USENIX Symposium on Usable Privacy and Security (SOUPS) 2020. August 9 -- 11, 2020, Virtual Conference.

.

BingeBot: An LLM for Finding the Perfect Netflix Movie

Reagan Sanz, Grant Alderson, Makenzie Johnson, Faith Chernowski
Department of Computer Science, University of Tennessee, Knoxville

Abstract

The creation of BingeBot was centered around the idea of providing users with an easy way to decide what to watch without spending hours flipping through trailers. With that goal in mind—and the knowledge that a Large Language Model (LLM) would be the best way to achieve it—the next challenge was to find or create a dataset that provided relevant data on the subject. Once the dataset was created, the next steps were to select a specific model, train it, and determine evaluation metrics that accurately represented the model's performance.

During the development of BingeBot, several challenges were encountered, including the difficulty of finding a comprehensive dataset, selecting a model that met the project's needs, and ensuring the model could correctly answer complex queries. These challenges were addressed, resulting in a functional chatbot capable of recommending movies based on user specifications—on par with similar real-world products

1. Introduction

Watching movies is a comforting and widely shared activity, enjoyed by billions of people around the world. In 2021, approximately 82% of American adults watched movies through online subscription services, with 46% doing so daily or several times a week[4]. With so many people engaging in this activity regularly, the amount of time spent merely deciding what to watch becomes a significant issue—one often overlooked.

Although many users may disregard this time, American streaming service subscribers spend an average of 110 hours per year just choosing what to watch[3]. This raises an important question: how can we streamline the process of choosing entertainment content?

1.1. Importance of Streamlining Content Decisions

While it may seem trivial, the time spent deciding what to watch could be better used on more productive or meaningful activities. For some, especially those with limited leisure time, optimizing this process could be highly beneficial.

Even though the average time spent searching is 110 hours per year, this figure varies among individuals[3]. Nevertheless, devoting nearly five full days annually to this task is significant—and most people would likely prefer to use that time more enjoyably or effectively.

Streamlining movie selection would also benefit streaming services by increasing user watch time. Users who spend less time browsing and more time watching are likely to enjoy the platform more, potentially leading to higher subscription rates. Additionally, increased viewing time means more opportunities for ad impressions and greater exposure for original content—ultimately boosting platform revenue and engagement. Both users and streaming platforms stand to gain significantly from a more efficient method of content recommendation.

1.2. Approach

With the project's motivation and objectives clearly defined, the next step is implementation. The project aims to replicate a traditionally human-driven task—choosing what to watch—using a Large Language Model.

Before a model can be used, however, it must be trained on data similar to the task it's intended to perform. After obtaining or building a dataset that captures the necessary information, an appropriate model will be selected based on the project's specifications. This model will then be trained using the dataset and thoroughly evaluated to assess its strengths and identify any shortcomings.

After initial evaluation, the model will undergo further training to address any performance gaps. The ultimate goal is to develop a movie recommendation chatbot that can

generate personalized suggestions based on user preferences, delivering results comparable in quality to existing commercial solutions.

2. Methods: Goal

The goal of this implementation is to create an LLM to help users find a movie on Netflix. It should be used to assist users by giving them relevant recommendations and answer questions on current Netflix movies. This is currently achieved through RAG training on relevant datasets containing information on various Netflix movies.

A total of 10 different files are used with RAG, each of these contain different, relevant information on netflix movies. The base LLM is Gemini 2.0 Flash, which was chosen for its efficiency and solid base knowledge. It has a high enough number of parameters for strong language processing while also having room for improvement in fine-tuning and RAG on Netflix data.

2.1. Datasets Collected

The first and second dataset [1][2] were collected from Kaggle, and these were stored as a .csv of information on Netflix titles (with around 6,129 and 3,720 entries, respectively). These contained the following information for each movie:

- Number
- Type (TV Series or Movie)
- Title
- Director
- Cast
- Country Produced In
- Date Added on Netflix
- Release Year
- Age Rating
- Duration
- Genre
- Description
- IMDB Rating* (Just on dataset [2])

This dataset contained additional information that was not needed. Given that the main focus of BingeBot is to suggest Netflix movies, this table was sorted by type (TV Series or Movie) and TV series were deleted from the table. This left 6,129 movies left in the dataset. The columns for both "number" and "type" were removed as well so only relevant information was included.

After our initial testing on these two datasets, it was discovered that the model struggled with certain categories (specifically, movies released recently, information on award winning movies, and accurate IMDB ratings). Therefore, additional datasets were collected. This includes:

- A csv file containing Netflix movies from 2024.
- Two pdfs of the IMDB list of Netflix movies released in April 2025 and March 2025
- A csv file containing information on Netflix movies released in 2025.
- A csv containing the Top 100 Rated Netflix movies
- A csv containing Rotten Tomatoes Movie Ratings
- A csv file containing Oscar winning Netflix Movies.

A few edits were made to these files, such as altering the Oscar movie dataset and Rotten Tomatoes dataset to *only* contain information on Netflix movies, specifically. This was done by comparing the movie titles to those in the Netflix datasets and removing movies that were not available on the platform. Given that our LLM is focused on Netflix movie data, specifically, TV shows and non-Netflix movies were removed from the datasets to improve the speed of RAG training.

The combination of these chosen datasets gave context to the model on recently released Netflix movies from the past year, updated user ratings, and a list of award winning and most popular movies on the platform. In addition, it covered most information on the movies themselves, from cast to runtime to genre.

2.2. RAG Training

```
Title: Spider-Man: Into the Spider-Verse
Director: Peter Ramsey, Rodney Rothman, Bob Persichetti
Cast: Shameik Moore, Jake Johnson, Hailee Steinfeld, Brian Tyree Henry,
Kathryn Hahn, Liev Schreiber, Kimiko Glenn, Nicolas Cage, John Mulaney
Runtime: 117 min
Country Created In: United States
Date Added to Netflix: 26-Jun-19
Year it was Released: 2018
Age Rating: PG
Genre(s): Action & Adventure, Comedies
Audience Rating on IMDB: nan
Description: After being bitten by a radioactive spider, Brooklyn teen
slinging from his alternate-dimension counterparts.
```

Figure 1: Example of a Movie Entry in the Text File to be RAG trained

The method of training our Gemini model involves RAG training on the csv, txt, and pdf files from above. The main datasets that include Netflix containing movie information can be seen above.

This file was then loaded into a script for RAG training.

Our implementation of RAG involved utilizing LangChain and RecursiveCharacterTextSplitter to ensure that the data provided was split into chunks and trained properly on the

model. The model was prompted to answer the question to the best of its ability with the provided information from the dataset, but not to inform the user if the LLM does not know the correct answer. The full code and datasets used can be seen on the most recently updated BingeBot GitHub (https://github.com/ReaganSanz/BingeBot).

2.3 System Pipeline

To help visualize how BingeBot processes user queries, we created a simple diagram of the system's architecture. The pipeline starts with the user entering a natural language prompt. This input is passed through our RAG setup, which pulls relevant context from the Netflix-specific datasets we collected. That context is then used to guide the Gemini 2.0 Flash model, which generates a personalized movie recommendation. The result is returned to the user in a conversational format. This flow is shown in Figure 2.

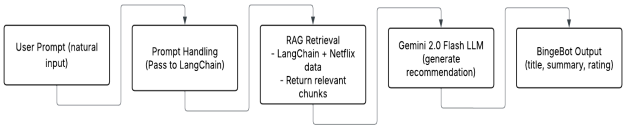

Figure 2: BingeBot Query-to-Output Workflow

3. Methods: Evaluation

3.1. Testing Setup

To evaluate how well BingeBot performs, we created a variety of test prompts that reflect the kinds of questions real users would ask when trying to decide what to watch. These prompts included requests for specific types of movies—such as genres, actors, languages, or ratings—as well as broader questions like asking for movies similar to one the user has already seen and liked. For example, we tested queries like:

- "Can you give me a recommendation for a Netflix movie that is a French sports film?"
- "I want to watch a movie on Netflix with Hugh Jackman. Can you give some suggestions with descriptions and their age ratings?"
- "I liked Les Misérables. What's something similar I might enjoy?"

We used these sample prompts to probe how well the model could pull from the RAG-trained dataset and generate useful, personalized responses. These tests also helped us spot gaps in how the model was understanding or retrieving movie data. Since Netflix's library is constantly changing and includes a wide variety of content, we chose prompts that pushed the model to search across different categories and be as flexible as possible with its output.

3.2 Evaluation Criteria

We scored the model's responses using a rubric we developed as a team. The main areas we focused on were:

- Relevance – Did the recommendation make sense for what the user asked? For example, if a user asked for a romance with time travel, did the response include movies that fit both?
- Accuracy – Was the information correct based on our dataset? We cross-checked things like actor names, release years, genres, and ratings with our source data.
- Completeness – Did the model give enough information to be useful? This included movie summaries, runtime, country of origin, and age rating, depending on what the user asked for.
- Clarity – Was the response easy to understand and well-written? Since this is a conversational tool, clear language and tone are important.

Each team member reviewed outputs and contributed to the scoring and discussion, which helped us align on what a "good" response looked like. This kind of peer review was important because it helped balance technical performance with user experience—something that automated metrics alone can't fully capture.

3.3. Qualitative Observations from Evaluation

As we reviewed BingeBot's responses during testing, we noticed several patterns that helped us better understand how the model was performing. For example, when asked "I liked Inception, what should I watch?", the model recommended movies like Tenet and Source Code. These were strong suggestions because they matched the genre and tone of the original movie, and the response included useful details like descriptions and age ratings.

During evaluation, each team member reviewed responses independently. Most of the time, we agreed on whether a recommendation was relevant and accurate. However, there were some differences in how we judged completeness. Some reviewers expected the model to include extra information like runtime or release year, while others were satisfied with just a summary and rating. This showed us that evaluation can sometimes be subjective, especially for movie recommendations.

We also realized that even accurate answers might not feel helpful to every user. For example, if the model suggests an award-winning drama in response to a prompt about "romantic comedies," it may technically be a good movie but does not match the user's intent. These cases reminded

us that understanding user expectations is just as important as having correct data.

4. Results

After training and testing, our model can accurately recommend a movie based on the prompt given by the user.

4.1. Foreign Movies

Foreign movies on the US version of Netflix make up 45% of the movie library. Users looking to explore the art of film outside of an American perspective have a lot of options to choose from. When testing this out with the Gemini model, we learned that it could only state that the movie "is an action film" and couldn't tell if the movie is on Netflix. After asking again with the RAG model, the model was able to give information about the film and could tell the user what region the film was based in.

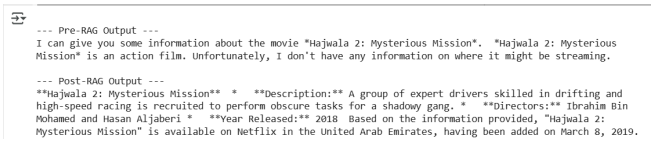

Figure 3: RAG Response to a Prompt About the Foreign Hajwala Movies

4.2. Recent Movies

BingeBot is RAG trained on data from recent Netflix movies from 2024 and early 2025. When the original Gemini model is given a prompt asking about "Plankton the Movie", it is unable to give information about this movie that released in March 7th, 2025. The Pre-RAG output states "*Ah, 'Plankton the Movie'! That's a fun one... except it doesn't actually exist!*". After being giving the RAG datasets for context, is it able to successfully give information about this new Netflix Movie: "*Here's information about the movie "Plankton: The Movie," based on the provided text: * **Title:** Plankton: The Movie * **Release Date:** March 7, 2025 * **Director(s):** Dave Needham * **Cast Highlights:** Mr. Lawrence...*" After RAG, it is able to give the movie release date, directors, cast, and a description of the movie.

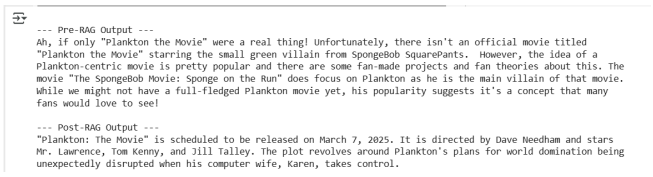

Figure 4: Output Before and After RAG Training for Plankton Prompt

4.3. Movie Ratings

Movie Ratings can help a user determine whether they would be interested in watching a particular movie. The original Geminai model couldn't give an accurate rating when prompted. For example, when asked, *"What is the rating for Mr. Peabody and Sherman?"*, it answered, *"Mr. Peabody and Sherman has the following ratings: IMBD: 6.8/10, Rotten Tomatoes: 79%, Common Sense Media: 3/5."* While the model was able to answer, it wasn't an accurate one. The real results are 81% Rotten Tomatoes, 6.7/10 IMBD rating, 3/5 from Common Sense Media. Not far off from the real ratings, but we want our model to be as accurate as possible. The discrepancy is most likely due to the LLM not being updated in real-time like the RAG model. The RAG model was able to give the correct ratings for the movie and included the PG rating to drive it home.

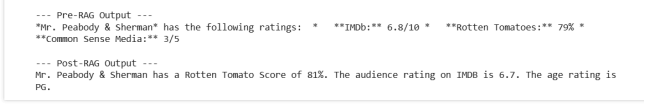

Figure 5: Accuracy Comparison of Model Responses to a Rating Query Before and After RAG Training.

4.4. Oscar Awards

Typically, movies that have won awards draw more people in. Maybe the user is interested in knowing more about if their favorite movie won an award. Lucky for them, BingeBot is able to tell which award said movie won, what year the award was given, the director, and recommend movies that have won an Oscar. It can even tell if a movie was nominated for an award but never won.

Question	Post-RAG Output Correctness:
Do you know any movies with Danny Deveto?	**Can't Answer-** Is it unable to identify the misspelled "Danny DeVito"
Do you know any movies with Leonardo DiCaprio?	**Correct-** Gives Movies with this actor
What is the rating of Twilight (2008)?	**Correct-** Gives the exact IMDB rating (5.3)
Tell me about the movie Past Lives?	**Can't Answer-** The data is only updated through April; this was released May 1st.
Tell me about the movie	**Correct-** Gives information

Electric State	on this March 2025 movie
Which actor plays Miles in Into the Spider-Verse?	**Correct-** Gives the specific actor that plays Miles
What actors are in Into the Spider-Verse?	**Correct-** Gives the actors from this movie

Table 1: Correctness of the Post-RAG LLM

5. Discussion

With every project/idea, there are challenges a team will encounter. When creating BingeBot we encounter challenges both involving the LLM and that datasets used to train it. Many of those challenges have already been fixed but some would require more time, and more structured and complete data to fix.

5.1. Issues/Challenges

One of the main challenges with this LLM was ensuring that the datasets used were accurate and up-to-date. Netflix, like many other streaming services, is constantly adding and removing movies to their platform. While RAG training on recent articles (that show movies that have been added/removed from Netflix within the past few months) helps the LLM to provide more accurate results, it was a struggle to identify datasets that were completely up-to-date. Netflix itself, also doesn't provide a current dataset of their full-library for the public. All datasets were collected and created by other third-parties other than Netflix. Therefore, it was impossible to collect datasets that were 100% accurate and current.

5.2. What's not covered

The datasets we used are mostly focused on movies that are currently in the United States Netflix catalog. This means if a user asks for a specific movie that is only available outside of the United States catalog, the model would not be able to help. Furthermore, we also included datasets that have IMDB ratings, Oscar nominations, and Rotten Tomatoes ratings that also include Netflix movies. Unfortunately, not all these datasets are complete, and some only have movies made in specific years. For example, the Oscar nominations dataset only has movies on Netflix with nominations before 2020.

5.3. Limitations

BingeBot still has a few limitations. For starters, while the goal is to implement recently updated data so it provides the most accurate and recent results, it may still be slightly outdated. While it is updated with April and March 2025 movies, it will need to be updated with additional Netflix movie information for May 2025 and future releases.

The model also struggles with misspelled movie titles and actor names. If the user asks about an actor, whose name is slightly incorrect, it will be unable to provide info on them. The model depends on spelling accuracy.

Lastly, the LLM sometimes fails to specify what region these Netflix movies are available in. Therefore, it can sometimes suggest a Netflix movie that isn't available in the user's country or region. Netflix is also changing their available movies rapidly, so sometimes the LLM will suggest a movie that has been recently removed from Netflix's library. Keeping the LLM consistently updated is important to ensure the results are accurate and up-to-date.

5.3. Lessons Learned

As we developed BingeBot, we learned that strong performance relies not only on selecting a capable model, but also on using accurate and complete data. Once we added RAG training with updated datasets, the model became much more reliable. The base Gemini model often gave incomplete or outdated answers, especially when users asked about recent releases or less common genres. After we trained it with new datasets that included current movies and ratings, its responses became noticeably more accurate and relevant.

We also realized that the quality of our datasets had a direct impact on the model's performance. Early on, missing ratings or partial descriptions caused the model to give vague or incorrect responses. When we addressed those gaps by collecting more specific datasets, such as Rotten Tomatoes scores or recent Oscar winners, the results improved. This showed us that adding high-quality data matters just as much as the model architecture.

Testing the model with real questions helped us see how users might interact with it. By writing prompts that matched how people usually search for movies, we were able to identify weak areas and figure out how to improve them. Reviewing the answers as a team helped us stay consistent and decide together what a good response looked like. This process also made it easier to balance technical accuracy with a good user experience.

In the end, we found that training the model, testing it carefully, and adjusting it based on real examples made the biggest difference. These lessons can guide how we would improve BingeBot in the future and will help us in other projects or jobs that involve LLMs and user-centered design.

6. Related Work

Streaming services regularly use Movie and TV show recommendations to entice their users to continue watching on their platform. This in turn allows their users to enjoy the platform more and spend more money on the platform benefiting the streaming service. A movie recommender is both a strategic source of income for a streaming service, and an easy way to find a movie to watch for the users.

6.1. Types of Products

Netflix, one of the biggest streaming service providers, uses a movie and tv show recommender called CinematchSM, which uses movie and tv show reviews to prioritize recommendations[5]. Previously Netflix used a different recommendation algorithm that focused on recommendations based off of renting patterns, but this algorithm did not bring in nearly as much revenue as CinematchSM. Hulu, another well known streaming service, uses a recommendation engine that allows users to specify that they no longer want to be recommended specific types of content[6].

6.2. Similarities

BingeBot and CinematchSM both use movie ratings to recommend movies to their users. This allows the users to specifically set the quality of the movies they are looking for and better personalize the users experience. This system is important given that it is used by commercial recommendation systems like CinematchSM and Hulu's recommendation system and likely entices users towards movies that they otherwise would not have considered watching. Similar to Hulu's movie recommender, you can specify not to recommend specific genres of movies or even individual movies and BingeBot will not recommend them. This allows users to filter what they want to be recommended and further customizes their experience.

6.3. Differences

CinematchSM recommends movies based on data collected on Netflix's application. This is different from BingeBot that allows the user to specify specific ratings, genres, actors and other characteristics that the LLM will then use to recommend movies. The prompting of the user will hopefully take into account a user's preferences and match them with movies they are more likely to enjoy. Hulu's recommendation system focuses on removing content the user doesn't want to see and while this can be done in BingeBot as well the user would be better off specifying specific aspects of a movie that they are looking for. BingeBot has both similarities and differences with current commercial recommenders but incorporates more personalization by prompting the users for specifics. This will hopefully entice users to spend more time on the application benefiting both the user and the streaming service.

7. Conclusion and Future Work

BingeBot is a movie recommendation system that aims to decrease the time users take when deciding what to watch. This will allow streaming services to entice their users to spend more time on their sites therefore making them more money. This will also benefit users by allowing them to spend more time watching movies rather than searching for something to watch. While many streaming services already have their own movie and TV show recommendations, BingeBot gives the user more freedom to specify their preferences with many different characteristics to narrow down their movie selection. BingeBot was made using Gemini 2.0 flash model that had been trained and fine tuned using RAG.

The model incorporates several different datasets focusing on Netflix movies and has many different fields including age ratings, genres, actors, oscar nominations, IMDB ratings, rotten tomato ratings and several more. The model itself is able to answer questions given specific parameters by the user and recommend movies given those parameters. This allows the user to search for movies with a specific intent and pick from a list of recommendations that the model returns. Overall, the model has performed well and will be able to help users arrive at a movie decision quicker while allowing them to find a movie that fits their tastes.

7.1 Future Work

There are several ways we plan to improve BingeBot moving forward. One of our main goals is to update the datasets more frequently so the model can continue to give accurate and current recommendations. Since Netflix regularly adds and removes titles, having the most recent data will help the model avoid suggesting movies that are no longer available. We also want to expand the data to better support international users by including region-specific availability.

Another area of focus is improving the user experience. While the current version responds to text prompts, we plan to design a more interactive interface. This could include search filters, clickable categories, or personalized watchlists that make it easier for users to explore their options. Adding these features would also help the model feel more like a complete tool rather than just a backend system.

In addition, we want to gather feedback from users outside our team. Testing the tool with new users will help us find areas that need more clarity or better performance. We may

also experiment with automated evaluation tools to measure the relevance of recommendations and track improvements over time.

In the long term, we hope to expand BingeBot to include other types of content like TV series or documentaries. We also plan to explore whether it could be adapted for use on other streaming platforms. These improvements will help BingeBot grow into a more useful and flexible system that meets a wider range of user needs.

8. References

[1]]S. Bansal, "Netflix Movies and TV Shows," *www.kaggle.com*, 2021. https://www.kaggle.com/datasets/shivamb/netflix-shows

[2] amirtds, "kaggle-netflix-tv-shows-and-movies/titles.csv at main · amirtds/kaggle-netflix-tv-shows-and-movies," *GitHub*, 2025. https://github.com/amirtds/kaggle-netflix-tv-shows-and-movies/blob/main/titles.csv (accessed May 07, 2025).

[3] "Lost in the Stream: Survey Finds Americans Waste Nearly Five Days a Year Just Deciding What to Watch," *Usertesting.com*, 2019. https://www.usertesting.com/company/newsroom/press-releases/lost-stream-survey-finds-americans-waste-nearly-five-days-year-just?utm_source=chatgpt.com (accessed May 07, 2025).

[4] "Frequency of going to the movies in the U.S. 2019 | Statistic," *Statista*, 2019. https://www.statista.com/statistics/264396/frequency-of-going-to-the-movies-in-the-us/

[5]]veeralakrishna,"GitHub-veeralakrishna/Case-Study-ML-Netflix-Movie-Recommendation-System: A Machine Learning Case Study for Recommendation System of movies based on collaborative filtering and content based filtering.," *GitHub*, 2019. https://github.com/veeralakrishna/Case-Study-ML-Netflix-Movie-Recommendation-System (accessed May 07, 2025).

[6] A. Bascetta, "Machine Learning for Recommendations on Streaming Services," *info.keylimeinteractive.com*. https://info.keylimeinteractive.com/machine-learning-for-recommendations-on-streaming-services

[7] Unanimad, "The Oscar Award, 1927 - 2025," Kaggle. Available: https://www.kaggle.com/datasets/unanimad/the-oscar-award

[8] IMDb, "IMDb Data Files," IMDb Datasets. Available: https://datasets.imdbws.com/