Design and Implementation of Chinese Movie Recommendation System Based on Large Model

Ziqian Hu
Guangzhou Southern College, Guangzhou, Guangdong, China

Abstract: Due to the rapid development of Internet video platform and the explosive growth of movie resources, users will inevitably encounter a serious problem of "information overload" when using the platform. At present, the mainstream movie recommendation system can't well tap users' deep interest preferences. However, the large language model has shown excellent performance in semantic understanding, knowledge representation and natural language generation, which points out a very clear new direction for the intelligent upgrade of recommendation system. In this paper, the Chinese film recommendation system based on the big model is designed clearly, and its fundamental purpose is to reasonably combine the semantic understanding ability of the big model with the modern architecture of front-end separation, so as to build an accurate, efficient and easy-to-use Chinese film recommendation platform. Therefore, this paper first introduces five core modules involved in the whole process of user viewing: user management, movie retrieval, personalized recommendation, viewing record management, recommendiation reason display and system management. At the same time, the recommendation engine integrates Chinese big models such as Baidu ERNIE and Deep Seek, and achieves semantic matching between user interests and movie content by means of model fine-tuning and prompt engineering.

Keywords: Large Language Model; Recommendation System; Movie Recommendation; User Portrait; Natural Language Processing

1 Research Background and Present Situation

1.1 Research Background

For fans who frequently watch movies, the stock of domestic mainstream video platforms is over 10,000, and hundreds of movies are added to the cinema every year. Users want to pick out what they are interested in, and the cost is getting higher and higher.

The recommendation system was originally designed to solve this problem, but the traditional recommendation algorithm mainly relies on the user's rating, click behavior and so on to complete the recommendation matching, which can only achieve the superficial behavior fitting, and can not fully understand the hidden characteristics of the film's plot style, director's narrative, local cultural connotation and so on; However, with the appearance of BERT, ERNIE, Chat GLM, Deep Seek and other large models, they have the ability of deep semantic understanding and text content generation, which can analyze the characteristics of natural language query and user's emotion and film content, and provide a brand-new technical path for optimizing Chinese film recommendation system. At the same time, the domestic big model is specially optimized for Chinese context and local culture, which makes the big model more suitable for the language of domestic film and television. Therefore, it is feasible to build a movie recommendation system based on the Chinese big model.

From the design significance of the system, it provides some application value to users, platforms and industries: for terminal users, oral search is realized by relying on the strong interactive ability of large model natural language, and at the same time, it is matched with interpretable personalized recommendation, which reduces the cost of user screening; For the video platform, accurate recommendation can revitalize the long-tailed niche films, thus improving the retention of users and the broadcast volume of the platform; For the industry, the landing recommendation scenarios of large models and platforms have been opened, the implementation scheme of Chinese generative recommendation project has been improved, and the cases of LLM landing in the

entertainment field have been enriched.

1.2 Research Status at Home and Abroad

Foreign recommendation algorithms started earlier and accumulated deeply. Bell and Koren systematically summarized the experience of this competition from Amazon collaborative filtering [1] to Netflix matrix competition, and pointed out that matrix decomposition algorithm has made a breakthrough in scoring prediction accuracy, and its variants are still the core components of many recommendation systems [2], while deep learning models such as Wide&Deep[3], NCF and SAS Rec[4] have realized nonlinear feature fitting. With the landing of large models such as Transformer[5], GPT[6] and BERT[7], the generative recommendation algorithm based on Prompt project has become a research hotspot, and with the automatic generation of recommendation reasons by XAI-Rec[8] and other frameworks, the recommendation systems such as Netflix[9] and Amazon Prime Video[10] have all landed large models to assist the video retrieval business.

Domestic research follows foreign countries closely, but pays more attention to the scene application of large model in Chinese. A large number of scholars have done a lot of large model applications for Chinese optimization. In the aspect of user portrait, some scholars put forward the framework of movie recommendation system based on user portrait, and improved the accuracy of user portrait through multi-dimensional feature fusion [11]. Chen Biyi and other systems research and integrate explicit feedback (scoring and collection) with implicit feedback (click and stay time) to make the user's portrait more plump [12]. On this basis, Wang Yuchen further introduced the trust factor to enhance the diversity of recommendation results[13]. In industrial application, Russo proposed a hierarchical large model architecture in his master's thesis to model objects and users respectively[14]. Tung system studies the movie recommendation method based on generative AI, which provides a theoretical reference for the application in Chinese scenes[15]. However, the mainstream domestic film and television platforms still rely on traditional collaborative filtering algorithms, and they generally have some problems: recommending top popular films, poor fault tolerance in natural language retrieval, and lack of recommended explanation documents; However, Cat's Eye and Douban gradually introduced the optimized retrieval function of domestic large models, but there were few cases in which the whole link of large models was embedded in recommendation system.

1.3 Development Trend and Research Content of This Paper

There are four trends in the future movie recommendation system as a whole: first, the large model will be more and more deeply integrated into the recommendation system[16], second, the system can better accumulate data, so the recommendation will be more accurate and personalized[17], third, natural language dialogue will become the mainstream interactive mode, and fourth, the dynamic and real-time update of user portraits, cross-platform deployment and privacy compliance will become the rigid development requirements of the system.

The main research content of this paper is: the system completes the full-function system design based on the front-end separation architecture, integrates Chinese large models such as ERNIE, Chat GLM and Deep Seek to realize semantic retrieval and personalized recommendation functions, optimizes the dynamic updating logic of user portraits, and solves the problems of large model call delay, sparse portraits and multi-terminal compatibility, and verifies the effectiveness of the system through multi-dimensional tests of function and performance.

2. Key Technical Principles

2.1 The Mainstream Chinese Big Model

2.1.1 ERNIE Model

ERNIE is a very representative model in the series of knowledge-enhanced semantic understanding models launched by Baidu, which has a very clear and beautiful improvement compared with BERT. The most outstanding and solid innovation of ERNIE model is its multi-level masking mechanism: ERNIE 1. 0 itself naturally and appropriately adopts three different levels of masking strategies: basic masking acts on single words or word granularity, phrase masking masks continuous phrases as a whole, and entity masking directly takes named entities (people's names, place

names and movie names) as masking units. Therefore, the model is naturally guided to learn the knowledge at sentence level and semantic level during training, and will not simply remember the local co-occurrence relationship.

2.1.2 Chat GLM Model

Zhipu Chat GLM is based on GLU activation function and RLHF human feedback to fine-tune and optimize dialogue ability, and is good at natural language intention analysis and free text generation. Its technical characteristics can be divided into three levels: the first is the efficiency of the model architecture, that is, Chat GLM uses GLU activation function and Deep Norm normalization technology, which greatly improves the stability of training while obtaining good performance. The second is the bilingual training strategy. The model is jointly trained on Chinese and English corpora, so it has a full grasp of Chinese understanding and excellent cross-language knowledge transfer ability. The third is dialogue optimization, which uses supervised fine-tuning and reinforcement learning (RLHF) based on human feedback to make the model more in line with human dialogue habits and human value orientation.

2.1.3 BERT Model

Google BERT model relies on bidirectional Transformer encoder to realize bidirectional semantic modeling of context, which is mature in technology and rich in open source weight, and is mostly used for text classification and emotion vector extraction.

2.1.4 Deep Seek Model

Deep Seek model is a large language model developed by Deep Search Company. Its V4 version adopts the sparse architecture of Mo E (Mixed Expert), with a total parameter of 1.6 trillion, but only about 49 billion parameters are activated each time, thus achieving a balance of "large parameters, low activation and high efficiency".. In the aspect of understanding Chinese, its ability to analyze Chinese metaphors and idioms is significantly better than that of the baseline model. In the task of semantic analysis of idioms, the accuracy of Deep Seek in judging the contextual applicability of "superfluous words" and "unnecessary words" is 92.3%, which can accurately capture the unique feature of "polysemy+contextual dependence" in Chinese(e.g., Table 1).

Table 1. Comparison of Main Chinese Large Models

model	Research and development institutions	Core characteristics	Applicable scenario
ERNIE	Baidu	Knowledge enhancement mask and multi-level semantic understanding	Feature Extraction and Interest Modeling of Film Content
Chat GLM	Zhipu AI/ Tsinghua	Dialogue optimization, strong generating ability and bilingual training.	Natural language interaction and recommendation reason generation
BERT	Google	Bidirectional encoder, mature and stable, easy to fine-tune.	Semantic parsing, text classification, sentiment analysis.
Deep Seek	Deep search	Mo E sparse architecture, million-long context and ultra-low API cost	Semantic parsing of long texts, generation of explainable recommendation reasons, and large-scale deployment.

2.2 Traditional Recommendation Algorithm

Collaborative filtering (CF) is one of the most classic and widely used algorithms in recommendation system. The core idea is "birds of a feather flock together, and people are divided into groups". User-based collaborative filtering is to find people with similar interests and recommend movies that those people like to you; Collaborative filtering based on items is to find movies similar to your favorite movies. It is mainly divided into two categories: user-based collaborative filtering (User-based CF) and item-based collaborative filtering (Item-based CF).

Neural Collaborative Filtering (NCF) is an extremely representative work in neural network recommendation model. It uses multilayer perceptron to replace the inner product operation used in matrix decomposition, so neural network can be used to learn the user-object interaction function. Specifically, the NCF framework generally includes two distinct branches: the general matrix decomposition branch (GMF) is used to model linear feature interaction, and the multi-layer perceptron branch (MLP) is used to model nonlinear feature interaction. The output results of the two branches are spliced and then

sent to the output layer for final prediction. Therefore, it not only inherits the linear expression ability of matrix decomposition, but also introduces the nonlinear modeling ability of deep neural network.

2.3 Development Framework and Storage Technology

The front-end of this system adopts the component library of Vue.js3+Element Plus, Vue Router handles page jumps, Pinia manages global status (such as user login information), and all kinds of tools are available to complete fully responsive page development. The back-end adopts Fast API, which is developed based on Starlette(Web part) and Pydantic (Data part), and has the characteristics of automatically generating interactive API documents, asynchronous support, type prompt and so on. At the same time, MySQL database is used to permanently store structured data such as users, movies and ratings. Redis is also used to cache hot movies, user portraits and recommendation results, which shortens the time-consuming interface response.

3 System Requirements and Scheme Design

3.1 Feasibility Analysis

On the technical level: the front-end framework and database are mature open source technologies, while Deep Seek opens the API calling interface to the outside world for free, and the threshold for integration is low; Resource level: relying on Movie Lens Chinese data set, Douban public film data set and TMDB open API to supplement movie metadata, simulated users are used to detect behavior data in the testing stage; In terms of hardware, the development and debugging can be completed by using ordinary PC, and the project cycle is reasonable and controllable.

The sources of the data set are divided into four categories: Movie Lens Chinese enhanced scoring data set, Douban 300,000 movie metadata and short reviews data set, domestic box office statistics data set, TMDB+ Douban API real-time incremental data, and the data is put into storage after being cleaned, duplicated and filled in.

3.2 Functional Requirements Analysis

The main functions of this system are divided into five core business modules: (1) User management module: account registration and login (password/SMS verification code), personal information editing, role authority division (ordinary users/administrators) and manual adjustment of interest labels; (2) Film retrieval module: multi-condition screening (year/region/director/genre), intelligent retrieval of natural language, and film details display; (3) Personalized recommendation module: multi-dimensional dynamic user portrait construction, multi-channel mixed recommendation (guess your favorite/similar movie/hot high score), large model generation recommendation reason, user negative feedback optimization recommendation; (4) Movie viewing management module: movie viewing progress record, movie collection, star rating+short comment, self-built private/public film list, and visual statistics of movie viewing data; (5) System management module (administrator): user account management, adding, deleting and checking movie information, recommendation weight parameter configuration, operation data kanban and user feedback processing.The figure 1 presented below demonstrates the five major functional modules of the proposed system.

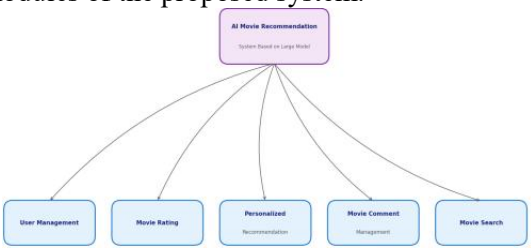

Figure 1. System Core Function Module

3.3 Performance Index Requirements

The interface response time of the recommendation system is 1 ~ 3s, the conventional screening retrieval is ≤1s, the page loading is ≤2s, the system supports concurrent online access of users ≥500, the natural language retrieval accuracy is ≥90%, the recommendation accuracy and the comprehensive satisfaction of users are ≥ 85%; The annual system availability rate is ≥99.5%, and the user's private data is encrypted and stored, which conforms to the specification of the Personal Information Protection Law.

3.4 System Overall Architecture Design

The system adopts front-end and back-end separation architecture. The front-end Vue3 is responsible for page rendering and user

interaction, while Nginx provides static resource services and forwards /api requests to the back-end of Fast API. The backend accesses MySQL through SQL Alchemy asynchronous ORM, encapsulates cache reading and writing through Redis_client, and calls Deep Seek interface through llm_service. The overall architecture of the system is shown in Figure 2.

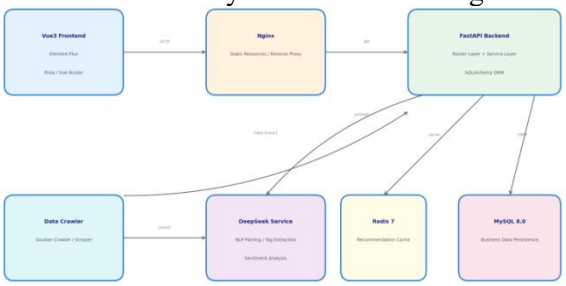

Figure 2. System Architecture

The presentation layer is the interface that users see, that is, the front end, running in the browser. Build components with Vue.js 3, use Element Plus as UI, use Vue Router to manage routing, and use Pinia to manage global status. Send a request to the back-end API through Axios.

The business layer is a back-end service that runs on the cloud server. Write RESTful interface with Fast API, and divide it into logic modules (not independent processes) such as user service, movie service, recommendation service, portrait service and interactive service. Call the big model SDK to complete the task. Time-consuming operations (such as batch reasoning) are handed over to Celery asynchronous queue to avoid blocking the main thread.

The data layer manages data. MySQL stores structured data, while Redis caches and stores temporary data.

3.5 Database E-R and Table Structure Design

The E-R diagram (entity-relation diagram) of the database describes the logical relationship between entities, as shown in Figure 3.

Figure 3. E-R diagram of Database

Core entities of the system: users, movies, movie viewing records, rating comments, user tags and movie lists, and users-movies form many-to-many relationships through collection/rating/movie viewing records; Users, movies and tags are many-to-many, and the relationship is stored by the middle tag weight table(e.g., Table 2).

Table 2. Design of Core Data sheet

Table name	Main field	explain
users	Id, username, phone, password_hash, role, status	Save user account, authentication information and role status.
movies	title, year, rating, directors, actors, genres	Save the basic information, rating and content characteristics of the film.
user_ratings	user_id, movie_id, rating, comment	Save user ratings and short comments to support collaborative filtering.
user_favorites	user_id, movie_id	Save favorite behavior for recommendation exclusion and preference analysis.
watch_history	user_id, movie_id, watched_at, progress	Save the movie record
playlists	user_id, title, is_public	Save user sheet
recommendations	user_id, movie_id, algorithm, score	Record the recommendation results and algorithm sources.
feedbacks	user_id, type, content, status, reply	Save feedback and processing results.

3.6 Detailed Design of Core Module

3.6.1 Portrait of users

User portrait is the basis of personalization.

Data acquisition: collect multi-source data. Explicit data: registration preference, manual tagging, rating and collection; Implicit data: viewing history, search keywords, clicks, page stay time, comment content. All collections are compliant, and users can export and delete them.

Feature extraction: structured data is directly mapped into label-weight; Sentiment analysis and keyword extraction of text data with large model; Behavior sequence uses algorithm to find short-term interest drift. The unified expression is label-weight, and labels cover types, specific works, people, styles, years and so on.

Dynamic updating: Because the incremental learning method is used, the formula for updating the weight when a new behavior occurs is naturally written as new weight = old weight × attenuation coefficient+behavior contribution value, and the selected attenuation coefficient is 0. 95, so the old interest will fade automatically and reasonably, and because the upper and lower limits of the weight are set, it can prevent anomalies well. Users can adjust labels directly, and manual adjustment has the highest priority.

The formula is as follows:

$$w_{u,t}^{(new)}=\alpha \cdot w_{u,t}^{(old)}+\sum_{b\in B_{u,t}}\beta_b \cdot imp_b \quad (1)$$

Among them:

(1) indicates the interest weight of the user U to the updated tag T;$w_{u,t}^{(new)}$

(2)α is the time attenuation coefficient (0.95), which is used to gradually dilute historical interest;

(3) Represents the behavior set of user U related to tag T in this update cycle;$\in B_{u,t}$

(4) The contribution coefficient of behavior type B (for example, 0.3 for complete viewing, 0.5 for rating 5 stars, 0.4 for collection, and-0.2 for fast-forward skipping);β_b

(5) The normalization factor for the importance of behavior B (e.g. the score value is normalized to [0,1]).imp_b

3.6.2 Recommendation Algorithm Optimization Module

The recommendation algorithm uses a mixed strategy, which combines large model recommendation, collaborative filtering and content-based recommendation.

Recall: Multi-channel recall, including tag recall (matching users' high-weight tags), collaborative filtering recall (similar users and similar items) and popular recall. Screen out hundreds of candidates from a massive library.

Ranking: Calculate the user-movie matching degree with the depth matching of large model, and combine the movie quality score (score, heat) and timeliness factor to get the comprehensive ranking score. The large model inputs user portrait description, movie content and historical interaction, and outputs matching scores. In order to reduce the delay, hot combinations are pre-calculated and cached, and personalized requests are processed asynchronously in batches.

Adjustment: Rearrange according to the diversity rules, so that various factors such as type, age and region are evenly distributed.

Explanatory generation: Because the recommendation reason is generated by a large model, it is natural and reasonable to adopt a template containing three elements: user preference basis, movie matching point and recommendation conclusion, input structured information and output natural language.

Evaluation and optimization: the accuracy, recall and NDCG are evaluated by offline method, and then the click-through rate and satisfaction are compared and analyzed by online A/B test, thus the optimization is naturally carried out.

The formula is as follows:

$$Score(u,i)=\lambda \cdot S_{LLM}(u,i)+(1-\lambda)\cdot S_{CF}(u,i)+\gamma \cdot Q(i)+\delta \cdot T(i) \quad (2)$$

Among them:

(1) the comprehensive recommendation score for the movie I by the user U;Score(u,i)

(2) Semantic matching score (value range [0,1]) output for large model;$S_{LLM}(u,i)$

(3) Similarity scores calculated for collaborative filtering algorithms (including user-based and item-based);$S_{CF}(u,i)$

(4) for the film quality score, the comprehensive douban score, platform score, heat and other calculations;Q(i)

(5) As a timeliness factor, the newly released films gain higher weight;T(i)

(6) λ, γ and δ are adjustable superparameters, which satisfy $\lambda\in[0,1]\lambda\in[0,1]$, and the optimal value is determined by A/B test.

$$Sim(q,i)=\cos(v_q,v_i)=\frac{v_q \cdot v_i}{\|v_q\|\|v_i\|} \quad (3)$$

Among them:

(1) query the semantic vector of Q encoded by ERNIE for users' natural language;v_q

(2) It is the semantic vector of the synopsis, labels and other information of movie I after being encoded by a large model;v_i

(3) Cosine similarity, with the value range of [-1,1], and the larger the value, the more relevant the semantics.$\cos(v_q,v_i)$

4 System Engineering Realization

4.1 Development Environment

Development equipment: 12th Genlntelr (R) Core i7-12700h, DDR 4 (16GB); Deployment server: Alibaba Cloud ECS 2 core 8 GB Ubuntu 20.04; Software environment: Python3.12, Node18, MySQL8.0, Redis7.0, VS Code+PyCharm, Dockers.

4.2 Front-end and Back-End Function Realization

4.2.1 Front-end Interface Implementation

The front-end project of this system is created by Vue CLI, in which several key components: Movie Card.vue is used to package movie cards, Search Bar.vue is used to support two search modes, and Rating Star.vue is used to display rating stars. The status is managed by Pinia, the user store is used to store login information, and the app store is used to manage UI status.

4.2.2 Implementation of Back-end Interface

The backend of this system adopts Fast API, so the project structure can be organized by convention. The implementation process of the designed recommendation interface is very clear: check the Redis cache first, and return it directly if there is one, without calling the recommendation service to get the result asynchronously and cache it. JWT authentication is realized by dependency injection, and permissions are controlled by decorator.

4.3 Large Model Integration Realization

The large model call is encapsulated in llm_service.py The key of natural language retrieval lies in transforming users' colloquial needs into structured query conditions. The parse_natural_query method in llm_service.py constructs a prompt word, which requires Deep Seek to return the JSON field. If the returned content cannot be parsed or the API call fails, the system calls the fallback_parse function to parse the rules, such as identifying four-digit year, movie type, domestic/American/Japanese words and the exclusion condition of "don't make a certain type". This can not only make use of the semantic understanding ability of the large model, but also ensure that the basic retrieval ability does not depend on external services.

4.4 Optimization Scheme for Key Issues

4.4.1 Large Model Call Response Delay Optimization

Lightweight and fine-tuning of the model: LoRA(Low-Rank Adaptation) technology is used to fine-tune ERNIE-Tiny in the natural language retrieval task, and quantization technology is used to quantize Chat GLM from FP16 to INT8 in the recommendation reason generation task.

Multi-level cache strategy: the first level cache is used to cache the features (plot vector, tag vector) of popular movies, and it is used for long-term cache. The second level cache is used to cache the retrieval results of common queries. The third-level cache stores personalized recommendation results, and directly returns the repeated requests of the same user in a short period of time to the cache.

Asynchronous processing and preloading: users execute recommendation tasks asynchronously and cache the results after logging in, and directly read the cache when users visit the recommendation page, so they can meet the requirements naturally and appropriately. At the same time, the scheme of asynchronous call+loading status prompt is adopted: after the front end sends out the request, it displays "I am understanding your problem." The back-end submits the task to the Celery queue for asynchronous processing, and the result is directly pushed by WebSocket or obtained by front-end polling after processing.

Parallel call and result fusion: Python's asyncio is used to perform various IO operations such as database query, cache reading and model call concurrently, thus changing the overall time consumption originally determined by the slowest serial call into the longest single call time.

The formula is as follows:

$$H_{total}=1-\prod_{k=1}^{n}(1-H_k) \quad (4)$$

$$T_{avg}=H_{total}\cdot T_{cache}+(1-H_{total})\cdot T_{LLM} \quad (5)$$

Among them:

(1) The hit rate of the K-level cache (the hit rate of the first-level cache is about 40%, the hit rate of the second-level cache is about 30%, and the hit rate of the third-level cache is about 20%);H_k

(2) is the total cache hit rate;H_{total}

(3) It takes time to read the results from the cache (about 50 ms);T_{cache}

(4) It takes time (about 2-3 seconds) to directly call the big model.T_{LLM}

4.4.2 User Portrait Accuracy Optimization

Multi-dimensional collection: in addition to scoring collection, it also collects retrieval records, residence time and scrolling depth.

Search results click to reflect interest, and long stay indicates high interest.

Data cleaning: identify the wrong click (return immediately after clicking) to lower the right; Merge and repeat operations; Abnormal behavior (a large number of scores in a short time) is directly ignored.

Transfer learning: Pre-train the basic model with public data sets such as Movie Lens, and then fine-tune it with the data of this system, which can also be used in cold start. The new user initializes the portrait according to the registration information and demographic characteristics.

Manual correction: users can view and manually adjust the label weights, and the manually adjusted weights will be retained when the system is updated.

4.4.3 Cross-platform Adaptation

Standardized development: Use standard HTML/CSS to avoid browser-specific features. Autoprefixer is automatically prefixed.

Responsive components: Element Plus has dealt with most compatibility issues. Use media query to adapt to different screens when writing CSS yourself. The mobile terminal optimizes touch events and increases the click area.

Multi-device testing: Cloud testing with BrowserStack covers mainstream browsers and mobile devices. Fix problems in time, such as dislocation of the fixed navigation bar on iOS Safari and dynamic adjustment with JavaScript.

5 System Test and Result Analysis

5.1 Test Environment and Use Cases

The test environment is consistent with the project code. Python 3.12, FastAPI, SQLAlchemy, MySQL 8.0 and Redis 7 are used in the back end, and Vue3, Vite, Element Plus and Axios are used in the front end. The deployment method can be Docker Compose, or the front and back ends can be started locally. The test focuses on the core processes such as user registration, movie browsing, natural language retrieval, recommendation display, collection rating, film list and background management.

5.2 Test Results

5.2.1 Functional Test

The function test adopts the black box test method, and the use case is designed according to the actual operation path of the user. When testing, pay attention to whether the input, operation steps, expected results and actual results are consistent, and avoid fabricating indicators that cannot be verified from the system(e.g., Table 3).

5.2.2 Performance Test

The performance test is mainly aimed at recommendation interface, conditional retrieval interface and natural language retrieval interface. Because the system uses Redis cache, the performance of the first request and cache hit request is quite different, so the cold start and cache hit scenarios should be distinguished during testing. The compatibility test focuses on verifying the mainstream browsers such as Chrome and Edge, as well as the page display with different window widths. The pages in the demo video can be displayed normally in the desktop browser(e.g., Table 4).

Table 3. System Function Test Cases

test module	test case	Expected result	net effect
user registration	Enter a valid mobile phone number/user name and password to submit.	Registration is successful and you can jump to login or homepage.	pass
user registration	Enter an existing account number	Prompt that the account number or mobile phone number has been registered.	pass
User login	Enter the correct account password.	Return to Token and enter the system.	pass
Movie browsing	Click on the Movie Library and select the Type tab.	The list is filtered and displayed by type	pass
Natural language retrieval	Enter "domestic suspense films in 2022, not horror films"	Returns movies that meet the criteria of year, region and genre.	pass
Guess you like it	Enter the personality recommendation page after logging in.	Show recommendation results based on portrait and collaborative filtering	pass
Collection score	Collect and rate movies	Collection status and scoring record were saved successfully.	pass
Sheet management	Create a film list and add movies	Movie list details show that movies have been added.	pass
Background user	Administrators search, disable or enable	User status updated successfully.	pass

management	users.		
Feedback management	Users submit feedback and administrators reply.	Feedback status and reply content are displayed correctly.	pass

Table 4. Performance Test Design Table

Test item	Concurrent number	Expected index	test specification
Recommended interface	50	The average response time is less than 3 seconds.	Distinguish between first-time calculation and cache hit scenarios
Multi-conditional retrieval	100	Average response time is less than 1 second.	Filter by type, year, score and other conditions.
Natural language retrieval	thirty	The average response time is less than 3 seconds.	Including large model analysis and rule degradation.
Mixed access	500	The error rate is less than 1%	Simulate mixed access of home page, search, recommendation and details

5.2.3 Compatibility Test

The main browser functions normally, and the layout is slightly different, but it does not affect the use. The fixed navigation bar problem of iOS Safari has been fixed. The mobile touch event has a good experience after optimization.

6 Summary and Prospect

6.1 Summary

This paper completed the requirement analysis and design: investigated the existing platform, defined the potential users, defined the functional and non-functional requirements, drew the architecture diagram, E-R diagram, and wrote the detailed design.

The core functions based on the big model are realized: front-end Vue.js 3+Element Plus, back-end FastAPI+MySQL+Redis, which integrates ERNIE, Chat GLM, BERT and Deep Seek, and realizes natural language retrieval, user portrait, personalized recommendation and interpretable recommendation.

Several hard bones have been solved: because the large model is called slowly, it is solved by lightweight, caching and asynchronous methods, and because the portrait is not accurate, it is improved by means of multidimensional data, cleaning, migration learning and manual correction. The cross-platform compatibility problem is naturally solved by standardized development, responsive design and multi-device testing.

The effect of the system is verified: it can be clearly and reliably seen from the experimental results that the recommended response time is 1-3 seconds, and it can support 500 users to respond at the same time. The accuracy of natural language retrieval has reached over 90%, and the user satisfaction has reached over 85%.

Although the scale of this system is not large, it is a very complete and solid engineering practice for me, so it can put all kinds of technologies learned into practice, and it is naturally more conducive to understanding the application of large models in recommendation systems.

6.2 Outlook

There are many plans for the future of this study, and there are still many improvements in the first system, so we should improve this system actively and in a planned way, strive to make the system bigger and better, and strengthen our own knowledge and practical ability. Secondly, there is still a lot of room to explore the large model ability of this system, so although only texts are used at present, it is natural to introduce posters, trailers and other visual materials for multimodal recommendation in the future, and at the same time, the relationship between films (such as the adaptation of the same original) can be used in combination with knowledge maps. Thirdly, there is still room for improvement in the recommendation algorithm of this system, so the future direction can be made clear: from the current rule fusion to end-to-end generative recommendation, the user behavior sequence is directly input into the large model to obtain the recommendation result, and reinforcement learning is used to optimize the strategy to maximize long-term satisfaction. Fourth, the user experience of this system can be more extreme, so you can be a conversational recommendation assistant (such as a small degree, bean curd, etc.). First, you can chat with users for several rounds to find out the needs, so you can be as natural as having friends who know movies around you, and you can add social functions appropriately, so that users can share movies and exchange their experiences, so as to complement each other. Fifth, the privacy protection function of this system should adapt

to the development of the times, so in the future, the federated learning method can be used to keep the data in the user's local area, which is naturally safer only by transmitting the model update, and differential privacy should also be used to add appropriate noise to the recommendation results. Sixth, this system has obvious advantages in cross-domain collaboration: movies, music, books and games are all entertainment contents, so if we can get through the user's interest, the recommendation will be more sufficient and reasonable, and seamless switching of multiple devices is the future development trend, and it is natural to watch half of them on mobile phones and then watch them on TV.

In a word, because the combination of recommendation system and large model is still in its infancy, there will inevitably be various new directions worth trying and studying in the future.

References

[1] Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering[J]. IEEE Internet Computing, 2003, 7(1):76-80.

[2] Bell R M, Koren Y. Lessons from the Netflix Prize Challenge[J]. SIGKDD Explorations Newsletter, 2007, 9(2):75-79.

[3] Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems[C]. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016:7-10.

[4] He X, Liao L, Zhang H, et al. Neural Collaborative Filtering[C]. Proceedings of the 26th International Conference on World Wide Web, 2017:173-182.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[C]. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017:6000-6010.

[6] Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-Training[R]. OpenAI, 2018.

[7] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019:4171-4186.

[8] Lim D, Bae Y, Kim K, et al. XAI-Rec: Explainable Recommendation Using Large-Scale Language Models[C]. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023:1842-1846.

[9] Netflix Technology Blog. Netflix Recommendations: Beyond the 5 Stars[EB/OL].https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429, 2012.

[10] Amazon Science. Conversational AI for Prime Video Search[EB/OL].https://www.amazon.science/blog/conversational-ai-for-prime-video-search, 2023.

[11] Huang Y C. Design and Implementation of Movie Recommendation System Based on User Profile[J]. Journal of Tongren University, 2023, 25(6):75-83.

[12] Chen B Y, Huang L, Wang C D, et al. Collaborative Filtering Recommendation Algorithm Integrating Explicit Feedback and Implicit Feedback[J]. Journal of Software, 2020, 31(3):794-805.

[13] Wang Y C. Diversified Movie Recommendation Algorithm Incorporating Trust Factor[J]. Computer Systems & Applications, 2021, 30(12):187-193.

[14] Russo C F. LLM-based RecSys: Multi-agent Architectures and Advanced RAG Techniques[D]. Università degli Studi di Padova, 2024.

[15] Tung C P L, Haw S C, Kong W E, et al. Movie Recommender System Based on Generative AI[C]. 2024.

[16] Xu L, Zhang Z, Wang Z, et al. Prompting Large Language Models for Recommender Systems: A Comprehensive Framework and Empirical Analysis[J/OL]. arXiv preprint, arXiv:2401.04997, 2024.

[17] Liu X J, Yuan X C, Liu P C. Top-N Recommendation Algorithm for Item-Based Collaborative Filtering Based on Long Tail Theory[J]. Journal of Qiqihar University (Natural Science Edition), 2019, 35(2): 1-4.

