International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

MOVIE SIMILARITY FROM PLOT SUMMARIES

Ms. AISHWARYA LAKSHMI M
Department of Artificial Intelligence and Machine Learning Sri Shakthi Institute of Engineering andTechnology Coimbatore, India

Ms.KAVIPRIYA S
Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of Engineering and Technology
Coimbatore, India

Ms.S.NIVEDHA
Department of Artificial Intelligence and Machine Learning Sri Shakthi Institute of
Engineering and Technology
Coimbatore ,India

***Abstract*---**This project focuses on developing a Python application to analyze and measure the similarity between movie plot summaries. The goal is to provide a tool that can assist in identifying similarities between movies based on their storyline, enabling users to discover related movies or recommend similar ones.The project utilizes natural language processing (NLP) techniques, particularly text preprocessing, vectorization, and similarity metrics, to achieve its objectives. First, it preprocesses the plot summaries by removing stop words, punctuation, and performing stemming or lemmatization to normalize the text data. Next, it employs vectorization methods such as TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec or GloVe to convert the text into numerical representations suitable for similarity calculations.

INTRODUCTION

Movie similarity analysis based on plot summaries is a sophisticated domain that integrates narrative comprehension with computational methodologies to unveil thematic and structural congruences among films. By deploying advanced natural language processing (NLP) techniques such as tokenization, sentiment analysis, and word embeddings, in conjunction with machine learning (ML) algorithms like cosine similarity and hierarchical clustering, we can systematically quantify and categorize the semantic and contextual similarities embedded within movie plot synopses. In this project, we delve into the fascinating realm of machine learning and text analysis to create a robust system that can recommend movies similar to a given film by analyzing their plot summaries. Through the magic of programming, we aim to bring together movie enthusiasts and their next cinematic adventure. Our project begins with data collection, where we gather a diverse range of movie plot summaries from various sources. These summaries encapsulate the essence of each film, offering a glimpse into its narrative arc, characters, and themes. Once we have amassed this treasure trove

of textual data, we move on to the next phase.The heart of our project lies in natural language processing (NLP) techniques. We harness the power of libraries like NLTK and SpaCy to preprocess the plot summaries, stripping away noise and extracting meaningful features. This preprocessing step involves tokenization, lemmatization, and removing stop words, ensuring that our model focuses on the essence of the text.With the preprocessed data in hand, we venture into the realm of machine learning. Our model employs advanced algorithms such as TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity to quantify the similarity between movie plot summaries. TF-IDF helps us understand the importance of words in each summary, while cosine similarity measures the angle between two vectors, providing a numerical measure of their similarity.But our journey doesn't end there. We believe in creating an intuitive user experience, so we wrap our model in a user-friendly interface using libraries like Streamlit. This interface allows users to input a movie title and receive a list of recommendations based on the similarity of plot summaries.

LITERATURE REVIEW

The literature review on movie similarity from plot summaries documentation delves into the existing research and methodologies employed to analyze and measure similarity between movie plots based on textual data. It encompasses studies exploring various approaches such as natural language processing (NLP) techniques, machine learning algorithms, and semantic analysis to quantify and compare narrative structures, themes, and character interactions across movies. Scholars have utilized methods like TF-IDF, word embeddings, cosine similarity, and clustering algorithms to identify common patterns, genres, and motifs within plot summaries, shedding light on underlying storytelling conventions and audience preferences. Additionally, research in this area investigates the impact of data preprocessing steps, feature selection strategies, and similarity metrics on the accuracy and robustness of similarity assessments. By synthesizing insights from these studies, researchers gain a deeper understanding of the challenges, opportunities, and potential applications in content recommendation systems, genre classification, plagiarism detection, and narrative analysis in the realm of film studies and computational linguistics. The literature review thus provides a valuable foundation for designing and implementing effective approaches to movie similarity analysis from plot summaries, informed by established methodologies and emerging trends in the field.The Jaccard similarity index also called as the Jaccard similarity *coefficient* compares elements of two set to figure out which members are common and which are distinct. It is a measure of commonality between two sets of data between 0 an 1. The higher the number, the more similar are the two sets.

In this article, I will be finding out the Jaccard Similarity between **Genre** and **Director.**

PROPOSED SYSTEM

Our proposed system aims to create a comprehensive movie similarity platform using Python. We begin by collecting a diverse dataset of movie plot summaries from sources like IMDb and Wikipedia, ensuring coverage across genres and time periods. The data is then cleaned to remove noise and undergoes text preprocessing, including tokenization, lemmatization, and stop words removal, to prepare it for analysis.

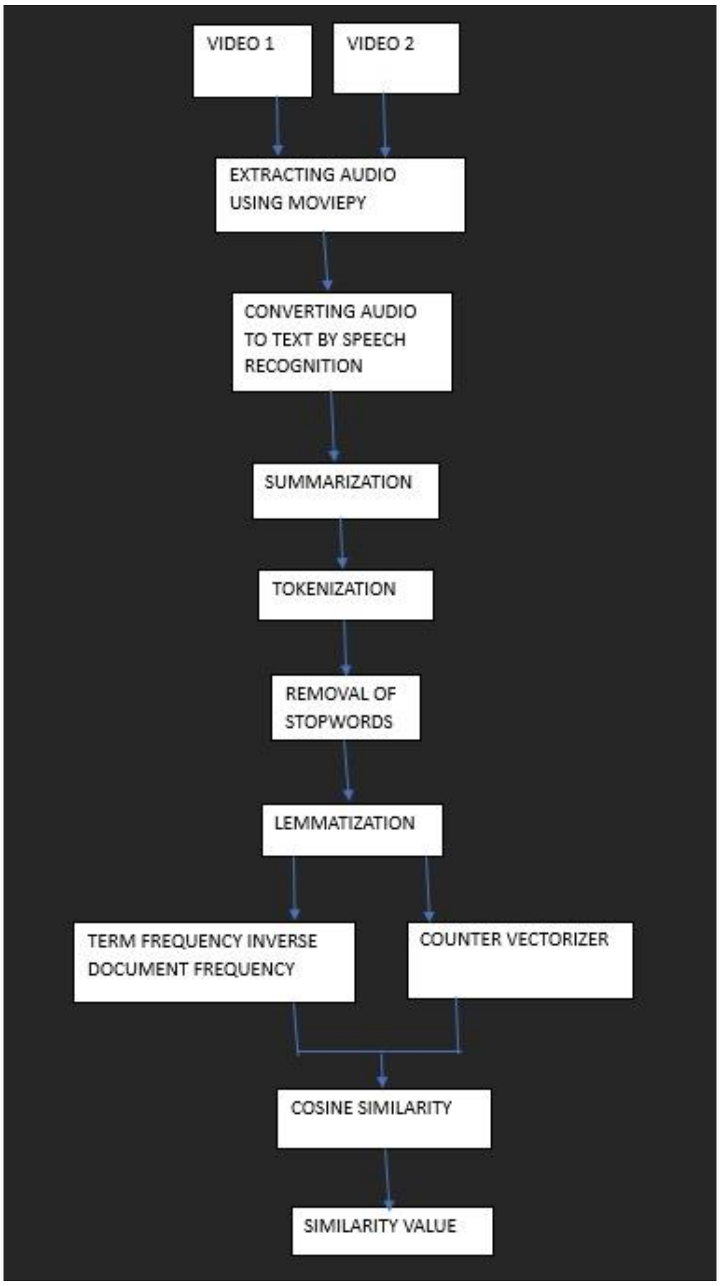
Next, we extract features from the plot summaries using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical vectors or word embeddings (e.g., Word2Vec, GloVe) to capture semantic relationships. These features serve as the basis for computing similarity measures between different movie plot summaries, with cosine similarity being a primary method used to quantify similarity.

For machine learning modeling, we employ libraries such as scikit-learn to develop a model that predicts movie similarities based on the extracted features. This model can be trained using labeled data if available or through unsupervised learning techniques in cases where labeled data is limited.

To provide a user-friendly experience, we create a web interface using Streamlit frameworks where users can input a movie title. Upon input, the system retrieves the corresponding plot summary, calculates similarities with other movies, and displays a list of recommended movies to the user. A feedback mechanism is integrated, allowing users to rate recommended movies and provide feedback, which is used to personalize recommendations and improve the system's accuracy over time.

Testing and validation are crucial steps, involving the evaluation of the system's accuracy, precision, recall, and other metrics using test datasets and cross-validation techniques. User feedback validation helps fine-tune the system parameters for optimization.

Figure 5.1: Proposed Model



METHODOLOGY:

Data Collection:Gather a diverse dataset of movie plot summaries from reliable sources such as IMDb, Wikipedia, or specialized movie databases.Ensure the dataset covers a wide range of genres, time periods, and popularity levels to capture diverse movie themes and narratives.

Data Preprocessing:Clean the raw data by removing HTML tags, special characters, and irrelevant information.Perform text normalization techniques like tokenization to split text into individual words or tokens.Apply techniques such as lemmatization or stemming to reduce words to their base forms.Remove stop words (common words like "the," "and," etc.) that don't contribute significantly to the meaning of the text.

Feature Extraction:Utilize the TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert text data into numerical vectors.TF-IDF emphasizes the importance of words in each plot summary by considering their frequency in a particular summary and across all summaries in the dataset.

Similarity Calculation:Compute cosine similarity between TF-IDF vectors of different movie plot summaries.Cosine similarity measures the cosine of the angle between two vectors and provides a numerical value indicating their similarity. Higher values indicate greater similarity.

Model Building:Implement a machine learning model or algorithm to calculate movie similarities based on cosine similarity scores.Consider using clustering algorithms (e.g., K-means clustering) to group similar movies together based on their plot summaries.

User Interface Development:Create a user-friendly interface using Flask or Streamlit to allow users to input a movie title.Upon receiving a movie title, the system should retrieve its plot summary, calculate similarities with other movies, and display a list of recommended movies to the user.

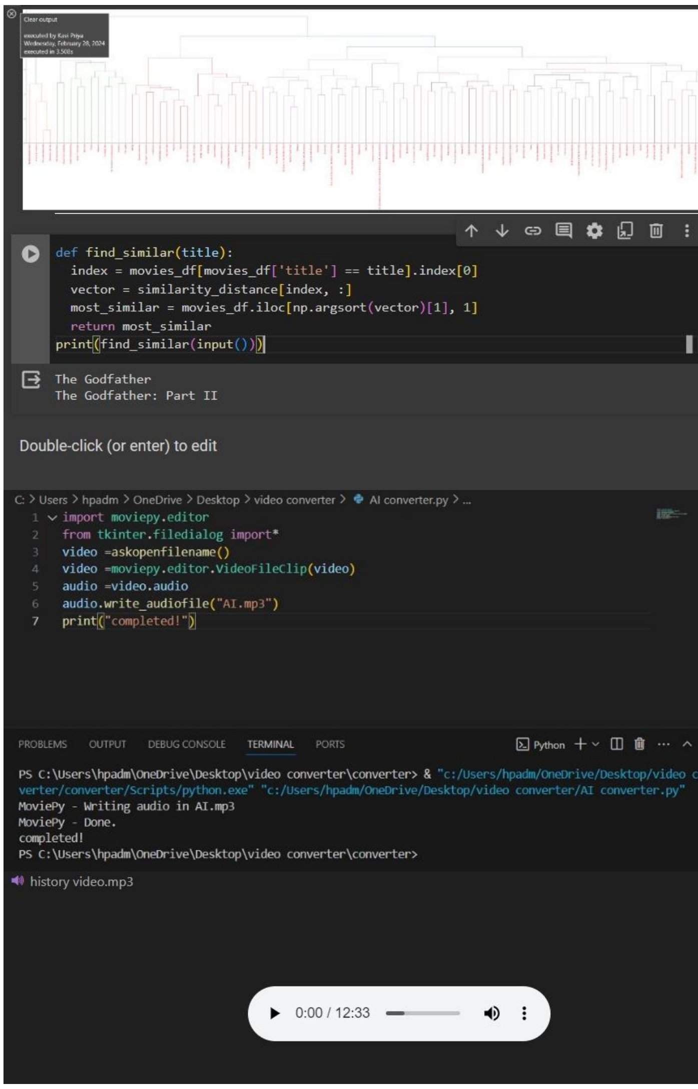
Testing and Validation:Evaluate the accuracy and performance of the movie similarity model using appropriate metrics and validation techniques.Conduct user testing to gather feedback on the recommendation system's effectiveness and user experience.

Optimization and Enhancement:

Fine-tune the model parameters, such as TF-IDF weights or clustering thresholds, to improve the quality of movie recommendations.Incorporate user feedback and iterate on the system to address any issues or limitations identified during testing.

Documentation and Deployment:Document the methodology, codebase, and system architecture comprehensively for future reference and collaboration.Deploy the movie similarity system on a suitable platform or server to make it accessible to users, ensuring scalability and performance optimization.

EXPERIMENTAL RESULT



```
def find_similar(title):
  index = movies_df[movies_df['title'] == title].index[0]
  vector = similarity_distance[index, :]
  most_similar = movies_df.iloc[np.argsort(vector)[1], 1]
  return most_similar
print(find_similar(input()))
```

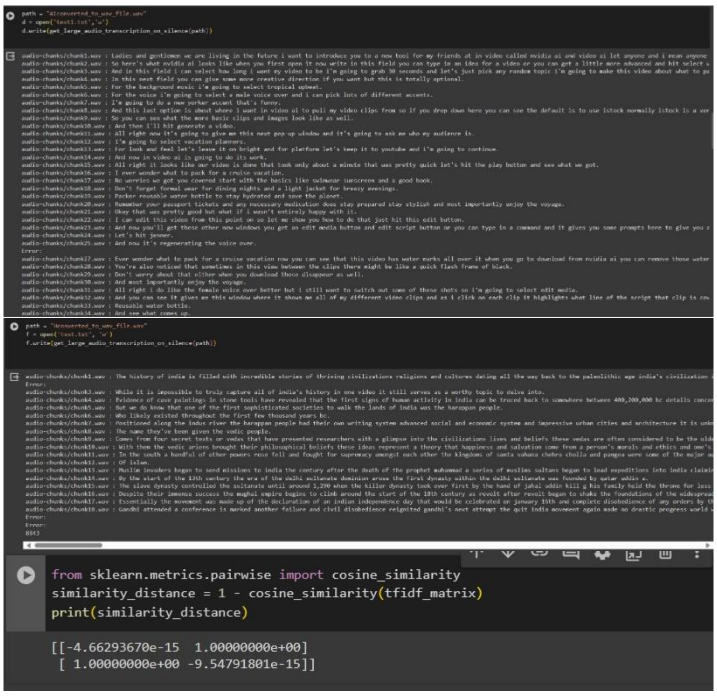
```
The Godfather
The Godfather: Part II
```

Double-click (or enter) to edit

C: > Users > hpadm > OneDrive > Desktop > video converter > AI converter.py > ...

```
import moviepy.editor
from tkinter.filedialog import*
video =askopenfilename()
video =moviepy.editor.VideoFileClip(video)
audio =video.audio
audio.write_audiofile("AI.mp3")
print("completed!")
```

```
PS C:\Users\hpadm\OneDrive\Desktop\video converter\converter> & "c:/Users/hpadm/OneDrive/Desktop/video c
verter/converter/Scripts/python.exe" "c:/Users/hpadm/OneDrive/Desktop/video converter/AI converter.py"
MoviePy - Writing audio in AI.mp3
MoviePy - Done.
completed!
PS C:\Users\hpadm\OneDrive\Desktop\video converter\converter>
```



```
path = "AIconverted_to_wav_file.wav"
d = open('test1.txt','w')
d.write(get_large_audio_transcription_on_silence(path))
```

```
audio-chunks/chunk1.wav : Ladies and gentlemen we are living in the future i want to introduce you to a new tool for my friends at in video called nvidia ai and video ai let anyone and i mean anyone
audio-chunks/chunk2.wav : So here's what nvidia ai looks like when you first open it now write in this field you can type in an idea for a video or you can get a little more advanced and hit select w
audio-chunks/chunk3.wav : And in this field i can select how long i want my video to be i'm going to grab 30 seconds and let's just pick any random topic i'm going to make this video about what to pa
audio-chunks/chunk4.wav : In this next field you can give some more creative direction if you want but this is totally optional.
audio-chunks/chunk5.wav : For the background music i'm going to select tropical upbeat.
audio-chunks/chunk6.wav : For the voice i'm going to select a male voice over and i can pick lots of different accents.
audio-chunks/chunk7.wav : I'm going to do a new yorker accent that's funny.
audio-chunks/chunk8.wav : And this last option is about where i want in video ai to pull my video clips from so if you drop down here you can see the default is to use istock normally istock is a ver
audio-chunks/chunk9.wav : So you can see what the more basic clips and images look like as well.
audio-chunks/chunk10.wav : And then i'll hit generate a video.
audio-chunks/chunk11.wav : All right now it's going to give me this next pop-up window and it's going to ask me who my audience is.
audio-chunks/chunk12.wav : I'm going to select vacation planners.
audio-chunks/chunk13.wav : For look and feel let's leave it on bright and for platform let's keep it to youtube and i'm going to continue.
audio-chunks/chunk14.wav : And now in video ai is going to do its work.
audio-chunks/chunk15.wav : All right it looks like our video is done that took only about a minute that was pretty quick let's hit the play button and see what we got.
audio-chunks/chunk16.wav : I ever wonder what to pack for a cruise vacation.
audio-chunks/chunk17.wav : No worries we got you covered start with the basics like swimwear sunscreen and a good book.
audio-chunks/chunk18.wav : Don't forget formal wear for dining nights and a light jacket for breezy evenings.
audio-chunks/chunk19.wav : Packer reusable water bottle to stay hydrated and save the planet.
audio-chunks/chunk20.wav : Remember your passport tickets and any necessary medication does stay prepared stay stylish and most importantly enjoy the voyage.
audio-chunks/chunk21.wav : Okay that was pretty good but what if i wasn't entirely happy with it.
audio-chunks/chunk22.wav : I can edit this video from this point on so let me show you how to do that just hit this edit button.
audio-chunks/chunk23.wav : And now you'll get these other new windows you get an edit media button and edit script button or you can type in a command and it gives you some prompts here to give you a
audio-chunks/chunk24.wav : Let's hit jenner.
audio-chunks/chunk25.wav : And now it's regenerating the voice over.
Error:
audio-chunks/chunk27.wav : Ever wonder what to pack for a cruise vacation now you can see that this video has water marks all over it when you go to download from nvidia ai you can remove those water
audio-chunks/chunk28.wav : You're also noticed that sometimes in this view between the clips there might be like a quick flash frame of black.
audio-chunks/chunk29.wav : Don't worry about that either when you download those disappear as well.
audio-chunks/chunk30.wav : And most importantly enjoy the voyage.
audio-chunks/chunk31.wav : All right i do like the female voice over better but i still want to switch out some of these shots so i'm going to select edit media.
audio-chunks/chunk32.wav : And you can see it gives me this window where it shows me all of my different video clips and as i click on each clip it highlights what line of the script that clip is cov
audio-chunks/chunk33.wav : Reusable water bottle.
audio-chunks/chunk34.wav : And see what comes up.
```

```
path = "Hconverted_to_wav_file.wav"
f = open('test.txt', 'w')
f.write(get_large_audio_transcription_on_silence(path))
```

```
audio-chunks/chunk1.wav : The history of india is filled with incredible stories of thriving civilizations religions and cultures dating all the way back to the paleolithic age india's civilization i
Error:
audio-chunks/chunk3.wav : While it is impossible to truly capture all of india's history in one video it still serves as a worthy topic to delve into.
audio-chunks/chunk4.wav : Evidence of cave paintings in stone tools have revealed that the first signs of human activity in india can be traced back to somewhere between 400,200,000 bc details concer
audio-chunks/chunk5.wav : But we do know that one of the first sophisticated societies to walk the lands of india was the harappan people.
audio-chunks/chunk6.wav : Who likely existed throughout the first few thousand years bc.
audio-chunks/chunk7.wav : Positioned along the indus river the harappan people had their own writing system advanced social and economic system and impressive urban cities and architecture it is unkn
audio-chunks/chunk8.wav : The name they've been given the vedic people.
audio-chunks/chunk9.wav : Comes from four secret texts or vedas that have presented researchers with a glimpse into the civilizations lives and beliefs these vedas are often considered to be the olde
audio-chunks/chunk10.wav : With them the vedic ariens brought their philosophical beliefs these ideas represent a theory that happiness and salvation come from a person's morals and ethics and one's
audio-chunks/chunk11.wav : In the south a handful of other powers rose fell and fought for supremacy amongst each other the kingdoms of santa vahana chebra cholla and pangea were some of the major a
audio-chunks/chunk12.wav : Of islam.
audio-chunks/chunk13.wav : Muslim invaders began to send missions to india the century after the death of the prophet muhammad a series of muslims sultans began to lead expeditions into india claimi
audio-chunks/chunk14.wav : By the start of the 13th century the era of the delhi sultanate dominion arose the first dynasty within the delhi sultanate was founded by qatar uddin a.
audio-chunks/chunk15.wav : The slave dynasty controlled the sultanate until around 1,290 when the killer dynasty took over first by the hand of jahal uddin kill g his family held the throne for less
audio-chunks/chunk16.wav : Despite their immense success the mughal empire begins to climb around the start of the 18th century as revolt after revolt began to shake the foundations of the widespread
audio-chunks/chunk17.wav : Essentially the movement was made up of the declaration of an indian independence day that would be celebrated on january 16th and complete disobedience of any orders by th
audio-chunks/chunk18.wav : Gandhi attended a conference is marked another failure and civil disobedience reignited gandhi's next attempt the quit india movement again made no drastic progress world w
Error:
Error:
8843
```

```python
from sklearn.metrics.pairwise import cosine_similarity
similarity_distance = 1 - cosine_similarity(tfidf_matrix)
print(similarity_distance)
```

```
[[-4.66293670e-15  1.00000000e+00]
 [ 1.00000000e+00 -9.54791801e-15]]
```

CONCLUSION

In conclusion, the movie similarity from plot summaries Python project showcases the power of natural language processing and machine learning in the domain of film recommendation systems. By leveraging advanced techniques such as semantic analysis, sentiment extraction, and recommendation algorithms, the project provides an efficient and personalized solution for users seeking movie suggestions based on plot similarities. The incorporation of interactive features, user feedback mechanisms, and data integration from external sources enhances the system's usability, accuracy, and relevance. Moving forward, continued research and development in this area can lead to even more sophisticated movie recommendation systems, catering to diverse user preferences and contributing to a seamless and enjoyable cinematic experience.

REFERENCES

1) https://github.com/geehaad/Find-Movie-Similarity-from-Plot-Summaries

2) https://towardsdatascience.com/using-nlp-to-find-similar-movies-based-on-plot-summaries-b1481a2ba49b

3) https://thepythoncode.com/article/extract-audio-from-video-in-python

4)https://medium.com/@kunalgupta4595/predicting-movie-genres-based-on-plot-summaries-bae646e70e04

5) https://medium.com/@kunalgupta4595/predicting-movie-genres-based-on-plot-summaries-bae646e70e04

6) https://www.geeksforgeeks.org/movie-recommender-based-on-plot-summary-using-tf-idf-vectorization-and-cosine-similarity/

7) https://github.com/Inyrkz/Movie-Similarity

8) https://www.geeksforgeeks.org/python-implementation-of-movie-recommender-system/

9) https://stackoverflow.com/questions/8897593/how-to-compute-the-similarity-between-two-text-documents

10) https://github.com/njugunaObi/Finding-movie-similarities-using-plot-summaries

11) https://www.geeksforgeeks.org/how-to-calculate-cosine-similarity-in-python/

12) https://www.geeksforgeeks.org/movie-recommender-based-on-plot-summary-using-tf-idf-vectorization-and-cosine-similarity/