**Automated Video Chaptering Using Python**

**Project Overview**

This project demonstrates how to **automatically generate video chapters** from transcripts using **Natural Language Processing (NLP)** and **Machine Learning** techniques. By analyzing the transcript of a video, the code identifies distinct topics, groups them into logical sections, and assigns meaningful chapter names based on the content. The result is an automated chapterization system that can be used to organize and structure video content for better accessibility and navigation.

**Key Features:**

* **Transcript Analysis**: The video transcript is analyzed to detect topics using **NMF (Non-negative Matrix Factorization)** for topic modeling.
* **Topic-Based Segmentation**: The transcript is segmented based on topic shifts, which helps in identifying logical breaks in the content.
* **Chapter Naming**: Chapters are given context-based names generated through TF-IDF or other feature extraction methods.
* **Visualizations**: Displays histograms and bar charts to analyze text lengths, most common words, and topic distributions.

**Purpose**

The goal of this project is to provide an automatic solution for chaptering video content. This can be particularly useful for educational videos, webinars, interviews, or any other video content where structured navigation is needed. The approach used can help in:

* Automatically dividing content into relevant sections
* Enhancing user experience by allowing users to navigate to specific topics easily
* Providing insights into the topic distribution within the video

**Installation**

To get started with this project, follow these installation instructions:

1. **Clone the Repository**:

bash

Copy code

git clone https://github.com/your-username/automated-video-chaptering.git

cd automated-video-chaptering

1. **Install Required Dependencies**:

This project requires the following libraries:

* pandas
* numpy
* matplotlib
* sklearn
* nltk
* textblob

You can install the required dependencies using pip:

bash

Copy code

pip install -r requirements.txt

**Usage**

1. **Prepare the Transcript File**: Ensure that the video transcript is in CSV format, with at least two columns: start (start time in seconds) and text (the speech transcript).
2. **Run the Chaptering Script**:
   * Load the transcript using pandas
   * Preprocess and clean the text
   * Apply **NMF (Non-negative Matrix Factorization)** for topic modeling to identify topics in the transcript
   * Segment the transcript into chapters based on topic shifts
   * Generate chapter names using TF-IDF or a more advanced NLP method
3. **Visualize the Results**: Plot histograms and bar charts to explore text lengths, the most common words, and topic distributions.
4. **Output**: The script outputs:
   * A list of chapters with timestamps and their corresponding names.
   * Visualizations to explore various text statistics.

**Example Output**

**Final Chapter Points with Names:**

makefile

Copy code

00:02:34 - Chapter 1: Introduction to Data Science

00:12:45 - Chapter 2: Data Preprocessing and Cleaning

00:25:17 - Chapter 3: Feature Engineering Techniques

00:40:30 - Chapter 4: Machine Learning Models and Algorithms

...

**Visualizations:**

* **Topic Distribution**: A plot showing the frequency of each topic across the video.
* **Most Common Words**: A bar chart visualizing the most frequent words in the transcript (excluding stopwords).

**Code Walkthrough**

1. **Transcript Loading**: The transcript is loaded into a pandas DataFrame for easy manipulation. Any invalid time values are coerced into numeric types.
2. **Text Preprocessing**: Text length is computed, and the most common words are visualized using a **CountVectorizer**.
3. **Topic Modeling**: **NMF** is used to extract topics from the transcript, and the dominant topic for each segment is determined.
4. **Chapter Generation**: Using topic distribution, logical breaks are identified, and chapters are generated by merging consecutive breaks with the same dominant topic.
5. **Chapter Naming**: The name of each chapter is derived from the most significant words in the text that represents each chapter.

**Advanced Features**

* **Sentiment Analysis**: Integrate sentiment analysis to track the emotional tone of each chapter.
* **Emotion Detection**: Apply emotion detection models to assess emotions like joy, sadness, anger, etc., in each chapter.
* **Interactive Web Interface**: Implement a Streamlit or Dash web app for users to interactively visualize topics, chapters, and more.
* **YouTube API Integration**: Fetch video metadata such as views, likes, and comments to analyze the relationship between video popularity and chapter topics.

**Future Improvements**

* **Better Chapter Naming**: Use NLP techniques like Named Entity Recognition (NER) or summarization models for more meaningful chapter names.
* **Model Tuning**: Experiment with different topic modeling techniques and hyperparameters to improve topic coherence and interpretability.
* **Audio Integration**: Integrate audio or video files directly to synchronize the transcript with video content.

**Conclusion**

This project provides an effective way to automatically structure video content into chapters, enhancing accessibility and navigation. By leveraging NLP and machine learning, we can achieve automated chapterization that helps users better interact with and consume video content.