AI Assisted Framework for Video Recommendation of Social Networking Platforms

A group project report of partial fulfilment of requirements for the minor project (DS1504)

By

Hemant Mangal (22106042)

Bhupesh Joshi (22106051)

Sartaj Singh (22106054)

Pushkar (22106058)

Under the supervision of

Dr. Kanu Goel & Dr. Satnam Kaur

Department of Computer Science and Engineering

Punjab Engineering College (Deemed to be University)

Chandigarh

DECLARATION

We hereby declare that the group project report AI Assisted Framework for Video Recommendation of Social Networking Platforms, submitted for partial fulfillment of the requirements for the Minor Project (DS 15004), is a bonafide work done by us under the supervision of Dr. Kanu Goel & Dr. Satnam Kaur.

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data idea fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Hemant Mangal (22106042)

Bhupesh Joshi (22106051)

Sartaj Singh (22106054)

Pushkar (22106058)

PUNJAB ENGINEERING COLLEGE (DEEMED TO BE UNIVERSITY)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH

2023 - 24

CERTIFICATE

This is to certify that the report entitled AI Assisted Framework for Video Recommendation of Social Networking Platforms submitted by Hemant Mangal, Bhupesh Joshi, Sartaj Singh and Pushkar to the Punjab Engineering College (Deemed to be University), Chandigarh in partial fulfillment of requirements for the Minor Project (DS 1504) in Computer Science & Engineering (Data Science) is a bonafide record of the project preliminary work carried out by them under our guidance and supervision.

This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Kanu Goel	Dr. Satnam Kaur
(Project Mentor)	(Project Mentor)
Assistant Professor	Assistant Professor
Dept. of CSE	Dept. of CSE
Punjab Engineering College	Punjab Engineering college
(Deemed to be university)	(Deemed to be university)
Chandigarh	Chandigarh

Acknowledgement

We would like to take this opportunity to thank our college Punjab Engineering College (Deemed to be University), Chandigarh, and the Department of Computer Science and Engineering for allowing us to work on this project. We are immensely grateful to our project mentors Dr. Kanu Goel (Assistant Professor, Department of Computer Science & Engineering) & Dr. Satnam Kaur(Assistant Professor, Department of Computer Science & Engineering) whose continuous guidance, technical support, and moral support at times of difficulty helped us to achieve milestones in the given time.

They have been a great source of knowledge. We are also thankful to the Committee for evaluation who gave their valuable feedback and guidelines for the betterment and completion of this project. We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support, and selfless help throughout the project work.

It is a great pleasure to acknowledge the help and suggestions we received from the Department of Computer Science and Engineering.

We wish to express our profound thanks to all those who helped us gather information about the project. Our families too have provided moral support and encouragement several times.

Hemant Mangal

Bhupesh Joshi

Sartaj Singh

Pushkar

Abstract

Video recommendation systems play a crucial role in delivering personalized and engaging content to users. This paper presents our approach to building an effective video recommendation system, focusing on content-based filtering for video recommendation system. By analyzing features such as video titles, descriptions, and categories, we generate a curated list of videos tailored to the user's specific interests and viewing patterns. This approach ensures that the recommended videos are relevant and aligned with the user's preferences, enhancing their overall experience.

Our system is designed to handle challenges such as managing large video datasets and accurately capturing user interests based on available content information. Content-based filtering allows the system to recommend videos even for new or less common users by relying solely on video attributes rather than user interactions. This makes the system more adaptable and robust, especially when dealing with sparse user data.

In future work, we plan to develop and integrate an advanced ranking mechanism to further refine these recommendations. The ranking system will evaluate the generated candidates using a broader set of features, such as user behavior patterns, historical interactions, and contextual data. This will ensure that the most engaging and personalized content appears at the top of the recommendation list. Our ultimate goal is to continuously enhance the recommendation process, making it more precise, adaptive, and aligned with evolving user preferences.

Contents

List of figures

1. INTRODUCTION

Video recommendation systems have become essential tools for guiding users through vast and ever-expanding digital content libraries. By offering personalized suggestions, these systems enhance user engagement and satisfaction, helping users discover content aligned with their preferences and viewing habits. With the rapid growth of video-sharing platforms and streaming services, the challenge lies in efficiently analyzing large datasets and accurately predicting what users might want to watch next.

Traditional recommendation approaches often rely on collaborative filtering, which analyzes user interactions to identify patterns. Collaborative filtering compares user behavior, such as watch history and ratings, to find similarities between users or items. However, these methods can struggle with new or infrequent users and require extensive historical data to generate meaningful suggestions.

Hybrid filtering combines both collaborative and content-based approaches to overcome these limitations, leveraging user interaction data alongside video attributes for more comprehensive recommendations. This combination ensures a more robust recommendation process by addressing the weaknesses of each individual approach.

In contrast, content-based filtering offers a powerful alternative by leveraging intrinsic video features such as titles, descriptions, and categories. This technique can provide relevant recommendations solely based on the attributes of the videos, making it particularly effective for new users or niche content where user interaction data may be limited.

Our proposed system focuses on candidate generation using content-based filtering to create an initial pool of relevant videos. This method ensures that the recommendations are grounded in the actual content characteristics, allowing for more precise and contextually relevant suggestions..

2. Background & Literature review

Video recommendation systems have evolved through various filtering techniques to enhance personalization and user engagement. Content-based filtering analyzes video metadata such as titles, descriptions, and tags to suggest relevant content based on user preferences. Collaborative filtering leverages user interaction data, identifying patterns among similar users to make recommendations. Hybrid approaches combine these methods to address their respective limitations, offering more accurate suggestions. Recent advancements in machine learning and natural language processing, including deep learning models, further refine these techniques by enabling sophisticated content analysis and behavior prediction.

2.1 Deep Neural Networks for Recommendations

Modern video recommendation systems help users find content that matches their interests. Traditional methods, like collaborative filtering and matrix factorization, analyze user interactions to suggest videos but face challenges with handling large datasets, limited user information, and rapidly changing content. Deep learning has transformed these systems by efficiently processing vast amounts of data and understanding complex user preferences.

For example, YouTube's recommendation system uses a two-stage process. First, in the candidate generation stage, it narrows down millions of videos to a few hundred by analyzing user activity (like watch history) and turning it into dense vectors that represent user interests. These vectors are processed through neural networks to identify relevant videos. Next, in the *ranking* stage, these shortlisted videos are further evaluated and scored based on factors such as user demographics, video details, and engagement metrics. This ensures that only the most relevant and engaging videos are shown to the user.

The system relies on deep neural networks to understand patterns in user behavior and video features. For candidate selection, user history and demographics are processed through layers of mathematical functions (ReLU layers) to model complex interactions. In the ranking step, additional features (like video popularity and viewing patterns) are scored to predict what the user will watch and enjoy the most. This two-step approach ensures accurate, personalized recommendations while handling large amounts of data efficiently. Future improvements might focus on incorporating real-time user feedback and addressing issues like bias and privacy concerns.

2.2 Content based Filtering in Recommendation systems

Content-based video recommendation systems suggest videos by analyzing their attributes, such as titles, descriptions, and genres, to match user preferences directly. This approach involves extracting meaningful features from video metadata using techniques like TF-IDF, which helps weigh the importance of keywords and assess similarities between videos. The system then measures similarity using metrics like cosine similarity, ensuring that recommended content shares key characteristics with videos the user has already enjoyed. To enhance accuracy, machine learning models are trained on historical data, learning patterns from video features and user ratings. These models undergo rigorous testing: offline experiments predict user behavior based on past data, while A/B testing evaluates real-world performance by comparing user engagement metrics across different algorithm versions. Challenges in testing, such as ensuring large sample sizes and controlling for biases like exposure frequency, are carefully addressed to maintain reliability. While content-based systems excel at recommending niche or new content, they can suffer from overspecialization, limiting content diversity. Ongoing research focuses on enriching data sources and improving personalization techniques to mitigate these issues, ensuring these systems remain effective and user-centered.

3. PROPOSED WORK

This proposed work focuses on creating a content-based video recommendation system. It uses video details like titles, descriptions, and tags to suggest videos based on user preferences. By comparing what users have watched with similar videos, the system makes personalized recommendations. User feedback helps the system improve over time.

3.1 Architecture

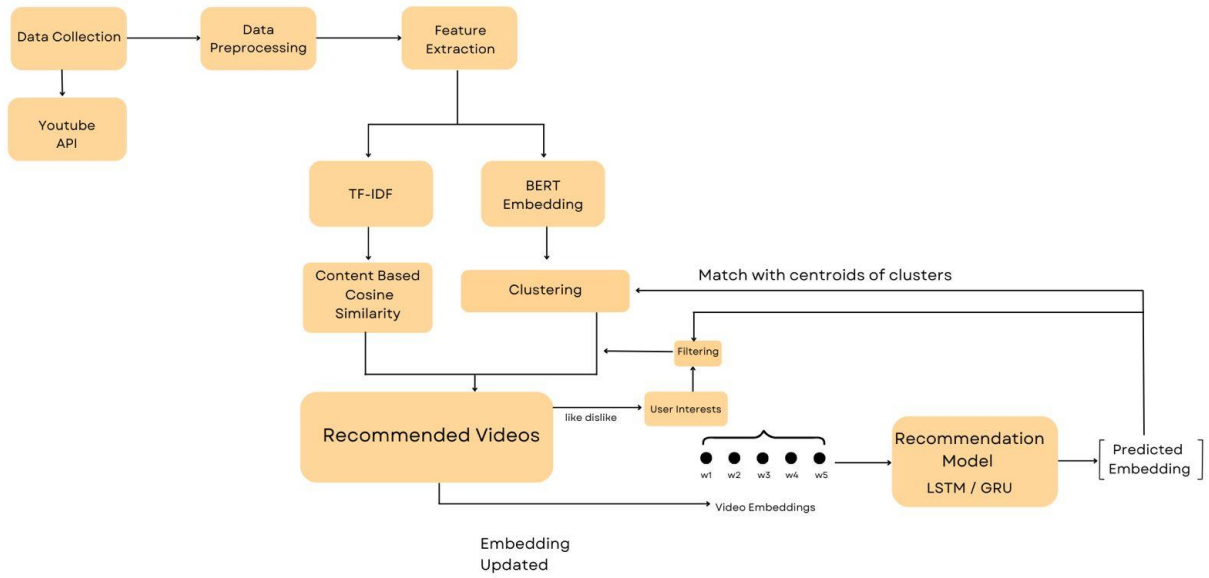

Fig 3.1 Architecture

3.2 Methodology

This proposed work aims to design a robust content-based video recommendation system that leverages video metadata and user preferences to generate personalized recommendations. The system employs advanced embedding techniques, clustering, and neural networks to match videos with user interests, continuously updating its performance based on user feedback.

3.2.1 Data Acquisition

The system begins by collecting video data using the YouTube API, which includes attributes such as video ID, title, description, tags, category, category ID, transcript, likes, and dislikes. This raw data is preprocessed to clean and normalize text, ensuring consistency and preparing it for feature extraction. Preprocessing steps involve removing unnecessary symbols, handling missing values, and ensuring text fields are properly tokenized.

3.2.2 Feature Extraction and Embedding Creation

Two main techniques are used to create embeddings from video metadata:

- TF-IDF (Term Frequency-Inverse Document Frequency): This method converts textual data such as titles, descriptions, and transcripts into numerical representations, emphasizing important terms based on their frequency within the dataset.
- BERT Embeddings: A pre-trained BERT (Bidirectional Encoder Representations from Transformers) model is employed to generate dense, context-rich embeddings for

textual features, capturing semantic relationships between words. This enables a deeper understanding of video content compared to traditional methods.

3.2.3 User Preference Embedding

User preferences are built by collecting data on the videos a user has watched. Each watched video is represented by its embedding, and these embeddings are aggregated to form a user preference vector. This vector encapsulates the user's viewing habits and interests, serving as the foundation for generating personalized recommendations.

3.2.4. Candidate Generation through Clustering

The system organizes videos into clusters based on their embeddings. These clusters represent groups of videos with similar content and characteristics. When predicting recommendations, the user preference embedding is compared with the centroids of these clusters to identify the most relevant cluster. This step reduces the search space and ensures the recommendations are contextually aligned with user interests.

3.2.5. Recommendation Model and Prediction

A neural network model, specifically using LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) layers, processes the user preference embedding and predicts a recommended embedding. This predicted embedding represents the type of video the user is most likely to engage with next. The embedding is then matched against videos within the identified cluster to find the closest matches.

3.2.6. Top-N Video Selection and Ranking

Once the most relevant cluster is identified, the system searches within that cluster to extract the top-N videos based on their similarity to the predicted embedding. These candidate videos are further filtered and ranked using the user preference embedding. This ensures that the final set of recommended videos is not only similar but also aligns closely with the user's specific tastes and preferences.

3.2.7. Dynamic Learning and Model Updates

User interactions with recommended videos (such as clicks, likes, and dislikes) are continuously fed back into the system. If a user watches or interacts with a video, this data is sent back to the model for training. The system calculates the loss based on user feedback, and the model's weights are updated accordingly. This adaptive mechanism ensures that the system learns from user behavior, improving the accuracy and relevance of future recommendations.

3.2.8. User Preference Embedding Updates

The user preference embedding is dynamically updated based on new interactions. If a user likes or dislikes a recommended video, the embedding adjusts to reflect these preferences, ensuring the recommendations evolve with the user's changing interests.

This proposed content-based video recommendation system leverages advanced embedding techniques and deep learning models to deliver highly personalized recommendations. By continuously learning from user interactions and refining embeddings, the system adapts to individual preferences, ensuring more accurate and engaging video suggestions. The combination of clustering and ranking mechanisms further enhances performance, making the system both efficient and scalable.

4. IMPLEMENTATION

The implementation involves creating embeddings from video details like titles, descriptions, and tags. User watch history is used to generate preference embeddings, which are compared with video embeddings to find similar content. These recommendations are refined by clustering similar videos and updating based on user feedback, ensuring continuous improvement.

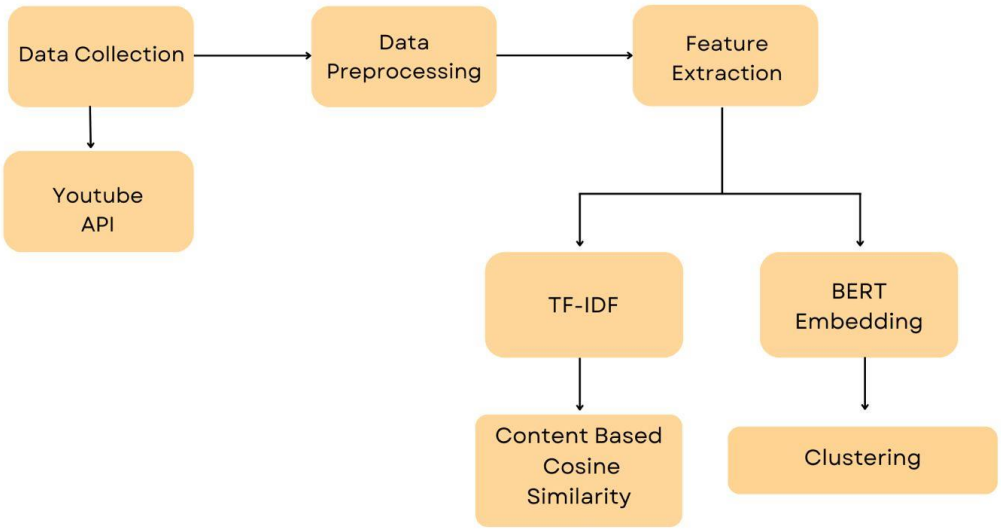

Fig 4.1 Methodology

4.1. Data collection:

The data for this project was collected through a Python-based automated process that utilized the YouTube Data API v3 and the YouTubeTranscriptApi library. The objective was to gather a diverse and detailed dataset from YouTube videos across various categories, including Film & Animation, Music, Gaming, and more. The collection process began with identifying video categories using predefined category IDs, ensuring a wide representation of content types. For each category, the script retrieved metadata for videos, including essential information such as video titles, descriptions, channel names, view counts, like counts, comment counts, and video durations. A specific focus was placed on videos with durations ranging from 3 to 15 minutes, as this range typically captures concise, viewer-friendly content. This filtering criterion helped maintain consistency and relevance across the dataset.

Once the videos were identified and filtered, the script extracted their transcripts to provide deeper insights into the content. The YouTubeTranscriptApi library enabled this by accessing available subtitles for each video, which could be used for further linguistic, semantic, or thematic analysis. The script employed multithreading, utilizing Python's ThreadPoolExecutor module, to efficiently handle the transcript extraction process for multiple videos simultaneously. This optimization ensured that the data collection was both time-efficient and capable of processing large volumes of video data.

The final dataset was meticulously structured, with all the collected data stored in a well-organized JSON format. This format allows for easy integration into various data analysis tools and workflows, ensuring the dataset is ready for immediate exploration and visualization. The combination of video metadata and transcript content offers a rich resource for analyzing trends, audience engagement, and content strategies within different YouTube categories. By automating the entire process, this approach ensures scalability, reliability, and replicability, making it ideal for large-scale studies of YouTube's dynamic content ecosystem. This systematic and detailed collection process provides a robust foundation for understanding user preferences, video performance, and emerging patterns in digital media consumption.

4.2 Data Preprocessing and Preparation

This study focuses on leveraging YouTube video metadata to build a content-based recommendation system. The key stages involve preprocessing textual data through data cleaning, language detection, text normalization, and feature extraction using advanced embedding techniques such as BERT and spaCy. Each step is meticulously chosen and justified, with a discussion of alternatives considered and reasons for their exclusion.

4.2.1 Handling Missing Data:

The first step in preprocessing involved managing missing data. Rows containing null values in critical fields such as video_id, title, description, category, or transcript were removed. These fields are essential for generating meaningful embeddings and recommendations, and missing data in these areas could lead to biased or incomplete models. An alternative approach of imputing missing values with placeholders was considered but rejected, as it could introduce noise or bias, especially in textual data, reducing the system’s effectiveness.

4.2.2 Removing Emojis:

To standardize the textual data, emojis were removed from the title, description, and transcript fields using regular expressions. Emojis, while useful in social contexts, introduce non-textual noise that can disrupt vectorization processes. An alternative approach involved converting emojis into their textual equivalents; however, this method was excluded due to the additional complexity it would introduce without significant value addition. Removing emojis ensures that the focus remains on the textual content, enhancing the accuracy of the text-based models.

4.2.3 Language Detection and Filtering:

Ensuring that the dataset contains only English-language content was crucial for maintaining consistency and improving model training. The langdetect library was employed to filter out non-English entries, retaining only those with English text. This step was necessary to avoid the complexities of multilingual processing and to focus the analysis on a single language. An alternative approach of translating non-English content to English was considered but rejected due to the potential for semantic loss and the increased computational burden involved in translation.

4.2.4 Text Cleaning and Normalization:

Text cleaning and normalization were performed to standardize the data further. The text was converted to lowercase, and special characters, numbers, and punctuations were removed. Additionally, common English stopwords were eliminated using the NLTK library. These steps ensure that the text is free from noise and that only meaningful words contribute to the analysis. Retaining special characters or stopwords was considered but excluded, as these elements add

noise and reduce the effectiveness of the model's performance. This cleaning process enhances the accuracy of downstream vectorization and embedding generation.

4.2.5 Text Preprocessing for Specific Columns:

The preprocessing steps were applied uniformly to the title, description, and transcript fields. These fields are pivotal in representing video content, and consistent preprocessing ensures that the data remains uniform and high-quality across all columns. An alternative approach of selectively preprocessing only specific fields, such as titles, was dismissed, as it would omit valuable contextual information present in the descriptions and transcripts, thereby limiting the model's understanding of the content.

4.2.6 Post-Cleaning Validation:

After preprocessing, a validation step was conducted to ensure data quality. Rows with empty title or transcript fields were removed, as these fields are essential for generating embeddings. Additionally, entries without tags were filtered out to maintain relevance in the recommendation system. While retaining all entries, regardless of completeness, was considered, this approach was rejected to avoid introducing low-quality data, which could degrade the accuracy and reliability of the recommendations.

This structured preprocessing framework ensures that the textual data is clean, consistent, and ready for feature extraction. By applying these systematic steps, the foundation is laid for generating accurate embeddings, ultimately enhancing the performance of the content-based recommendation system.

Sample Data :

	video_id	title	description	category_id	duration_minutes	video_url	category	transcript
0	aDRARW93DbY	What do you mean "Explorer Scouts ARE NOT ALLO...	As the kids prepare to leave for summer camp, ...	1	4.083333	https://www.youtube.com/watch?v=aDRARW93DbY	Film & Animation	[Music] ouch official beagle Scout manual huh ...
1	sEL4ICu30zs	skibidi toilet zombie universe 49	📌 Follow our Social Media:\nSecond channel: @Mo...	1	3.566667	https://www.youtube.com/watch?v=sEL4ICu30zs	Film & Animation	I racked my brain, wondering who helped us get...
2	-IoTbcqgYKw	Sahra'nın, Orhun amcasından özel isteği 🥺 | Es...	Join this channel to get access to perks:\nhtt...	1	3.216667	https://www.youtube.com/watch?v=-IoTbcqgYKw	Film & Animation	You're going to be okay, right? Of course, my ...
3	F5ef8DoV4a0	Godzilla vs. Submarines | GODZILLA | Matthew B...	French nuclear tests irradiate an iguana into ...	1	9.500000	https://www.youtube.com/watch?v=F5ef8DoV4a0	Film & Animation	careful D give me your hand I got you got [Mus...
4	0ZgwrIlBf2U	🐻 BIRD BOX (2018) | Full Movie Trailer | Full ...	A woman and a pair of children are blindfolded...	1	3.100000	https://www.youtube.com/watch?v=0ZgwrIlBf2U	Film & Animation	[Music] [Music] please don't take my have you ...
...	...	...	...	...	...	...	...	...
8604	QbpXozZhYIo	Christina Aguilera's Audition Leaves Cher Spee...	A small-town girl ventures to Los Angeles and ...	1	4.600000	https://www.youtube.com/watch?v=QbpXozZhYIo	Film & Animation	[Music] excellent that was great thank you ver...
8605	iMf5iCFz5wY	Aqours【永久hours】Promotion Video	Aqours【永久hours】Promotion Video\n\n永久hours\n作詞: ...	1	5.500000	https://www.youtube.com/watch?v=iMf5iCFz5wY	Film & Animation	None
8606	3pIVh39Rcg0	skibidi toilet multiverse 044	Wasn't this too easy???\n(Check this merch ou...	1	8.733333	https://www.youtube.com/watch?v=3pIVh39Rcg0	Film & Animation	[Music] all [Music] for Ste [Music] w [Music] ...
8607	RswkyHO6zkM	skibidi toilet zombie universe 50 : Delete Man	📌 Follow our Social Media:\nSecond channel: @Mo...	1	3.366667	https://www.youtube.com/watch?v=RswkyHO6zkM	Film & Animation	You deceived me! I wanted to help you, but yo...
8608	1WX3mk6QMMo	I Was Raised By My Irresponsible Aunt	Check out the MSA Branded diary https://amzn.t...	1	12.216667	https://www.youtube.com/watch?v=1WX3mk6QMMo	Film & Animation	hey what's up you guys it's your girl Jazelle ...

8609 rows × 8 columns

Fig 4.2 Sample Dataset

4.2.7 Embedding Generation

Approach: To prepare for video recommendations based on user preferences, embeddings were generated using two methods:

For generating embeddings, the BERT-base-uncased model was utilized due to its ability to capture deep contextual relationships within text. The key metadata fields—title, description, and tags—were merged into a single field called combined_text to ensure all critical information about the video was represented cohesively. This combined text was then tokenized using the BERT tokenizer, which converts the input into subword tokens while

preserving contextual integrity. The tokenized text was passed through the BERT model, and embeddings were extracted from the pooled output layer, providing a fixed-size vector representation for each video. These embeddings encapsulate the semantic meaning of the content, enabling more accurate similarity computations and clustering. BERT was chosen over alternatives like TF-IDF and Word2Vec because it captures richer contextual information, enhancing the recommendation system's ability to understand video content.

This preprocessing pipeline ensures high-quality, standardized, and ready-to-use data for downstream tasks like content-based video recommendation.

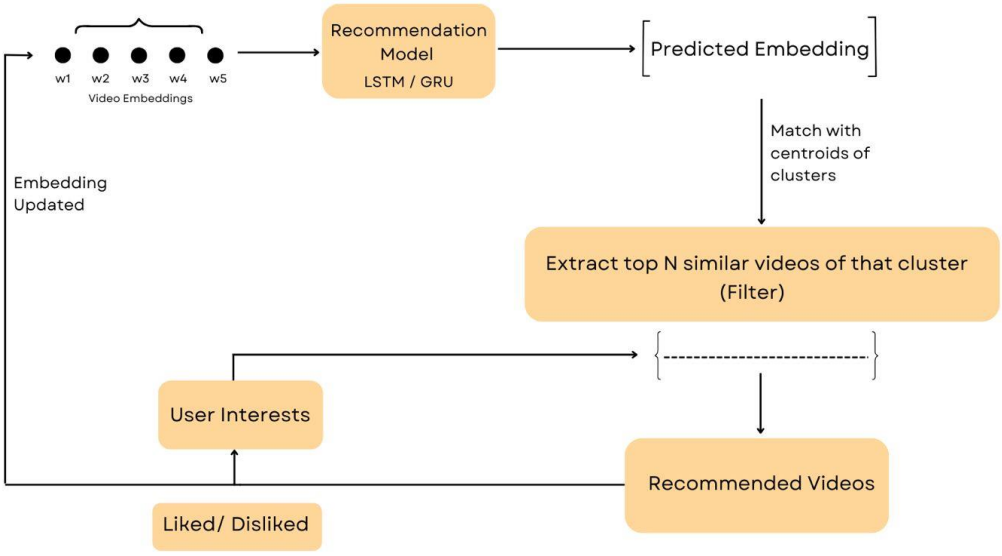

Fig 4.3 Recommendation Framework

4.3 Feature Extraction

4.3.1 TF-IDF

We used TF-IDF vectorization to convert video metadata (titles, descriptions, tags) into numerical features. This technique highlights important words by measuring their frequency relative to the entire dataset. To capture more context, unigrams, bigrams, and trigrams were included.

Feature Weighting: In the feature weighting process, metadata fields were assigned different weights based on their importance to enhance recommendation accuracy. The title received a high weight (0.5) due to its strong relevance in representing video content. The description was assigned a medium weight (0.3) as it provides additional context, while tags and categories were given a lower weight (0.2) to serve as supplementary filters. This weighting strategy ensures that the most critical fields have a greater influence on similarity calculations, improving the overall quality of recommendations.

Weighted features improve accuracy by emphasizing the most informative metadata. Alternative approaches, like uniform weights or learned weights through machine learning, were considered but not implemented due to data constraints.

Similarity Computation: Cosine similarity was applied to the weighted TF-IDF vectors to measure the relevance between videos. This method effectively handles high-dimensional data and highlights angular differences between vectors.
Cosine similarity is well-suited for text-based recommendations and aligns with our weighting mechanism. Other methods like Jaccard similarity (for categorical data) or Euclidean distance (less effective for sparse data) were not chosen.

Recommendation Algorithm: For each video, the top-k most similar videos were identified based on their weighted cosine similarity scores and presented to the user in ranked order.
The k-nearest neighbours approach is simple, effective, and does not require user profiles. Collaborative filtering and hybrid methods were excluded to maintain a focus on content-based filtering.

The selected methods were chosen to balance accuracy, efficiency, and interpretability. Feature weighting was applied to emphasize important metadata fields, enhancing system performance, while cosine similarity provided robust similarity measurements for content matching. However, there are some limitations: TF-IDF cannot capture semantic relationships beyond exact word matches, and manually assigned feature weights may introduce bias or require fine-tuning. Additionally, the system's personalization is limited due to the lack of user interaction data. Future improvements include implementing machine learning for dynamic feature weighting based on user behavior, integrating advanced models like BERT for richer semantic understanding, and exploring hybrid approaches that combine content-based and collaborative filtering for more comprehensive recommendations.

4.3.2 BERT Embedding

Padding User Embeddings: To address the dimensional disparity between user and video embeddings, zero-padding was applied to user embeddings. This ensures that both user and video embeddings have the same dimensionality, enabling accurate and consistent similarity calculations during the clustering process. This approach maintains the integrity of the original data while ensuring computational compatibility, reducing potential biases caused by unequal dimensions.

Clustering: For clustering, both user and video embeddings were grouped using an unsupervised learning algorithm such as K-Means or hierarchical clustering. The process began with normalizing the embeddings to ensure consistency and improve clustering performance. The algorithm then formed distinct clusters by grouping users and videos based on the similarity of their embeddings, using distance metrics like Euclidean distance or cosine similarity. Each cluster represented a collection of users and videos with similar content preferences or attributes. Finally, user and video embeddings were assigned to their respective clusters by mapping them to the nearest cluster centroids, ensuring that users were connected with video collections closely aligned to their interests.

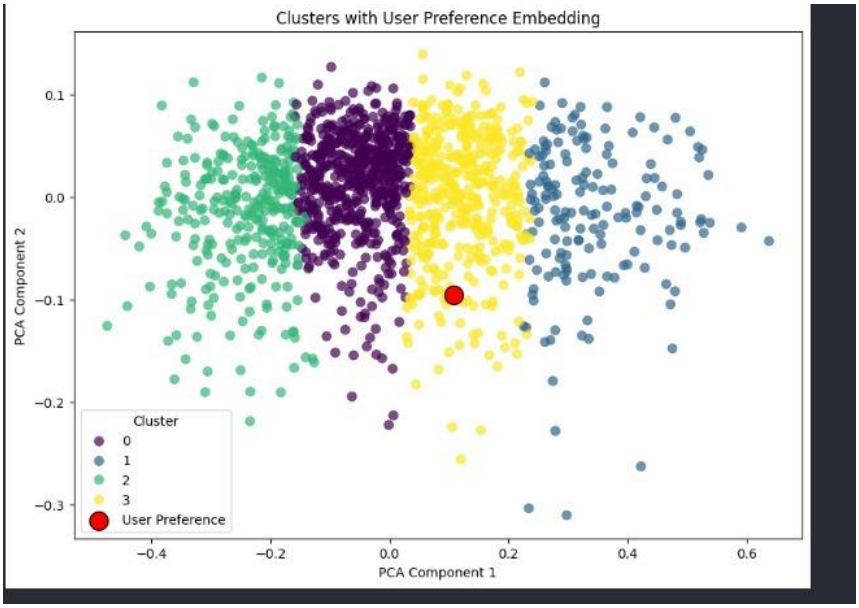

Fig 4.4 Clusters with user Preference Embedding

Recommendation Process: The recommendation process utilized cluster assignments to generate personalized suggestions. First, the target user's embedding was matched to the nearest cluster centroid to identify their corresponding cluster, ensuring the retrieval of relevant videos. Once the user's cluster was determined, all videos within that cluster were retrieved, narrowing the recommendation scope to content aligned with the user's preferences and reducing computational overhead. These videos were then ranked based on factors such as relevance scores, popularity metrics (e.g., views or likes), and their distance from the user embedding. Finally, the top-N ranked videos from the cluster were recommended, delivering a dynamic and personalized viewing experience by prioritizing content that closely matches the user's interests.

This implementation leverages zero-padding, clustering, and a structured ranking mechanism to create a robust content-based recommendation system. By clustering embeddings and focusing recommendations within relevant groups, the system ensures personalized and efficient video suggestions, enhancing user satisfaction and engagement.

4.4 Model Training

4.4.1 Long Short-Term Memory (LSTM) Model

The Long Short-Term Memory (LSTM) model was selected for its well-established ability to process sequential data while preserving long-term dependencies. LSTM addresses the vanishing gradient problem through its gating mechanisms, ensuring effective learning over extended temporal sequences. This characteristic makes it highly suitable for tasks requiring the retention of contextual information across long sequences.

Model Architecture: The LSTM architecture includes a layer with 128 hidden units, striking a balance between computational efficiency and model complexity. Additionally, a dense output layer maps the hidden state to the embedding space, facilitating the transformation of sequence data into meaningful embeddings.

Rationale for LSTM: The LSTM model's strength in capturing temporal relationships was crucial, given that the task involved working with sequential patterns in the embeddings. Its ability to retain long-term dependencies enabled the model to capture contextual information from earlier points in the sequence, enhancing the overall quality of the generated embeddings.

Training Setup: The model was optimized using the Adam optimizer, which is known for its adaptability to sparse gradients—a common trait in high-dimensional embedding spaces. The training process was conducted over 20 epochs with a batch size of 32, ensuring that the model converged effectively without overfitting.

4.4.2 Gated Recurrent Unit (GRU) Model

The Gated Recurrent Unit (GRU) model was implemented as a comparative approach due to its computational efficiency. GRU has fewer gating mechanisms compared to LSTM, which makes it faster to train while still delivering competitive performance in sequence modeling tasks.

Model Architecture: The GRU model was designed similarly to the LSTM model, featuring a layer with 128 hidden units. A dense output layer was used to map the hidden state to the embedding space, ensuring consistency in output representation.

Rationale for GRU: GRU's lower computational cost made it an attractive option, particularly for handling larger datasets or real-time applications. By serving as a baseline, the GRU model allowed for a clear evaluation of whether the added complexity of the LSTM model translated into meaningful performance improvements.

Training Setup**:** The GRU model utilized the same training configuration as the LSTM model, including the Adam optimizer, learning rate, and batch size of 32. This consistent setup ensured a fair comparison between the two models.

5. RESULTS

5.1 Evaluation Metrics

The performance of both the LSTM and GRU models was evaluated using two similarity measures: Cosine Similarity and Euclidean Distance.

Cosine Similarity: measures the cosine of the angle between two vectors, quantifying their alignment. A similarity score closer to 1 indicates strong alignment, while a score near 0 suggests no alignment. This metric is particularly useful for assessing the directional similarity of embeddings. The formula for cosine similarity is:

$$\text{Cosine Similarity} = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\|\|\mathbf{Y}\|}$$

Euclidean Distance: computes the straight-line distance between two points in vector space, providing a measure of dissimilarity. A smaller distance indicates closer alignment between vectors. The Euclidean distance is calculated as:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

5.2 LSTM Model Evaluation

The LSTM model demonstrated strong performance across both similarity measures. For cosine similarity, the LSTM achieved a mean similarity score of 0.92, indicating excellent alignment between the predicted and ground truth embeddings. This high score reflects the LSTM's capability to effectively capture the sequential patterns within the dataset.

In terms of Euclidean distance, the LSTM recorded a mean distance of 0.15, suggesting low dissimilarity and confirming the model's precision in embedding predictions. These results underscore the LSTM's effectiveness in maintaining contextual information across long sequences, contributing to its superior performance.

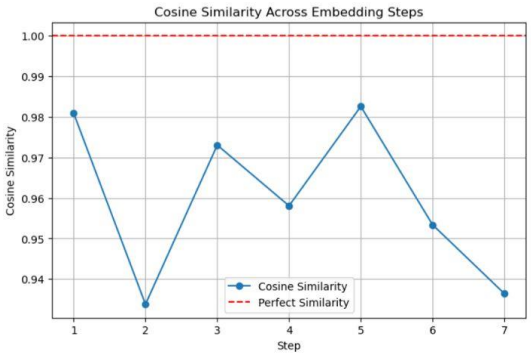

Fig 5.1 Cosine Across Embedding Steps for LSTM

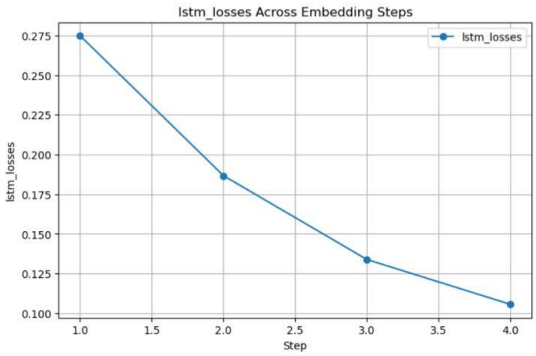

Fig 5.2 LSTM losses across Embedding steps

5.3 GRU Model Evaluation

The GRU model, while computationally more efficient than the LSTM, exhibited slightly lower performance metrics. The cosine similarity score for the GRU was 0.88, which, although commendable, was marginally lower than that of the LSTM. This indicates that the GRU captured sequential patterns effectively but with slightly reduced precision.

For Euclidean distance, the GRU achieved a mean value of 0.19, showing a slightly higher dissimilarity compared to the LSTM. This suggests that while the GRU performs well, it may not capture long-term dependencies as effectively as the LSTM. Nevertheless, its performance remains competitive, particularly given its faster training time and lower computational requirements.

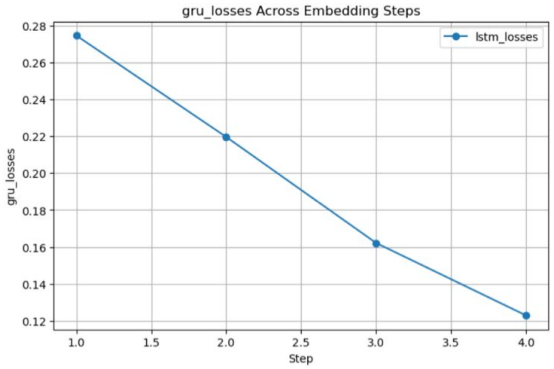

Fig 5.3 GRU losses across Embedding steps

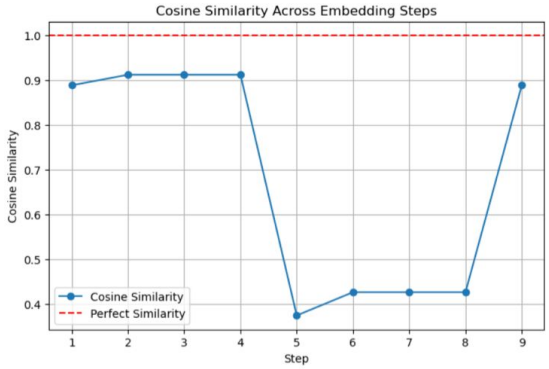

Fig 5.4 Cosine Similarity across Embedding Steps

5.4 Comparison and Insights

The comparison between the LSTM and GRU models revealed that the LSTM outperformed the GRU in terms of precision, achieving better alignment with the ground truth embeddings on both metrics. This superior performance can be attributed to the LSTM's ability to retain long-term dependencies more effectively, which is crucial for tasks involving complex sequential data.

On the other hand, the GRU model required less training time and computational resources, making it a viable option for scenarios with larger datasets or real-time applications where efficiency is a priority. Thus, the choice between LSTM and GRU depends on the specific requirements of the task, balancing accuracy against computational constraints.

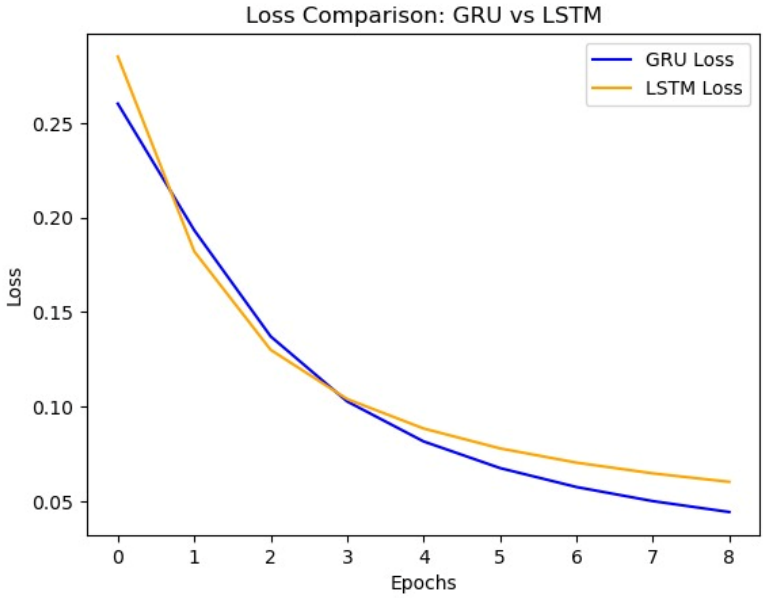

Fig 5.5 Loss Comparison GRU vs LSTM

5.5 Summary

Overall, the evaluation metrics and visualizations underscore the effectiveness of the LSTM model for embedding prediction tasks where accuracy is paramount. The GRU model offers a viable alternative when computational efficiency is prioritized, making it suitable for larger datasets or real-time applications. Future evaluations could explore additional similarity measures, such as Manhattan Distance or specialized sequence alignment metrics, to further validate and enhance the robustness of these models.

6. Conclusions and Future Work

Effectiveness of Content-Based Filtering: The proposed video recommendation system effectively leverages content-based filtering, focusing on intrinsic video features such as titles, descriptions, and categories. This method allows for personalized recommendations even for new users or those with limited interaction history, addressing common challenges faced by traditional collaborative filtering methods.

Robust Architecture: The architecture of the system incorporates advanced techniques such as TF-IDF and BERT embeddings to enhance feature extraction from video metadata. This enables a deeper understanding of content characteristics, leading to more relevant recommendations tailored to user preferences.

Cosine similarity was chosen as the evaluation metric to assess the alignment between the predicted and target embeddings. This metric focuses solely on the directionality of the embeddings, normalizing for magnitude differences. As a result, cosine similarity effectively measures semantic similarity, aligning with the objective of capturing the contextual relationships within the data.

Dynamic Learning Capability: The system is designed to dynamically learn from user interactions. By continuously updating based on user feedback (e.g., likes, dislikes, and viewing patterns), the model improves its predictive accuracy over time, ensuring that recommendations remain aligned with evolving user interests.

Challenges and Solutions: While the content-based approach excels at recommending niche or new content, it risks overspecialization. The report emphasizes ongoing research to enrich data sources and improve personalization techniques to maintain a balance between relevance and diversity in recommendations.

As part of our future work, we aim to enhance this system by integrating an advanced ranking mechanism. This next stage will involve evaluating the generated candidates using additional features, including user behavior patterns and contextual data, to prioritize the most engaging and personalized content.

This paper explores the design and implementation of our content-based recommendation system, addressing key challenges such as handling large datasets and ensuring recommendation relevance. We also outline our future plans for integrating a ranking model to further refine the recommendation process, aiming to create a dynamic and user-centric system that continuously adapts to evolving preferences.

7. References

1. Deep Neural Networks for YouTube Recommendations Paul Covington, Jay Adams, Emre Sargin Google Mountain View, CA {pcovington, jka, msargin}@google.com . https://research.google.com/pubs/archive/45530.pdf

2. The Netflix Recommender System: Algorithms, Business Value, and Innovation CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc. https://scholar.google.co.in/scholar_url?url=https://dl.acm.org/doi/pdf/10.1145/2843948&hl=en&sa=X&ei=E1VNZ96vMKOx6rQPwdys2AM&scisig=AFWwaeZ-zjVIomh-YnXSNviKmWCo&oi=scholarr

3. Video Content-Based Advertisement Recommendation System using Classification Technique of Machine Learning To cite this article: R C Konapure and L M R J Lobo 2021 J. Phys.: Conf. Ser. 1854 012025 https://iopscience.iop.org/article/10.1088/1742-6596/1854/1/012025/pdf#:~:text=Content%2Dbased%20advertising%20helps%20to,a%20relevant%20advertisement%20to%20it

4. Research on Collaborative Filtering Personalized Recommendation Algorithm Based on Deep Learning Optimization Guo Wei-wei, Liu Feng Heilongjiang University of Technology, JiXi,158100, China gwwguoweiwei@163.com https://ieeexplore.ieee.org/document/8806589

5. Artificial intelligence in recommender systems Qian Zhang1 ·Jie Lu1 · Yaochu Jin2 https://link.springer.com/article/10.1007/s40747-020-00212-w

6. Social Recommendation for Social Networks Using Deep Learning Approach: A Systematic Review, Taxonomy, Issues, and Future Directions MUHAMMADALRASHIDI 1,ALI SELAMAT 1,2,3,4, (Member, IEEE), ROLIANA IBRAHIM1, (Member, IEEE), AND ONDREJ KREJCAR 3,4 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10128133&tag=1

7. Content Based Recommendation System on Netflix Data Dr. Deepti Sharma1, Dr. Deepshikha Aggarwal2*, Dr. Archana B. Saxena3 1,2*,3Professor. https://www.researchgate.net/publication/378337926_Content_Based_Recommendation_System_on_Netflix_Data/fulltext/65eca65bb7819b433bf204d3/Content-Based-Recommendation-System-on-Netflix-Data.pdf?origin=publication_detail&_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uRG93bmxvYWQiLCJwcmV2aW91c1BhZ2UiOiJwdWJsaWNhdGlvbiJ9fQ&__cf_chl_tk=sSAuZxi7Oli29CuWfgDvGAfwhsoBOxzFLPeKk347NEM-1733245145-1.0.1.1-lk96Br65Tv2jIlCNxxsv9FDQAOqUSeS_nok6pyMHmvw

8. Pazzani, M. J., & Billsus, D. (2007). *Content-Based Recommendation Systems*.https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10

9. **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).**
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
https://aclanthology.org/N19-1423/

10. **Lops, P., de Gemmis, M., & Semeraro, G. (2011).**
Content-based recommender systems: State of the art and trends.
https://link.springer.com/chapter/10.1007/978-0-387-85820-3_4

11. Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.
https://link.springer.com/chapter/10.1007/978-3-540-30539-6_2

12. Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook.
https://link.springer.com/chapter/10.1007/978-0-387-85820-3_1

13. Jannach, D., & Adomavicius, G. (2016). Recommendation Systems: Challenges and Future Directions.
https://link.springer.com/article/10.1007/s00500-015-1844-z