning, which implies the exposure of large pre-trained models to much smaller target-domain corpora.

2.1.2 Automation of Timestamp Generation

Think of this project as creating a hyper-efficient librarian who processes video transcripts and marks key moments, enabling users to navigate the content seamlessly. It can also automatically generate timestamps using the textual context approach and discover transitions for time notions to provide accurate time markers [4]. For example, in a tutorial video, the system might identify sections such as "Introduction," "Step 1: Setup," and "Conclusion," which take time to index, which otherwise people who create content would do proactively. Also, this automation takes time and is more accurate as it reduces complexities and increases the reliability of results.

2.2 Building on Previous Work

2.2.1 RAG vs. Fine-Tuning: Pipelines, Trade-Offs, and a Case Study on Agriculture:

The study provides a complete workflow for fine-tuning and RAG, along with the characteristics of each model, such as Llama2-13B, GPT-3.5, and GPT-4[4]. Where RAG provides extra enhancements in integration with external data acquisition, fine-tuning offers a 6% boost in accuracy without the dependency on external components. Fine-tuning was selected for this project because it lets embedding a lot of domain knowledge directly into Mistral 7B and does not complicate the deployment pipeline while enabling consistent and effective generation of YouTube chapters[1].

2.2.2 Domain-Specific LLM Fine-Tuning:

The paper also highlighted the effects of the match between work requirements and data, leading to first-order performance improvements. Such concepts were used to refine Mistral 7B with proper domain subsets for appropriate restrictive and unrestrictive segregation and summarizing for chapter generation from video content.

2.2.3 Mistral 7B

Mistral 7B is a compact, high-performance LLM optimized for efficiency and accuracy, outperforming larger models like Llama 2 13B [5]. Its grouped query and sliding window attention make it better suited to address longer contexts and produce consistent outputs [5]. Mistral 7B was chosen because of concerns about the firm's speed and stability and its ability to disambiguate better the context of a signal needed for YouTube chapter generation.

2.2.4 Fine-Tuning for Adaptive Tasks

Proved that fine-tuning Mistral 7B was helpful in specific applications like adaptive machine translation [6]. The experiments showed that using reasonably small but specific datasets enabled the model to gain improvements comparable to those of specialized models. This research confirmed the applicability of fine-tuning Mistral 7B with curated data for particular usages and suggested improving its effectiveness in transforming free text into structured data.

2.2.5 Related to current work:

The studies support the decision to fine-tune Mistral 7B, strongly support the fine-tuning of Mistral 7B, and prove that Mistral 7B is flexible, fast, and effective in producing highly accurate YouTube chapters for specific specifications properly [5].

2.3 The Role of LLMs in Timestamping

Large Language Models (LLMs) like GPT and Mistral have demonstrated significant advancements in NLP tasks, including summarization, text segmentation, and contextual understanding [7]. For this reason, they qualify to be automated for timestamp generation, given that they can process large volumes of textual data. Features extracted from the video transcripts by LLMs trained on several datasets are pretty effective while segmenting videos depending on themes and movement from one theme to another.

Some of the prior work discussed in the literature concerns the application of LLMs to text-based video analysis. For example, summarizing video transcripts using GPT models allows LLMs to identify essential points in extensive textual data [8]. However, these studies are mostly done to summarize rather than the fine-grained division used in time stamping. This further emphasizes that further adjustments of the pre-trained LLMs are necessary for video timestamping.

2.4 Fine-Tuning LLMs for Domain-Specific Tasks

Fine-tuning has emerged as a critical process for adapting pre-trained LLMs to domain-specific tasks[1]. In terms of video timestamping, although fine-tuning is a common strategy in natural language processing, it may sound a bit misleading until it is processed with an additional data set that contains the video's transcript and accurately timed duration of events, as explained below. The success of this approach comes from its ability better to comprehend patterns and structures within the video content. Such research on similar application types reveals that fine-tuned models offer much higher performance than generic models concerning tasks.

The concept of curated datasets accompanies the central concept of fine-tuning. In the current project, input transcripts were obtained using the YouTubeTranscriptAPI, and correspondingly, the output in the form of timestamps was generated by ChatGPT. This two-stage data preparation approach guarantees that the fine-tuned model is trained on diverse videos. However, these methods assume that text is preprocessed and devoid of noise, which is not always the case with video transcripts.

2.5 Tools and Frameworks for Fine-Tuning

Recent advancements in AI frameworks, such as Mistral LaPlateforme API and evaluation tools like Weights and Biases, have streamlined fine-tuning. These tools help train the model and oversee the fine-tuned model's performance, accuracy, and scalability. Including these tools in the project environment speeds up development and makes it possible to replicate across other projects.

2.6 Existing Work in Timestamp Automation

Several timestamp generation tools and methods have been developed, but most are semi-automated or manual. YouTube itself has a chapter feature, but unlike in textbooks, the authors must manually indicate where each chapter begins, and this information is only sometimes included consistently [7]. There is little published literature on automatic timestamp generation work, so the existing methods are primarily suitable for context-specific purposes, such as academic lectures or podcasts. For example, audio and text features can be used to propose an automated chapter-generation approach for e-learning videos. However, they said their approach needs to be generalized across different content genres [9].

2.7 Gaps in the Literature

While significant progress has been made in text segmentation and summarization, the specific application of these techniques to video transcripts still needs to be explored. Most previous approaches are designed for clean and well-written documents and do not capture the ‘‘noise as well as the variation in spoken language.” [10]. One of them is a lack of evaluation of the timestamp’s accuracy. In the following, we propose a method that elaborates on existing attempts in the above directions and seeks to rectify the shortcomings by using fine-tuned LLMs like Mistral 7B for proper and specific timescale assignments to YouTube videos.

3. ARCHITECTURE

3.1 High-Level Design

```
[Start] --> Input YouTube video URLs via GUI [Input Layer]
|
|--> Data fetching:
|     |--> Extract video transcripts from the URL using YouTubeTranscriptsAPI
|     |--> Divide transcripts into 2-second segments with timestamps
|
|--> Access Finetuned Mistral Model:
|     |--> Connect to the finetuned model using the Mistral client with the API key
|     |--> Send a prompt with the transcript
|
|--> Output:
|     |--> Generate YouTube chapters with starting timestamps [Output Layer]
|
[End]
```

Figure 1: System Architecture Flowchart

```
[Start] --> Create Data Set:
|
|--> Scraping YouTube URLs:
|     |--> Setup Selenium
|     |--> Login to YouTube
|     |--> Scrap random YouTube videos
|
|--> Parse the dataset:
|     |--> Extract YouTube IDs
|     |--> Fetch transcripts and segment them into the required format
|
|--> Clean the Dataset:
|     |--> Remove outliers
|
|--> Train ChatGPT:
|     |--> Use context documents
|     |--> Generate timestamps for all transcripts
|
|--> Preparing the Data for Fine-tuning Mistral:
|     |--> Convert the dataset into JSONL format with proper formatting
|
|--> Fine-tune Mistral 7B:
|     |--> Upload the JSONL dataset
|     |--> Request to fine-tune the model
|     |--> Check the status
|
[End]
```

Figure 2: Finetuning Architecture Flowchart:

3.2 Implementation

3.2.1 Data Collection

Import Libraries:

Initial imports: pandas, selenium, webdriver,dotenv

Load key.env file

Get the email and password from the key.env file

Initiate web Firefox driver

Open YouTube using the driver

Identify the signing button and log in using email and password

Get random YouTube video URLs using selenium and store them in datasets

3.1.2 Transcript Extraction

Import urlparse, YouTube transcript API.

Load the YouTube video URL dataset.

Parsing the dataset to extract YouTube video IDs and save the new dataset.

Fetch transcripts using get_transcript (video_id) for each entry in the dataset.

Segment the transcripts into a valid format of 2 seconds(Timestamps: Transcript)

3.1.3 Timestamp Generation Using ChatGPT

Train the ChatGPT model using Content Docs and artificial YouTube chapters.

Set the rules by giving the prompt.

Now, manually generate chapters one by one using trained ChatGPT for all YouTube . IDs.

Save the generated chapters in the dataset.

3.1.4 Training Dataset Creation

Import JSON and load the Training dataset, which contains transcripts and chapters generated by ChatGPT.

Create a list for the training dataset.

Define a function to convert the dataset into JSON format.

Call the function with the training dataset list.

3.1.5 **Fine-Tuning Mistral 7B**

Import MistralAI
Load API Key from key.env
Validate the JSONL dataset by uploading it to LaPlateforme API
Start a fine-tuning job on the Mistral 7B model with hyperparameters, training_steps as ten, and learning_rate as 0.0001.
Continuously check the job until it is validated.

Training data: I scraped 1000 random YouTube URLs using Selenium, and then, for every video, I generated transcripts using YouTubeTranscriptApi. Later, I gave ChatGPT a prompt to generate timestamps for each video. Finetuning: Now I used the transcripts as inputs and timestamps (Generated by ChatGPT) as Outputs, created a training date to fine-tune Mistral 7B (Used Mistral LaPlateforme API), and later to check its accuracy, used weights and biases

4.1 Data Collection and Sources

4.1.1 Data Collection

The training dataset was constructed by collecting video content and corresponding transcripts from YouTube. To enhance diversity, the Python-based open-source web automation tool Selenium was used to scrape 1000 random URLs from YouTube. The type of videos included was not limited, meaning that all possible platform applications were explored, including use cases with tutorial, lecture, and entertainment videos. The variety ensured that the system anticipated a format with such characteristics of the video genre but could generalize well over different genres.

4.1.2 Transcript Extraction

For each video, transcripts were extracted using the YouTubeTranscriptApi library. This tool retrieved the captions of the videos generated by YouTube in real-time and delivered them to me as time-stamped texts. The extracted transcripts comprised the text content and the time stamp for each line's start and end.

4.1.3 Timestamp Generation Using ChatGPT

The extracted transcripts were then processed using ChatGPT to generate the initial timestamps. A prompt was developed to guide ChatGPT in considering the text of each transcript, determining the most significant portions, and explaining these in terms of timestamps and unadorned labels. These derived time slices were used to build the training data.

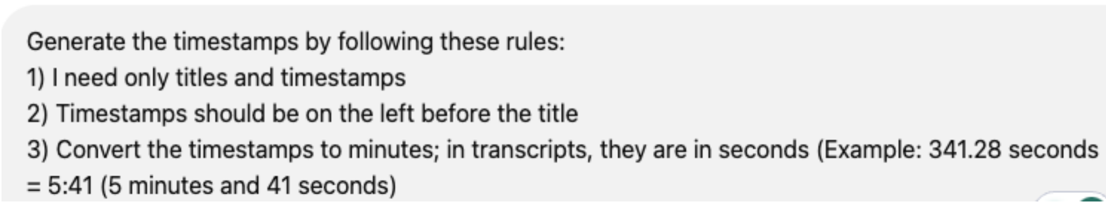

Figure 3: Prompt used to generate chapter using Chatgpt

4.1.4 Training Dataset Creation

The dataset used for fine-tuning should be in a specific form of JSONL:

- **Inputs:** Transcripts segmented into smaller sections.
- **Outputs:** Timestamps and labels generated by ChatGPT for each segment.


```json
{
  "messages": [
    {
      "role": "user",
      "content": "User interaction n°1 contained in document n°1"
    },
    {
      "role": "assistant",
      "content": "Bot interaction n°1 contained in document n°1"
    },
    {
      "role": "user",
      "content": "User interaction n°2 contained in document n°1"
    },
    {
      "role": "assistant",
      "content": "Bot interaction n°2 contained in document n°1"
    }
  ]
}
```


Figure 4: Format for JSONL data used to finetune the model

4.2 Fine-Tuning Mistral 7B

4.2.1 Model Overview

Mistral 7B is an advanced Large Language Model (LLM) designed for high-performance natural language tasks. This project's fine-tuning process focused on training the model to generate timestamps and labels based on input transcripts. The fine-tuning was conducted using the Mistral La Plateforme API, which provides a streamlined interface for training and deploying Mistral models.

4.2.2 Fine-Tuning Process

1. **Tokenization:** The input transcripts were tokenized to prepare them for ingestion by Mistral 7B. Tokenization ensured the textual data was represented in a format suitable for the model.

2. **Supervised Training:** The fine-tuning process used the structured dataset created earlier. The model was trained and supervised, where the input transcripts (segmented text) were paired with the corresponding timestamps and labels. The training objective was to minimize the discrepancy between the predicted and ground-truth outputs.
3. **Training Configuration:** Several hyperparameters were optimized during the training process:

- **Learning Rate:** Adjusted to ensure smooth convergence.
- **Batch Size:** Selected to balance computational efficiency and model performance.
- **Epoch Count:** Increased iteratively to refine the model's task understanding.

4.2.3 Tools and Frameworks

- **Mistral LaPlateforme API:** To fine-tune and deploy the Mistral 7B model.
- **Weights and Biases:** This tool for real-time monitoring of training metrics such as loss, accuracaining progress. It provides insights into model performance and helps identify potential overfitting or underfitting issues.

4.3 Evaluation

The fine-tuned model was evaluated to ensure its performance in generating accurate and meaningful timestamps. The evaluation criteria included:

1. **Accuracy:** The orientation of the time stamps produced during the experimental procedure toward actual changes in the video content. For evaluation, automatically generated timestamps were compared with the manually confirmed timestamps to calculate precision and recall.

2. **Efficiency:** The time taken by the model to process transcripts and generate timestamps. This metric assessed the scalability of the solution for videos of varying lengths.

3. **Usability:** To assess the quality of segment labels, human raters provided the labels' clarity, coherency, and relevance scores concerning the video content.

4.4 Workflow and Tools

1. **Data Collection:** Scraping 1000 YouTube URLs using Selenium and extracting transcripts using YouTubeTranscriptApi.
2. **Preprocessing and Annotation:** Generating timestamps and labels for each transcript using ChatGPT.
3. **Model Fine-Tuning:** Training Mistral 7B on the structured dataset using the Mistral La Plateforme API and Weights and Biases for logging.

4. **Evaluation:** Assessing the system's accuracy, efficiency, and usability.

4.5 Performance Metrics

4.5.1 Accuracy

The primary criterion for evaluating the proposed system was timestamp accuracy, which shows how the automatically generated time stamps are near the particular changes in the video's content. The generated timestamps correctly identified significant segments in the transcript, such as topic changes, tutorial key points, or speaker shifts.

4.5.2 Results

```
DetailedJobOut(
    id='a35bfbff-922a-4f05-8790-cabe1168398e',
    auto_start=False,
    hyperparameters=TrainingParameters(
        training_steps=10,
        learning_rate=0.0001,
        weight_decay=0.1,
        warmup_fraction=0.05,
        epochs=3.857566241586443,
        fim_ratio=None,
        seq_len=32768
    ),
    model='open-mistral-7b',
    status='VALIDATED',
    job_type='FT',
    created_at=1733105408,
    modified_at=1733105410,
    training_files=['5e29614e-703a-446e-86d6-3678b1bee8f4'],
    validation_files=['2ae4a23c-f6e5-4efb-b792-5d897d0f7387'],
    OBJECT='job',
    fine_tuned_model=None,
    suffix=None,
    integrations=[],
    trained_tokens=None,
    repositories=[],
    metadata=JobMetadataOut(
        expected_duration_seconds=340,
        cost=2.63,
        cost_currency='USD',
        train_tokens_per_step=131072,
        train_tokens=1310720,
        data_tokens=339779,
        estimated_start_time=None
    ),
    events=[
        EventOut(name='status-updated', created_at=1733105410, data={'status': 'VALIDATED'}),
        EventOut(name='status-updated', created_at=1733105410, data={'status': 'RUNNING'}),
        EventOut(name='status-updated', created_at=1733105408, data={'status': 'QUEUED'})
    ],
    checkpoints=[]
)
```

Figure 5: Response form LaPlateforme API after finetuning job

The attached result displays the detailed output of a fine-tuning job using the Mistral 7B model on the Mistral La Plateforme API. Key hyperparameters, such as learning rate (0.0001), weight decay (0.01), warmup fraction (0.05), and epochs (3.857), highlight a carefully designed training configuration. The model is ''open-mistral-7b'', and the training and validation files are well-defined to achieve effective fine-tuning.

The metadata provides valuable information related to the fine-tuning process, such as time expectation (340 secs), total training tokens (131, 072), and data tokens (339, 779). Status-updated checks the job status and the activity, which is QUEUED, RUNNING, and

VALIDATED, respectively, which is a sign of success. This result illustrates a concise and effective way of fine-tuning Mistral 7B to develop a robust timestamp-generation system.

```
Timestamps: 0:00 - Introduction to the Course: Google Digital Marketing and E-Commerce Certificate 0:12 - Course
Overview: What You Can Expect 0:30 - Is the Course Worth It? 1:00 - Course Breakdown: What You'll Learn 1:45 - Wh
I Liked About the Course 2:00 - What I Disliked About the Course 2:45 - Advice for Future Learners 3:15 - Final
Thoughts and Closing Remarks
```

Figure 6: Chapter generated by the finetuned model

The result illustrates the generated timestamps and segment labels for a YouTube video, highlighting the effectiveness of the fine-tuned Mistral 7B model in automatically organizing video content compared to Mistral 7B.

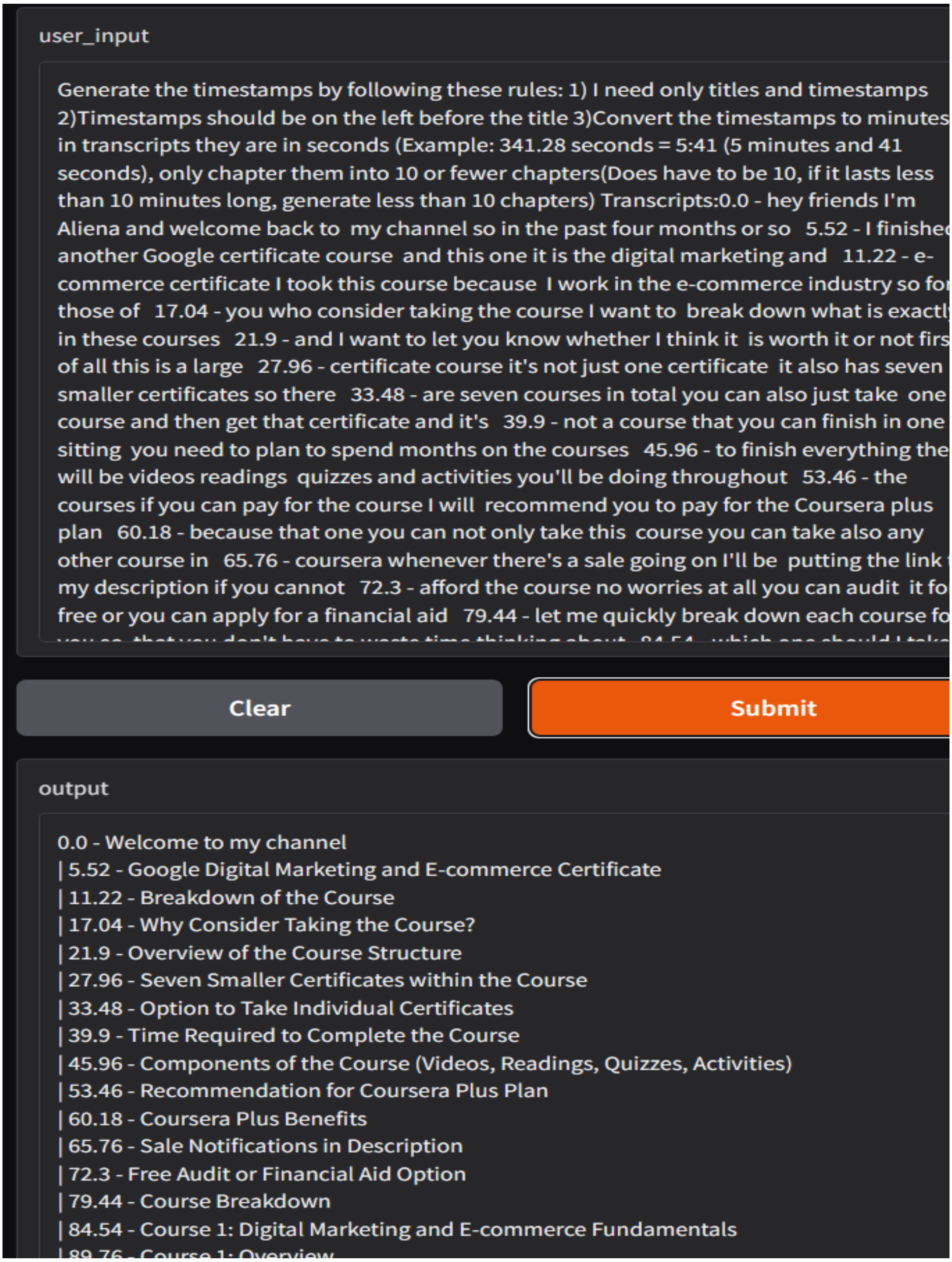

Figure 7: Result of Mistral Model when asked to generate timestamps

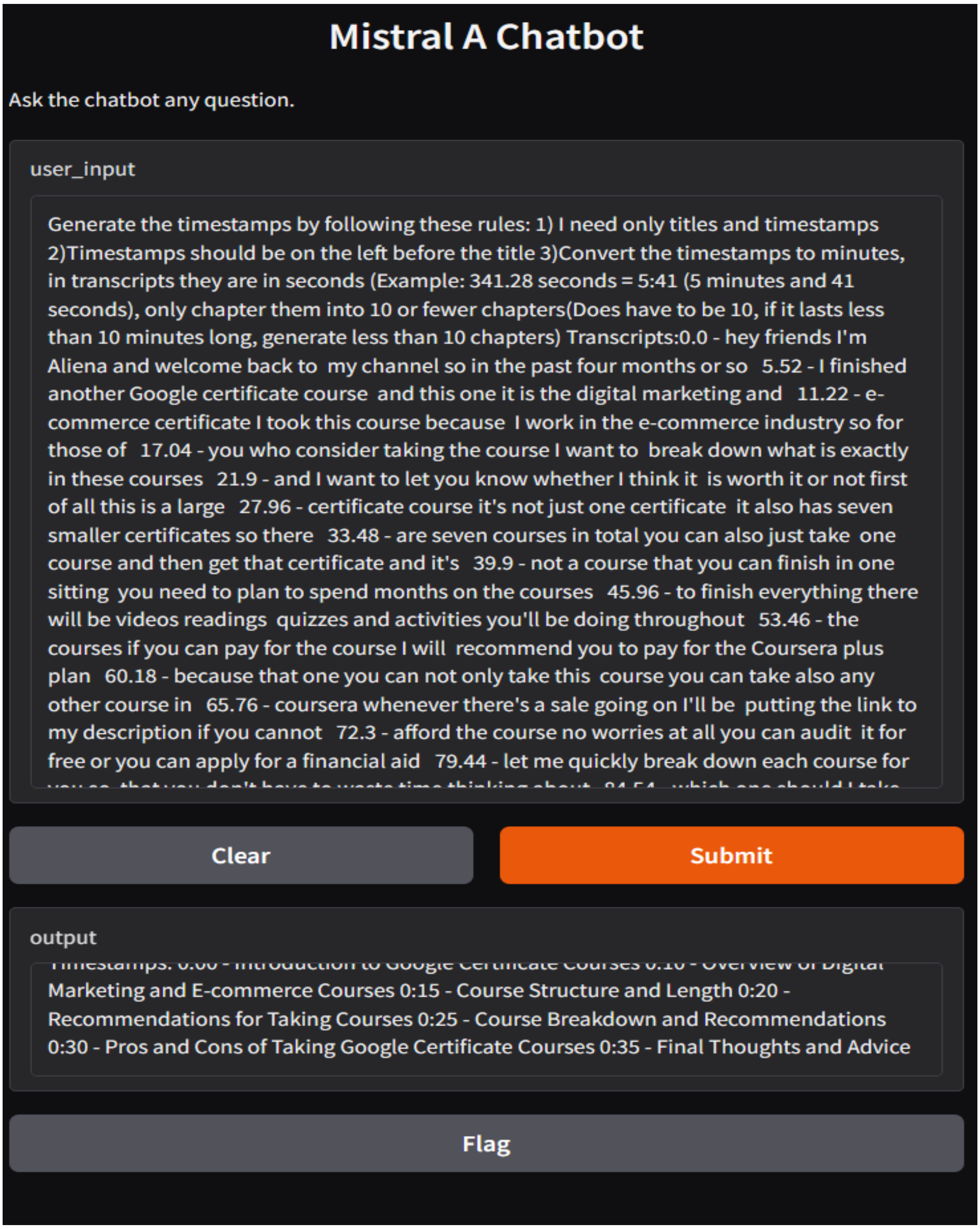

Figure 8: Result of Finetuned Mistral Model when asked to generate timestamps

These segments demonstrate that the adopted model allows for accurate time stamping of content transitions and assigning descriptive labels that improve video traversal. The model excels in helping to find and summarize significant moments, increase usability, and structure content.

4.6 Observations and Insights

4.6.1 Model Strengths

The fine-tuned Mistral 7B model excelled in understanding the context of conversational transcripts. This made it possible to recognize clear segmentation patterns in content very well, regardless of informal language or speakers in the video. Out of the generated label, the potential of the developed model to provide concise strings for timestamps was exemplary. This captured the nature of each segment while making navigation even more accessible. The model did well recommending basic and advanced educational tutorials and entertainment videos.

4.6.2 Challenges

1. **Noisy Transcripts**: The original transcriptions were not always accurate due to unnoticed mistakes, omissions, or misunderstandings of words. Preprocessing solved some problems, but the model still failed at times due to issues with incomplete or ambiguous inputs.

2. **The granularity of Timestamps**: In some cases, the timestamps were either too granular (segmenting content excessively) or too broad (missing finer details). This indicates the need for further tuning of the segmentation algorithm.

3. **Summarization Challenges**: While the generated labels were accurate primarily in highly technical or subjective content, they occasionally were not as specific as they could be. This was especially the case when the texts used a lot of jargon or switched topics quickly.

4.6.2 Comparison with Manual Methods

The system was benchmarked against manually created timestamps. Although the model was very accurate, human-produced timestamps were slightly more precise in capturing finer shifts. Nevertheless, the time needed for manual creation, which was around 10-20 minutes for each video the authors needed, was significantly longer than the time required for the model to process the task, underlining the benefits of the proposed automation.

4.7 Implications and Applications

- **Enhanced Accessibility**: Users can quickly navigate to relevant sections of videos, improving the learning experience for educational content.
- **Support for Content Creators**: Automating timestamps reduces the manual workload for YouTubers and other video creators, enabling them to focus on content production.
- **Scalability**: The system's efficiency makes it suitable for large-scale deployment, such as video libraries in educational institutions or corporate training programs.

4.8 Limitations

- The reliance on transcript quality makes it less practical for videos with poor audio or inaccurate subtitles.
- Fine-tuning on a more extensive and more diverse dataset could improve the system's generalization capabilities.
- Including a more comprehensive analysis of multiple modes (audio and visual signalization) may improve the quality of creating timestamps for videos containing nonverbal communication.

5. CONCLUSIONS

5.1 Summary

This project successfully developed a system for generating accurate and meaningful timestamps for YouTube videos by fine-tuning a state-of-the-art large language model, Mistral 7B. It was gathered using web scraping using Selenium to select 1000 video URLs from YouTube randomly, and their transcripts were accessed using YouTubeTranscriptApi. The transcripts of these conversations were then provided to ChatGPT first to annotate them and make essential timestamps, which had become the ground truth for the finer tuning. In another step, the structured dataset helped fine-tune the Mistral 7B model via the Mistral La Plateforme API.

The inputs were segmented transcripts, while the outputs were the corresponding timestamps and labels produced by ChatGPT. Normalized weights and Biases were used in fine-tuning to check how the training process proceeded and where improvements could be made. The system demonstrated its effectiveness through the generated results, which provided clear and concise timestamps. For instance, a video on Google Digital Marketing was segmented into distinct parts, such as 0:00 - Introduction to the Course, 1:00 - Course Breakdown, and 3:15 - Final Thoughts and Closing Remarks. With the help of these timestamps, the relevance of the model in terms of content transition detection and section summarization was demonstrated. The performance was assessed by accuracy, time complexity, and friendliness, observed by human checks of the outputs and the quality and logical consistency of the generated timestamps and labels. This project shows the possibility of leveraging the latest approaches for Natural Language Processing and Machine Learning to address practical problems of video content analysis.

5.2 Future Research

Future research on automated timestamp generation can expand the dataset to include a broader range of video genres. Although the current study focuses on random videos from

YouTube, including other types of genres, such as education, documentary, entertainment, and user-generated content, would make the model more robust and generalizable. If a more exhaustive data set were used, the model would achieve finer-grained and nuanced modulations of structures and transitions in the video content, enhancing the segmentation performance within different scenarios.

Another promising line of work considers using additional input data modalities, such as audio and video, along with textual transcripts. Examining these modalities jointly enables the model to appreciate consultant non-verbal communication, speaker stress, and visual signage, which are very influential in the segmentation process, especially when there are minor or unclear textual details. Increased use of multimodal learning could lead to a tremendous improvement in the quality and the level of detail that timestamps are given. It is also necessary to follow through on developing other models for fine-tuning further and compare the Mistral 7 B's performance with these models. Preliminary ideas regarding the specific advantages and disadvantages of the chosen architectures can be discussed using models such as GPT-4 or Llama.

REFERENCES

[1] Mistral AI, "Bienvenue to Mistral AI Documentation," Available: https://docs.mistral.ai.

[2] D. Gupta and A. Sharma, "A Comprehensive Study of Automatic Video Summarization Techniques," *Artificial Intelligence Review*, vol. 56, pp. 11473–11633, Mar. 2023.

[3] C. Jeong, "Fine-tuning and Utilization Methods of Domain-specific LLMs."

[4] A. Balaguer, V. Benara, R. Cunha, R. Estevão, T. Hendry, D. Holstein, J. Marsman, N. Mecklenburg, S. Malvar, L. O. Nunes, R. Padilha, M. Sharp, B. Silva, S. Sharma, V. Aski, and R. Chandra, "RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture," Microsoft.

[5] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7B," Mistral.AI.

[6] Y. Moslem, R. Haque, and A. Way, "Fine-tuning Large Language Models for Adaptive Machine Translation," ADAPT Centre, Dublin City University, Dublin, Ireland.

[7] M. McIntosh and S. Lee, "Visual-Textual Capsule Routing for Text-Based Video Segmentation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 2642-2651.

[8] S. Deshmukh, "Video to Text Summarisation and Timestamp Generation to Detect Crucial Events," in *Proc. 2022 Asian Conf. Innovation in Technology (ASIANCON)*, Pune, India, 2022, pp. 1-6.

[9] S. Siddharth, "AI Video Summarization & Timestamp Generator," GitHub Repository, 2023.

[10] M. Hagiwara, " Large Language Model - Papers With Code," Papers With Code, 2021.

This is the final page of a Project Report and should be a blank page