CAST-A-PLOT: Cast Recommendation System

Rakshit Bang
IIIT-Bangalore, India
Rakshit.Bang@iiitb.ac.in

Rahul Jain
IIIT-Bangalore, India
Rahul.Jain@iiitb.ac.in

Anurag Singh Naruka
IIIT-Bangalore, India
Anurag.Naruka@iiitb.ac.in

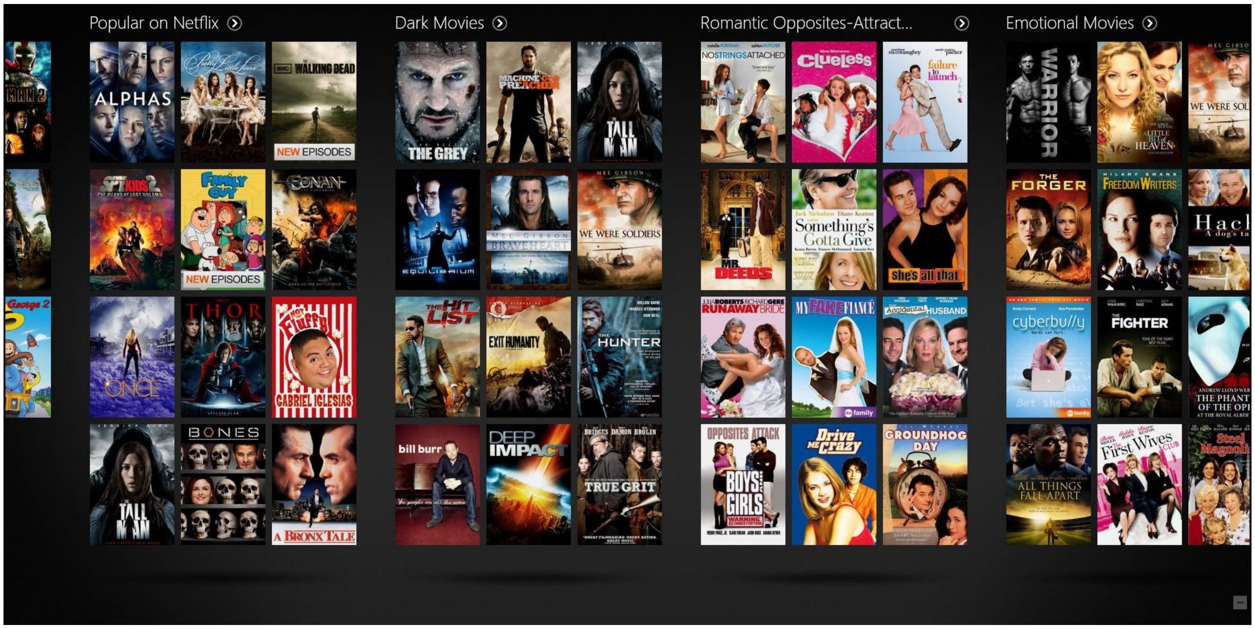


Figure 1. Cast-A-Plot: Cast Recommendation System

Abstract

In this paper, we propose Cast-A-Plot, a movie cast recommendation system that predicts suitable actors for movies based on movie plots and genres. We leverage BERT pre-trained models to obtain actor and plot embedding and use clustering and ranking techniques to recommend a list of actors. The proposed method demonstrates high accuracy in suggesting appropriate actors for movies with varying ratings.

Keywords: Cast Recommendation System, Neural Network, Transformers, Content-Based Filtering, Web Scraping, Glove Word Embedding, K-means, BERT Embedding, Actor Rating, NLP, Cosine Similarity

1 Introduction

Recommendation systems have become an increasingly important part of many industries, including the movie industry. Casting decisions are critical to the success of a movie, and traditionally, these decisions have been based on the intuition and experience of casting directors and producers. However, with recent advancements in natural language processing and machine learning, it is now possible to develop data-driven approaches for recommending suitable actors for movies.

In this paper, we introduce Cast-A-Plot, a movie cast recommendation system that utilizes BERT embedding and K-Means Clustering techniques. Our approach leverages information on the movie's plot and genres to suggest relevant actors for a given movie. We used different approaches to develop our recommendation system, ranging from basic methods such as using word embedding model such as Glove and cosine similarity to more advanced techniques that involved fine-tuning the BERT pre-trained models and using clustering, BERT model,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and embedding to recommend actors. We also explored using multi-label classification, although this approach falls outside the scope of traditional recommendation systems techniques.

To evaluate the effectiveness of our Cast-A-Plot system, we conducted a comprehensive evaluation using a diverse dataset that included information on various attributes of movies, such as their titles, genres, plots, and IMDb ratings. This evaluation enabled us to assess the accuracy and relevance of the recommended actors and determine the overall performance of Cast-A-Plot. We used different metrics, including ratios of good and bad predictions and a score, all of which we defined in the Evaluation section.

Overall, Cast-A-Plot has the potential to revolutionize the casting process in the movie industry by providing a more data-driven approach to actor selection. By leveraging machine learning and natural language processing, we can provide personalized and effective recommendations for movie casting that are more likely to result in successful movies. In the following sections, we will discuss our approach in detail and present our findings.

2 Dataset Creation

In order to develop an effective movie cast recommendation system, the CAST-A-PLOT project required a diverse range of high-quality datasets for testing and experimentation. To this end, we obtained four datasets.

Our first dataset, DATASET-1, was obtained directly from IMDb and contained information on every movie released after the 1960s. While this dataset provided extensive coverage, fetching it through the IMDb API was a slow process that took several days to complete. To overcome this issue, we created a temporary dataset, DATASET-2, consisting of recent movies released between 2018-2020, which enabled us to test various approaches and hypotheses.

To create our final optimal dataset, we compiled information on the top 88 actors of the 21st century and all their movies, resulting in a well-balanced and optimal-sized dataset consisting of approximately 2000 movies, known as DATASET-3. However, obtaining this information was not straightforward as the IMDb API did not provide an option to fetch all the movies given an actor's name. Therefore, we used web scraping to obtain the names of all the movies that these actors had worked in, and then used the API to retrieve the other details of the movies.

We also obtained a dataset from Kaggle, referred to as DATASET-4, which consisted of around 5000 movies and was originally designed for a movie recommendation system. However, this dataset did not include the plot of the movies, so we had to retrieve this information by using the list of movie names provided in the dataset.

Overall, the creation of these datasets required a significant amount of effort and attention to detail to ensure that we had a diverse and high-quality set of data to evaluate the effectiveness of the CastAPlot system.It should be noted that the results presented in this paper were obtained from DATASET-3.

Note: We are recommending one actor, specifically the protagonist for each datapoint in the dataset. Therefore, each data point in the dataset contains only main actor as their caste.

3 Pre-Processing

The pre-processing of DATASET-3 was relatively straightforward due to its carefully designed, well-balanced, and optimal size. We checked the dataset for null and duplicate values in the columns title, plot, actor, and genres. We found a few null values, which we chose to drop since they were minimal and would not significantly affect the system's performance.

We also noticed that the actor column contained actor names of varying lengths, including those with two words (first name and last name), three words (first name, middle name, and last name), or even longer. To ensure consistency in the actor names and facilitate finding the actor embedding from the BERT hidden layers, we concatenated the words in the name to form a single string for each actor.

Finally, we conducted an analysis using Kmeans clustering on the actor embedding distribution and found that Bruce Willis and Robert Duvall were outliers. As a result, we decided to drop all the data entries containing these actors. Overall, these pre-processing steps helped to refine and clean the dataset, enabling us to conduct robust and accurate analyses and evaluations of our CastA Plot system.

4 Approach

The approach to the Cast A Plot system can be divided into two main sections. In the first section, we generate high-level contextual embeddings for the actor names and movie plots. As plot, genres, and actors are not numeric data, we cannot use them directly to find similarity between them. Thus, we find the embedding of the actor name to recognize similar actors and the embedding of the plot of the movie to recognize similar plots.

In the second section, we use the similarities generated in the first section to find the top actor recommendations. Specifically, we find similar movies and actors and rank them using clustering and rating techniques. This approach allows us to streamline our analysis and

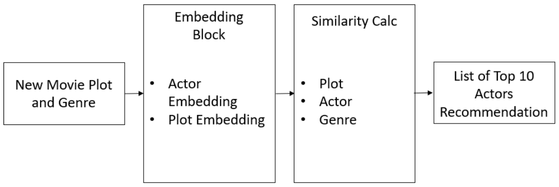


Figure 2. Approach Overview

evaluations and ensure that we are providing the most relevant and accurate recommendations for the protagonist role in each movie. In the following sections, we will discuss the two approaches we have used in more detail.

4.1 Approach-1

Our first approach to the CastAplot recommendation system is a modified version of the basic approach of finding the numeric embedding of the description of the datapoint and then finding similarity between them and ranking them based on the similarity score. Here are the steps we followed in our Approach-1:

- We use the plot and genres of each data point as the description of the datapoint.
- To use Glove Word embedding, we need to preprocess the plots by removing stop words, punctuations, converting to lowercase, and lemmatizing.
- Using the Glove Word Embeddings, we get the embedding for each token in the tokenized plot, and then average out all the token embeddings to get the final embedding of the plot.
- We one-hot-encode the genres to take care of them, as there are only a few unique genres.
- Using cosine similarity, we find the plot similarity score and genre similarity score of the given datapoint with all the datapoints in the dataset.
- For the given data point, we use the combined similarity score (CSS), which is

  $$CSS = 0.75 * plotSiml + 0.25 * genreSiml$$

  with all the movies in the dataset. This gives us the top ten similar movies to the given data point.
- We recommend the ten actors of these top ten movies based on the combined similarity score, following the same ranking approach.

Overall, this approach involves preprocessing the plot, finding the Glove Word Embedding, and using cosine similarity to find the similarity score of the given datapoint with all the datapoints in the dataset. Then, we use the combined similarity score to recommend similar movies and actors. In the following sections, we will discuss the results of this approach.

4.2 Performance and Analysis

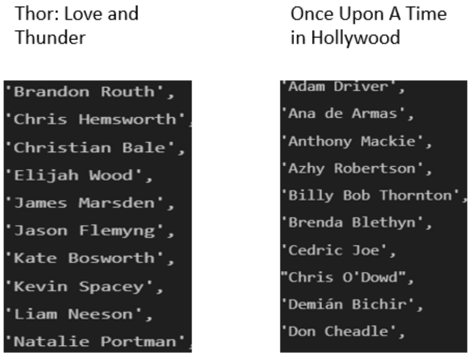
Our Approach-1 for the Cast-A-Plot recommendation system involved generating embeddings for the plot and genres of each movie to find similar movies and actors. However, we faced several challenges in achieving satisfactory results with this approach. Here are the issues we encountered and the steps we took to address them:

- **Embedding Generation:** We used Glove Word Embedding to generate embeddings for the plot and genres, but this resulted in contextless embeddings that did not take into account the context of the words in the plot. This led to poor performance in recommending actors for random movies whose prequel was not present in the dataset.
- **Length of Plot:** Different plots have different lengths, which resulted in high variance and made it difficult to effectively use the mean of the embeddings.
- **Rating of the Actor:** We tried incorporating the IMDb rating of the movie and the rating of the actor as a third parameter, but this did not improve the performance of the recommendation system.
- **Neural Network:** We attempted to train a neural network to learn the optimal weights of plot similarity and genre similarity, but this did not significantly improve the system's performance.

Despite these attempts to improve the performance of Approach-1, we found that it was not satisfactory for recommending actors for random movies. However, the approach worked well for recommending actors for sequels of movies already present in the dataset, as the plot of both movies would have similar words and context.

For example, Approach-1 recommended Chris Hemsworth for the movie Thor: Love and Thunder, as the plot of the movie is a continuation of the story from Thor: The Ragnarok, which is present in our dataset. However, for movies with no prequel in the dataset, such as Once Upon A Time in Hollywood, the approach did not recommend top actors such as Leonardo DiCaprio and Brad Pitt.

Overall, the main issue with Approach-1 was the contextless embeddings generated using Glove Word Embedding. This led to poor performance in recommending actors for random movies, as the embeddings did not capture the context of the words in the plot. In the following section, we will discuss our second approach, which addressed these issues and led to better results in recommending actors for random movies.



Thor: Love and Thunder

```
'Brandon Routh',
'Chris Hemsworth',
'Christian Bale',
'Elijah Wood',
'James Marsden',
'Jason Flemyng',
'Kate Bosworth',
'Kevin Spacey',
'Liam Neeson',
'Natalie Portman',
```

Once Upon A Time in Hollywood

```
'Adam Driver',
'Ana de Armas',
'Anthony Mackie',
'Azhy Robertson',
'Billy Bob Thornton',
'Brenda Blethyn',
'Cedric Joe',
"Chris O'Dowd",
'Demián Bichir',
'Don Cheadle',
```

Figure 3. Approach-1: Actor Recommendation
A: Thor: Love and Thunder
B: Once Upon A Time in Hollywood

4.3 Approach-2

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning model developed by Google that is designed to generate high-quality contextualized word embeddings. Using BERT we will first find the Actor Embedding.

4.3.1 Actor Embedding.

- **Actors Genres Dataset (Input):** The input to the BERT model is a dataset containing the actors' names and the genres they have worked in. Each datapoint is represented as a sentence that includes the actor's name and the genres separated by commas. For example, "Leonardo DiCaprio in genres drama, romance, tragedy".
- **BERT Model (Fine-tuned on Input):** In this step, the BERT model is fine-tuned on the input dataset. The model learns to assign attention scores to each genre based on the actor's experience in those genres. This is done using the attention mechanisms in the transformers of the BERT model, which focus on specific words or phrases in a sequence.
- **Attention Scores:** The attention scores are calculated by comparing each word in the sequence to all the other words and assigning a score based on their similarity. Higher attention scores are given to words that are more relevant to the actor's experience in the genres, while lower scores are given to less relevant words.
- **Actor Embeddings (Output):** The actor embeddings are extracted from the hidden layers of the fine-tuned BERT model. These embeddings capture the actor's strengths and characteristics based on the attention scores assigned to the genres they have worked in. The resulting embeddings can be used to recommend actors for specific roles in the CastAplot recommendation system.

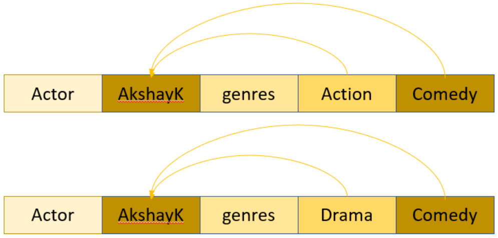


Figure 4. Actor Embedding Learning - BERT

The inner workings of the BERT model involve a multi-layer transformer architecture that processes the input data in parallel, rather than sequentially. During fine-tuning, the model learns to generate attention scores for each word in the input sentence. These attention scores are used to weight the importance of each word in the sentence, allowing the model to focus on the most relevant words (i.e., the genres) when generating the actor embeddings.

This approach is novel and great because it leverages the power of BERT's attention mechanisms to focus on the actor's experience in specific genres. By fine-tuning the model on a dataset of actors and their associated genres, the model learns to generate embeddings that are primarily driven by the genres. This approach differs from traditional methods that rely on basic similarity metrics or collaborative filtering. This approach has the potential to revolutionize the way actors are recommended for roles in movies and other productions.

4.3.2 Plot Embedding.

- We use the DistilBERT model from the transformers library to generate embeddings for the plot of each movie. The process involves several steps:
- We import the DistilBERT model and tokenizer from the transformers library and specify the pre-trained weights to use.
- We retrieve the plot data from the dataset and tokenize it using the BERT tokenizer. Tokenization involves breaking down the plot text into individual words or subwords and mapping them to unique integer IDs.
- We apply padding to the tokenized inputs to ensure that all sequences are of the same length. The padding function adds zeros to the end of shorter sequences to match the length of the longest sequence.

- We create a mask to indicate which tokens are actual text and which are padding. The mask is used to exclude padding tokens from the attention mechanism in the BERT model.
- We pass the tokenized and padded inputs and the mask through the DistilBERT model to get the final hidden states. We extract the embeddings for each plot by taking the first token embedding for each sequence.

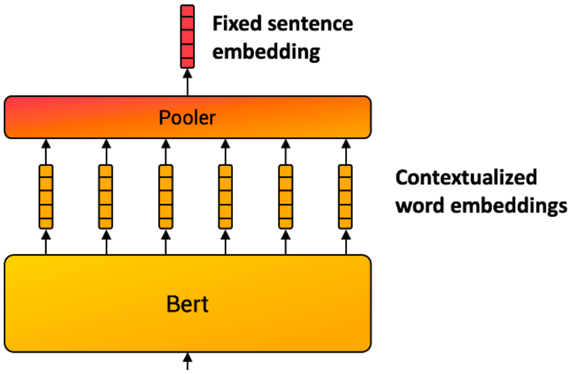


Figure 5. Plot Embedding Learning - BERT

The resulting plot embeddings are high-quality and informative because they capture the context of the words in the plot text. BERT is a transformer-based model that is designed to generate high-quality contextualized word embeddings by considering the surrounding words in a sentence or sequence. This allows it to capture the meaning and context of words more effectively than traditional methods like bag-of-words or TF-IDF.

Compared to the plot embeddings obtained in the first approach, which used a simple averaging method, the BERT embeddings are more informative and impactful. The BERT embeddings capture the context and meaning of the words in the plot text, which allows them to better represent the overall plot of the movie. This makes them more effective for recommending similar movies based on plot similarities.

4.3.3 Ranking Actors. In our CastAplot recommendation system, we use two ranking mechanisms to rank the top 10 actors based on their suitability for specific roles. These mechanisms are the Average Actor Rating and the clustering of actor embeddings. Here are the details of each mechanism:

1. **Average Actor Rating:** This mechanism involves calculating the average rating of all the movies in the dataset in which the actor had worked. This gives us an overall high-level knowledge of how well the actor can perform when selected for a particular task. The actors with higher average ratings are considered to be more suitable for specific roles. To calculate the Average Actor Rating, we first retrieve the ratings data from the dataset and filter it to only include movies in which the actor had worked. We then calculate the average rating for each actor and use this value to rank the actors.
2. **Clustering of Actor Embeddings:** This mechanism involves clustering the actor embeddings based on their similarity. We use unsupervised learning techniques like K-means clustering to group the actors based on their embeddings. The actors belonging to the cluster with the maximum frequency in the top 10 are placed at the top, and the actors with lower frequency clusters are placed lower. To cluster the actor embeddings, we first generate the embeddings for each actor using the BERT model. We then use K-means clustering to group the actors based on their embeddings. Once the clusters are formed, we count the number of actors in each cluster that are in the top 10 and place the actors in descending order based on the frequency of their clusters.

Both of these ranking mechanisms are useful in determining the suitability of actors for specific roles. The Average Actor Rating provides a high-level overview of the actor's performance in previous roles, and the clustering of actor embeddings provides a more fine-grained analysis of the actor's strengths and characteristics based on the movies they have worked in. By combining the Average Actor Rating and the clustering of actor embeddings, we can generate more informative and impactful recommendations for the CastAplot recommendation system.

4.3.4 Final Implementation. The final implementation of CastAplot recommendation system involves several steps:

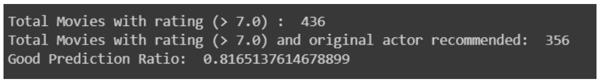
- Pre-processing the data - This involves cleaning and formatting the movie dataset to obtain the necessary information for the recommendation system.
- Fine-tuning BERT - We fine-tune the BERT model to generate highly contextualized actor embeddings based on the genres and characteristics of the actor. This allows us to capture the unique features and strengths of each actor to recommend them for specific roles.
- Finding contextual plot embeddings - We use the BERT model to generate contextualized embeddings for the plot of each movie. These embeddings capture the meaning and context of the words in the plot text, allowing us to find similar movies based on plot similarities.

- Finding the top 10 similar movies - Based on the contextual plot embeddings, we use cosine similarity to find the top 10 movies that are most similar to the input movie.
- Finding the top 10 actors - From the top 10 similar movies, we extract the top 10 actors who have worked in those movies. These actors are considered to be suitable for the input movie based on their performance in similar movies.
- Ranking the actors - We rank the top 10 actors using a two-step mechanism. First, we cluster the actors based on their embeddings using unsupervised learning techniques like K-means clustering. Second, we rank the actors within each cluster based on their Average Actor Rating, which is the average rating of all the movies in the dataset in which the actor had worked.
- Final recommendations - We obtain the final top 10 actor recommendations for the input movie based on the rankings generated in step 6.

4.3.5 Performance and Analysis. The evaluation metric for the CastAplot project is designed to measure the effectiveness of the recommendation system in recommending actors for specific roles based on the success of the movies they have worked in. The metric is based on the IMDb movie rating, which is a widely recognized measure of a movie's popularity and critical acclaim.

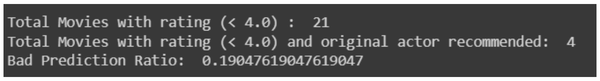
The evaluation metric distinguishes between successful and unsuccessful movies based on their IMDb rating. For successful movies with IMDb rating greater than 7.0, the recommendation system is expected to recommend the same actor for the role, as their performance in the movie was considered to be good. Similarly, for unsuccessful movies with IMDb rating less than 4.0, the recommendation system is not expected to recommend the same actor, as their performance in the movie was considered to be poor.

To measure the effectiveness of the recommendation system, two ratios are defined: the Good Prediction Ratio (GP) and the Bad Prediction Ratio (BP). The GP is calculated as the ratio of the total number of movies with IMDb rating greater than 7.0 and the original actor in the top 10 recommended actors, to the total number of movies with IMDb rating greater than 7.0. The BP is calculated as the ratio of the total number of movies with IMDb rating less than 4.0 and the original actor in the top 10 recommended actors, to the total number of movies with IMDb rating less than 4.0. We get the following results on the DATASET-3 using approach 2:



```
Total Movies with rating (> 7.0) :  436
Total Movies with rating (> 7.0) and original actor recommended:  356
Good Prediction Ratio:  0.8165137614678899
```

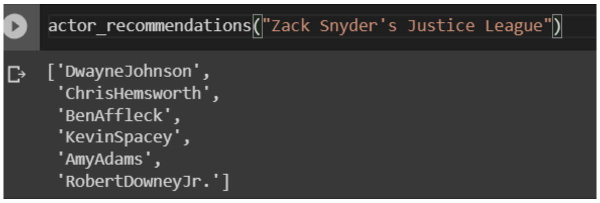
Figure 6. Good Prediction Ratio : Approach-2



```
Total Movies with rating (< 4.0) :  21
Total Movies with rating (< 4.0) and original actor recommended:  4
Bad Prediction Ratio:  0.19047619047619047
```

Figure 7. Bad Prediction Ratio : Approach-2

A good recommendation system is expected to have a high GP and a low BP. This means that the system is able to recommend the same actor for successful movies, while avoiding recommending the same actor for unsuccessful movies. By using this evaluation metric, the CastAplot project can measure the effectiveness of the recommendation system in generating informative and impactful recommendations for actors based on their performance in similar movies. The system build using approach 2 on the DATASET-3 gave **GP = 0.8165** and **BP = 0.1904**. As we can see from the results, our system shows excellent performance !
Below are the actor recommendations for some famous movies:



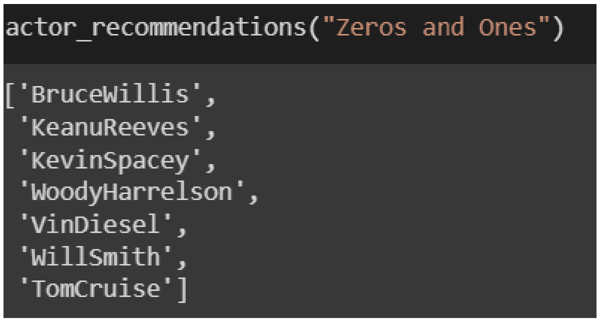
```
actor_recommendations("Zack Snyder's Justice League")
```

```
['DwayneJohnson',
 'ChrisHemsworth',
 'BenAffleck',
 'KevinSpacey',
 'AmyAdams',
 'RobertDowneyJr.']
```

Figure 8. Zack Snyder's Justice League: Recommended Actors

We have Ben Affleck for Zack Synders's Justice League. Infact, we consider the recommendation for the Zack Synders's Justice League, we can see Dwayne Johnson, Chris Hemmsworth, Robert Downey Jr., Kevin Spacey this all are very highly suitable for the role. Dwayne Johnson is known for his muscular physique and action-oriented roles, which may make him a suitable choice for an action movie or superhero film. Similarly, Chris Hemsworth has played the role of Thor in the Marvel Cinematic Universe, demonstrating his ability to handle superhero roles. Robert Downey Jr. has also played a superhero role as Iron Man, and has a wide range of acting experience that could make him a strong choice for such types of role. Kevin Spacey has a long history of acclaimed performances in film and television, and could bring a level of gravitas and skill to such role. Thus, it shows our system is successfully taking in the context and characteristics of the movie and its character.

The movie "Zeros and Ones" has a low IMDB rating of 3.3 and does not have any famous actors with genres such as action, thriller, or war. This suggests that the



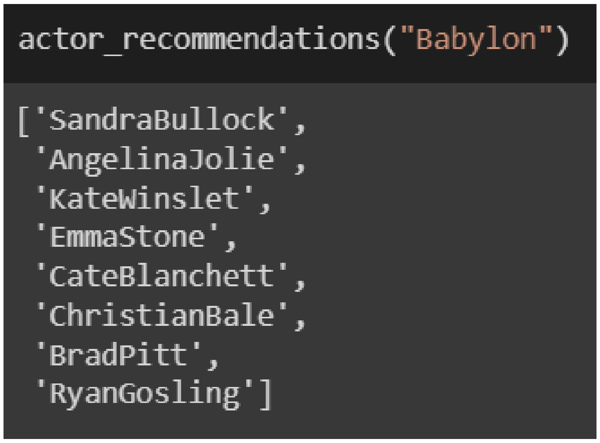
```
actor_recommendations("Zeros and Ones")

['BruceWillis',
 'KeanuReeves',
 'KevinSpacey',
 'WoodyHarrelson',
 'VinDiesel',
 'WillSmith',
 'TomCruise']
```

Figure 9. Zeros and Ones: Recommended Actors

movie was not successful and that the actor was not well-suited for their role. As expected, the original actor of this movie is not included in the recommended list of actors by the Cast-A-Plot system. The recommended actors by the system are known for their ability to handle the challenges of action, thriller, and war movies and have built up a strong reputation with audiences and critics alike. Their experience and versatility make them ideal choices for movies in these genres.

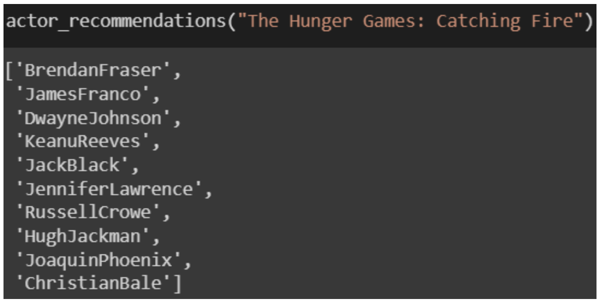
The Cast-A-Plot system has recommended Brad Pitt for the movie "Babylon" and Jennifer Lawrence for "The Hunger Games," both of which are well-suited for their respective roles. Interestingly, the system has recommended similar actors for both the superhero movie "Zack Snyder's Justice League" and "The Hunger Games". Notably, the system has learned that "The Hunger Games" requires a female lead, and has recommended Jennifer Lawrence as the best fit for this role. However, the system did not recommend any female lead for "Zack Snyder's Justice League." as the super hero is male in the movie's story-line. This highlights the implicit learning of key characteristics for roles, such as gender, to ensure that the system provides the most appropriate recommendations.



```
actor_recommendations("Babylon")

['SandraBullock',
 'AngelinaJolie',
 'KateWinslet',
 'EmmaStone',
 'CateBlanchett',
 'ChristianBale',
 'BradPitt',
 'RyanGosling']
```

Figure 10. Babylon: Recommended Actors



```
actor_recommendations("The Hunger Games: Catching Fire")

['BrendanFraser',
 'JamesFranco',
 'DwayneJohnson',
 'KeanuReeves',
 'JackBlack',
 'JenniferLawrence',
 'RussellCrowe',
 'HughJackman',
 'JoaquinPhoenix',
 'ChristianBale']
```

Figure 11. Hunger Games: Recommended Actors

5 Future Scope

The proposed future scope of the CastAplot project involves several exciting ideas that could significantly enhance the recommendation system's effectiveness and accuracy.

Firstly, the project aims to expand the recommendation system to predict the entire cast of a movie instead of just the protagonist actor. This will require incorporating additional data about the other actors and their performances in similar movies to generate relevant recommendations for the entire cast.

Secondly, the future scope of the project involves incorporating more personal information about the actors in the fine-tuning sentence, such as their physical build, sense of humor, voice quality, age, gender, awards, other hobbies, and experience. This will allow the system to generate more personalized embeddings of the actor, which will denote the actor as a whole, not just based on the genres they have worked in.

Thirdly, the project aims to incorporate the storyline of the movie and find the characteristic qualities of the characters to recommend actors who would best suit these qualities. This will involve finding similarities between the characters the actor has played in different movies and the character that needs to be played in the new movie.