Genres and Actors/Actresses as Interpolated Tags for Improving Movie Recommender Systems

Nghia Duong-Trung[1], Quynh Nhut Nguyen[2], Dung Ngoc Le Ha[3],
Xuan Son Ha[4], Tan Tai Phan[5], and Hiep Xuan Huynh[6,*]

[1, 3]Can Tho University of Technology, Can Tho city, Vietnam
[1, 4]FPT University, Can Tho city, Vietnam
[2]Vietnam Posts and Telecommunications Group (VNPT)
[5]National Chiao Tung University, Hsinchu, Taiwan
[6]Can Tho University, Can Tho city, Vietnam
[*]Corresponding Author

***Abstract*—A movie recommender system has been proven to be a convincing implement on carrying out comprehensive and complicated recommendation which helps users find appropriate movies conveniently. It follows a mechanism that a user can be accurately recommended movies based on other similar interests, e.g. collaborative filtering, and the movies themselves, e.g. content-based filtering. Therefore, the systems should come with predetermined information either by users or by movies. One interesting research question should be asked: "what if this information is missing or not manually manipulated?" The problem has not been addressed in the literature, especially for the 100K and 1M variations of the MovieLens datasets. This paper exploits the movie recommender system based on movies' genres and actors/actresses themselves as the input tags or tag interpolation. We apply tag-based filtering and collaborative filtering that can effectively predict a list of movies that is similar to the movie that a user has been watched. Due to not depending on users' profiles, our approach has eliminated the effect of the cold-start problem. The experiment results obtained on MovieLens datasets indicate that the proposed model may contribute adequate performance regarding efficiency and reliability, and thus provide better-personalized movie recommendations. A movie recommender system has been deployed to demonstrate our work. The collected datasets have been published on our Github repository to encourage further reproducibility and improvement.**

***Keywords*—Movielens; movie recommender systems; tag interpolation; colloborative filtering**

I. INTRODUCTION

Recommender systems (RSs) have been developed to generate meaningful recommendations any products or items to a group of users that might get their attention. RSs [1], [2] are now widely use in research [3], industry [4], and education community [5], [6], where many approaches have been developed for improving recommendations. Many real world examples of recommendation operation can be found for books on Amazon [7], music on Spotify [8], activities on social media [9], [10], services on Twitter [11], [12], or movies on Netflix [13]. The design of these systems depends on the particular characteristics of the datasets, e.g. the ratings of 1 (most disliked) to 5 (most liked). Additionally, the systems might incorporate other information such as descriptions, multimedia contents, and demographic knowledge. Such data sources capture the interactions between items-items, users-users, and users-items. Recommender systems then analyze and learn the underlying patterns in these data sources to develop a correlation between users/items and or items/users which can be used to predict similar pairs. The architecture and evaluation of RSs are an active research area. The infinite solutions to RSs can be categorized into several categorizations. Content-based recommendation models recommend items that are similar to the items that a user has interacted in the past. The second approach is collaborative filtering that recommends items based on all users' past ratings collectively. Tag-aware recommendation [14], [15] approaches the interaction among items that are independent of the existence of users. Our contribution enhances the current research on the tag-based recommendation [16], [17]. Application of tag-based recommendation have been exploited in various domains from personalized social media services [18], e-learning environments [19], personalized location recommendation [20], image search [21], personalized news recommendation [22], personalized music recommendation [23], [24], and many others. A fourth category is a hybrid approach that combines two or more of the previously mentioned categories [25], [26].

Content-based recommendation [27], [28] is the proposal of items based on a comparison between the content of data items and/or user profiles. The content of each item is presented as a set of descriptions, lists of terms or tags, often words that appear in the textual form. The recommended items are primarily related to the items that are relatively rated as a recommendation. Content suggestions use different types of models to find similarities between sources to create the best proposal. The term collaborative filtering (CF) was introduced in a commercial recommender system that recommends newsgroups documents to users [29]. CF analyzes data interaction across users to find matching patterns resulting in other items recommendation [30]. The cold-start problem arises in CF systems where users exist and they have not rated several items before. The motivation of CF is to leverage social collaboration recommend the most similar items/products/services despite a large amount of data. Applications of CF have been developed in a wide range of domains from recommending books [31], musics [32], movies [33], advertisements [34] and other consumer products [35].

Twenty years of MovieLens datasets have witnessed a blossom of research that is garnering a remarkable significance with the advent of e-commerce and the whole industry. Variations of the dataset have been downloaded hundreds of thousands of times, reflecting their popularity and distinctive contribution in the field of recommendation systems and connected subjects. The samples take the form of <*user, item, rating, timestamp*> tuples where each tuple represents a personal preference for a movie at a particular time. A report made by their inventors shows that more than 7500 references to the keyword *movielens* have been made in Google Scholar [36]. A live research system[1] of the MovieLens datasets has been developed and maintained by experts to nurture the personalization and recommendation research. GroupLens research group[2] developed MovieLens as an online movie recommendation system that allows users to rate movies and integrates rating from different sources to collaboratively recommend to other people. Averaged 20-30 new users have signed every day for a long period. This system allows people to create profiles, rate movies, establish tastes and receive recommendations.

Our approach interpolates genres and actors/actresses as tags that predict similarity among movies and provide an appropriate suggestion. We investigate the two-way interactions between $\{item_i,\ [tags]_i\}$ and $\{item_j,\ [tags]_j\}$, where an item i is similar to an item j using the similarity score between their two tags [25], [37]. Practically, tags are collected from users' annotations during the involvement of a recommender system. However, what if the information is missing or does not exist in the first place? Table I presents a quantitative summary of the MovieLens datasets in which the first two variations of the datasets contain no tag information. In this paper, the authors consider another principle design of a movie recommender system: watching movies containing similar genres and actors/actresses (as other movies) lead to watching more same movie categories, which leads to an approach called tag interpolation-based recommendation. We have evaluated the proposed approach on Movielens' variations that contains no manual and/or collected tags from users. Instead, the tags come from movies' genres and actors/actresses. To the best of our knowledge, the research on a movie recommender system based on tags has never been done on the MovieLens 100K and MovieLens 1M variations.

TABLE I. The quantitative summary of variations of the MovieLens datasets.

Variation	# Users	# Movies	# Ratings	# Tags
MovieLens 100K	943	1,682	100,000	**0**
MovieLens 1M	6,040	3,706	1,000,209	**0**
MovieLens 10M	69,878	10,681	10,000,054	95,580
MovieLens 20M	138,493	27,278	20,000,263	465,564

II. Related Work

One of the early attempts to develop a model and build a movie recommender system has been proposed by Azaria *et al.* [38]. In that paper, the authors introduce the profit and utility maximizer algorithm (PUMA) which mounts a black-boxed movie recommender system and predicts movies that will maximize the system's revenue. Another research direction focus on human emotions as the input for movie recommendation [39]. The approach accepts the user profile as part of the system. Deldjoo *et al.* introduce multi-modal content-based movie recommender system [40] that is evaluated on the MovieLens 20M dataset. They exploit the effects of genres as the metadata feature. However, the tags have already provided in MovieLens 20M. The genre features have been further addressed by the same research team of Deldjoo [41]. Another interesting paper that focuses on tag-aware recommendation and the effects of tags over a recommender system is presented in [42]. In that paper, the authors investigated tags from genres and textual reviews on the tag-available dataset, e.g. MovieLens 10M. These models have one thing in common that they perform the recommendation task on tag-available datasets with extra agglomeration from other sources, e.g. genres, users' information, and textual reviews. Our approach differs from these previous ones by the fact that the tags are automatically interpolated from genres and actors/actresses, without any additional manual effort from the users' side and predetermined tags. Consequently, this work is the first to exploit tag interpolation in MovieLens 100K and MovieLens 1M datasets and can be furthered referred for tag-based recommendations.

III. Experiments

A. Datasets

As mentioned in the previous section, the authors employ the MovieLens 100K and MovieLens 1M variations in the experiments because there are no pre-defined tags, but instead, tags are interpolated from the movies' genres and actors/actresses. The datasets [36] can be downloaded on the MovieLens 100K[3] and MovieLens 1M websites[4]. The MovieLens 100K dataset consists of 100,000 ratings (from 1 to 5) from 943 users on 1,682 movies. Each user has rated at least 20 movies. The MovieLens 1M dataset comprises 1,000,209 ratings (from 1 to 5) from 6,040 users on 3,706 movies. For each dataset, the training and test sets have been already split into five-fold cross-validation. The authors run the proposed model on all sets and take an average in the end.

Interpolated tags. We could identify 19 different genres in both MovieLens variations. These tags can be easily extracted from the *u.genre* file of each MovieLens dataset. Regarding actors/actresses, these tags are not included explicitly. Instead, the authors link the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb)[5] from the *u.item* file of each MovieLens dataset, and extract actors/actresses from the IMDB database. One movie in 100K variation contains at least 1 and at most 45 actors/actresses while one movie in MovieLens 1M consists of at least 2 and at most 235 actors/actresses. The number of 14291 and 46198 actors and actresses can be extracted from MovieLens 100K and 1M respectively. The summary of interpolated tags is presented in Table II. The top 10 most used interpolated tags are summarized in Tables (III and IV). The authors make the

[1] https://movielens.org/

[2] https://grouplens.org/

[3] https://grouplens.org/datasets/movielens/100k/

[4] https://grouplens.org/datasets/movielens/1m/

[5] https://www.imdb.com/

collected datasets available at our Github repository[6]. We encourage reproducibility, further comparison and improvement.

TABLE II. INTERPOLATED TAGS FROM MOVIELENS VARIATIONS.

Variation	# Genres	# Actors/actresses
MovieLens 100K	19	14,291
MovieLens 1M	19	46,198

TABLE III. THE NUMBER OF APPEARANCES THE MOST 10 POPULAR GENRES AND ACTOR/ACTRESSES AS INTERPOLATED TAGS IN MOVIELENS 100K DATASET.

Nr.	MovieLens 100K			
	Genres	Freq.	Actors/Actresses	Freq.
1	Drama	725	Samuel L. Jackson	21
2	Comedy	505	Robert De Niro	19
3	Action	251	Steve Buscemi	17
4	Thriller	251	Christopher Walken	14
5	Romance	247	Meg Ryan	14
6	Adventure	135	Christopher McDonald	14
7	Children's	122	Robert Duvall	14
8	Crime	109	Harrison Ford	13
9	Sci-Fi'	101	Gwyneth Paltrow	13
10	Horror	92	Anthony Hopkins	13

TABLE IV. THE NUMBER OF APPEARANCES THE MOST 10 POPULAR GENRES AND ACTOR/ACTRESSES AS INTERPOLATED TAGS IN MOVIELENS 1M DATASET.

Nr.	MovieLens 1M			
	Genres	Freq.	Actors/Actresses	Freq.
1	Drama	1633	Samuel L. Jackson	25
2	Comedy	1218	Joan Cusack	25
3	Action	508	M. Emmet Walsh	25
4	Thriller	495	James Stewart	24
5	Romance	482	Christopher McDonald	24
6	Horror	344	Dan Hedaya	24
7	Adventure	289	Robert Duvall	23
8	Sci-Fi'	276	Frank Welker	23
9	Crime	216	Whoopi Goldberg	22
10	War	159	Robert De Niro	21

B. Evaluation Metric

Root mean squared error (RMSE) and mean absolute error (MAE) are widely used to evaluate the performance of a recommender system given a rating prediction task. The errors quantify the difference between the true rating values and the predicted rating values made by the recommender system. In this work, the authors evaluate the performance of the system by RMSE. We denote r_{ij} and $\hat{r}_{ij}$ as the true rating value and the predicted rating value respectively. Then the RMSE e is calculated as follows:

$$e = \sqrt{\frac{1}{n}\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2} \;, \qquad (1)$$

where the smaller the e is, the better the result is.

Furthermore, based on the computed rating scores of movies, the similarity between any two movies u and v is calculated by using cosine similarity $c_{i,j}$ as follows.

$$c_{u,v} = \cos(\overrightarrow{r_u}, \overrightarrow{r_v}) = \frac{\sum_{i=1}^{m} r_{u,i} r_{v,i}}{\sqrt{\sum_{i=1}^{m} r_{u,i}^2}\sqrt{\sum_{i=1}^{m} r_{v,i}^2}} \;, \qquad (2)$$

where m is the dimensional space of u and v. The list of recommended movies is sorted by values of $c_{u,v}$.

Equation (1) is used to evaluate the performance of the recommendation system. Based on the list of recommended movies created by Equation (2), the system compares the most similar movie's ratings with its prediction by Equation (1). However, in the system deployment presented in Section (IV), the list of recommended movies is more important to the users than the RMSE scores. Hence, the calculation of Equation (1) is ignored in real-time.

C. Experimental Results

The MovieLens 100K and MovieLens 1M datasets contain information on several meta-data such as genres and the names of actors/actresses. The information is considered as the feature of movie observations. We describe the experimental results in the following three scenarios.

1) Scenario 1: Tags are interpolated from the movies' genres.: In this scenario, the authors investigate the performance of the recommender system by utilizing genres as the tags. The experimental results are presented in Tables V and VIII for MovieLens 100K and MovieLens 1M respectively.

2) Scenario 2: Tags are interpolated from the movies' actors/actresses.: Scenario 2 is about the effect of actors/actresses tags to the system's performance. The experimental results are presented in Tables VI and IX for MovieLens 100K and MovieLens 1M respectively.

3) Scenario 3: Tags are interpolated from the movies' combination of genres and actors/actresses.: In the last experimental scenario, the authors combine the tags of both genres and actors/actresses. Tables VII and X presents the system's performance in this scenario.

TABLE V. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 100K. TAGS ARE THE MOVIES' GENRES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	851.67
	u1.test	1.1672	
2	u2.base	–	816.51
	u2.test	1.1373	
3	u3.base	–	900.46
	u3.test	1.1298	
4	u4.base	–	802.54
	u4.test	1.1455	
5	u5.base	–	782.53
	u5.test	1.1489	
Average score		**1.1457 ± 0.0126**	**830.74 ± 41.52**

IV. SYSTEM DEPLOYMENT

A. System Design

The movie recommender system implements the Model-View-Template model during creating an application with user interaction. This model includes HTML codes with Django

[6] https://github.com/nnquynh4496/MovieLens-Tags-Interpolation

TABLE VI. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 100K. TAGS ARE THE MOVIES' ACTORS/ACTRESSES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	47,561.84
	u1.test	1.0626	
2	u2.base	–	48,091.09
	u2.test	1.0472	
3	u3.base	–	52,400.74
	u3.test	1.0343	
4	u4.base	–	50,335.71
	u4.test	1.0960	
5	u5.base	–	48,721.73
	u5.test	1.0382	
Average score		**1.0556 ± 0.0250**	**49,422.22 ± 1,964.03**

TABLE VII. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 100K. TAGS ARE THE MOVIES' THE COMBINATION OF GENRES AND ACTORS/ACTRESSES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	47,561.84
	u1.test	1.0626	
2	u2.base	–	48,091.09
	u2.test	1.0472	
3	u3.base	–	52,400.74
	u3.test	1.0343	
4	u4.base	–	48,073.16
	u4.test	1.0960	
5	u5.base	–	51,526.38
	u5.test	1.0382	
Average score		**1.0556 ± 0.0250**	**49,530.64 ± 2,252.39**

TABLE VIII. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 1M. TAGS ARE THE MOVIES' GENRES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	22,400.17
	u1.test	1.0560	
2	u2.base	–	27,995.05
	u2.test	1.0206	
3	u3.base	–	24,450.00
	u3.test	1.0236	
4	u4.base	–	24,360.05
	u4.test	1.0236	
5	u5.base	–	23,687.43
	u5.test	1.0584	
Average score		**1.0364 ± 0.0190**	**24,578.54 ± 2078.23**

TABLE IX. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 1M. TAGS ARE THE MOVIES' ACTORS/ACTRESSES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	1,343,201.13
	u1.test	1.0354	
2	u2.base	–	1,594,259.98
	u2.test	1.0305	
3	u3.base	–	1,347,105.26
	u3.test	1.0324	
4	u4.base	–	1,379,111.58
	u4.test	1.0324	
5	u5.base	–	1,779,768.24
	u5.test	1.0368	
Average score		**1.0335 ± 0.0025**	**1,488,689.23 ± 193,062.32**

TABLE X. THE SUMMARY OF PREDICTION RESULTS AND RUNNING TIMES ON THE MOVIELENS 1M. TAGS ARE THE MOVIES' THE COMBINATION OF GENRES AND ACTORS/ACTRESSES. – MEANS NO MEASUREMENT.

No.	Train-test sets	RMSE	Running times (ms)
1	u1.base	–	726,754.79
	u1.test	1.0354	
2	u2.base	–	811,877.18
	u2.test	1.0304	
3	u3.base	–	735,897.88
	u3.test	1.0323	
4	u4.base	–	779,554.22
	u4.test	1.0323	
5	u5.base	–	693,466.31
	u5.test	1.0368	
Average score		**1.0334 ± 0.0025**	**749,510.07 ± 46,465.86**

Templage Language [43]. A controller is written to control the interaction between Model and View. When a user requests, the controller processes the user's request using Model, View, and Template. It acts as a Controller to check if it is available by URL mapping and if the URL is successful. The View will start interacting with the Model and return the Template to the user as Response. The website is written in Django's default SQLite database and it also integrates a lightweight server for application development and testing.

The system provides functionality for two groups of users, e.g. the administrator(s) and the user(s). An administrator performs functions of managing users, users' information, movies, movies' information, databases, and suggestions. Administrators have the highest rights in the system that can perform addition, editing, deletion, and search for movies and users. Users are allowed to register, log in, search for movies and actors/actresses. The overview of our proposed movie recommender system is illustrated in Fig. 1.

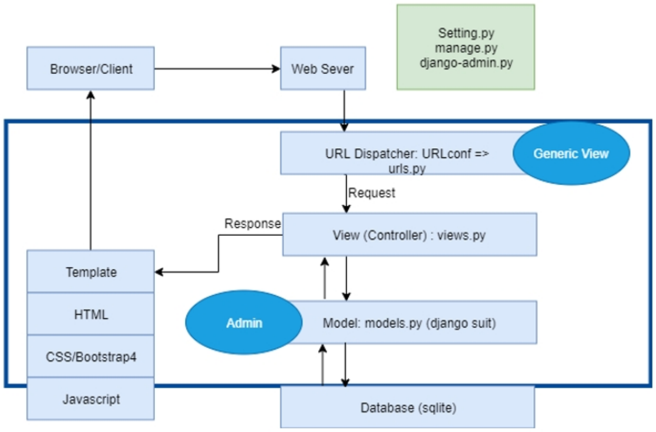


Fig. 1. The design of our movie recommender system.

B. System Implementation

As the implementation of our recommendation system, the authors have deployed a website for our movie recommender system[7]. The application has 15 preliminary features for both users and administrators. The functionality of our website is shown in Fig. 2. The database design is presented in Fig. 3. The website has been developed using Django framework[8] [44] and the relational database management system SQLite[9] [45]. A screenshot of our website can be seen in Fig. 4. The

[7] https://goiyphim.herokuapp.com/
[8] https://www.djangoproject.com/
[9] https://www.sqlite.org/index.html

recommendation function is demonstrated in Fig. 5 where the watched movie is in the main position on the left and its list of similar movies is presented on the right. All the work of the model's training and prediction and the website's deployment are done on a normal laptop. The hardware configurations are the following: Intel Core i5, 12GB of RAM, 240GB high-speed SSD, and Windows 10.

Before using the system, users need to register an account without specifying their preferred movie genre. The interaction with the recommender system can be done through the web interface. By watching any movies and rating them, user profiles are created. A list of recommended movies is generated every time a movie is watched. This system is deployed in real-time scenarios to generate an automatic recommendation.

V. Remarks and Discussion

The experimentation has been conducted on the MovieLens 100K and MovieLens 1M whose tags are missing originally. Remember to note that information of tags is only available in modern variations of the datasets, e.g. MovieLens 10M and MovieLens 20M. The authors interpolate the movie genres and actors/actresses as the tags. The experimental results lead us to believe that the proposed tag interpolation should work properly and yet improve the development of movie recommender systems whose tags are missing. We have achieved better RMSE scores as other approaches running on the tag-available MovieLens datasets [42]. From the experimental results conducted in [42] on a similar movie recommender system, we can agree that our proposed tag interpolation approach is more effective than probabilistic matrix factorization [46], collaborative topic regression [47], factorization machines [48], and regression latent factor model [49].

The RMSE scores are quite similar in all experimental scenarios. Regarding MovieLens 100K, the average score is achieved by 1.1457 ± 0.0126, 1.0556 ± 0.0250, and 1.0556 ± 0.0250 in case of genre tags, actors/actresses tags and the combination of genres and actors/actresses respectively. The running time is super fast in the case of movies' genres, e.g. less than 1 second. In case of MovieLens 1M, the RMSE scores are slightly better than those of 100K variation. the average score is achieved by 1.0364 ± 0.0190, 1.0335 ± 0.0025, and 1.0334 ± 0.0025 in case of genre tags, actors/actresses tags and the combination of genres and actors/actresses respectively. The running times increase through the extension of the number of interpolated tags. The effect is quite understandable that the more data processed, the more times required.

VI. Conclusion

The prevalence of movie recommendation systems has been an indispensable component in a wide range of websites and e-commerce applications. And tag usability is increasing in many recommendation systems, yet appropriate algorithms are available to exploit these tags. This work addresses a simple research question: what if the tags are missing or do not exist in the first place? Therefore, tags can be interpolated from any other characteristics of the movies themselves. Our proposed approach makes it highly convenient for users to get meaningful movie recommendations. Several experimental scenarios have validated the effectiveness of our proposed solution. The significant contribution of the paper is to the MovieLens-based research where previous work has never done on the 100K and 1M variations. As we illustrated in our experimental results, the effects of genres and actors/actresses as interpolated tags have proved the effectiveness and applicability. We have implemented a complete movie recommender system with 15 preliminary functions for both users and system administrators. The interpolated-tags datasets are also available on our Github repository. Future work will focus on the implementation of datasets that emerge the similar characteristics.

References

[1] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016.

[2] P. Melville and V. Sindhwani, "Recommender systems," *Encyclopedia of Machine Learning and Data Mining*, pp. 1056–1066, 2017.

[3] C. He, D. Parra, and K. Verbert, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," *Expert Systems with Applications*, vol. 56, pp. 9–27, 2016.

[4] X. Amatriain and J. Basilico, "Past, present, and future of recommender systems: An industry perspective," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 211–214.

[5] A. C. Rivera, M. Tapia-Leon, and S. Lujan-Mora, "Recommendation systems in education: A systematic mapping study," in *International Conference on Information Theoretic Security*. Springer, 2018, pp. 937–947.

[6] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender systems handbook*. Springer, 2015, pp. 1–34.

[7] B. Smith and G. Linden, "Two decades of recommender systems at amazon. com," *Ieee internet computing*, vol. 21, no. 3, pp. 12–18, 2017.

[8] M. Millecamp, N. N. Htun, Y. Jin, and K. Verbert, "Controlling spotify recommendations: effects of personal characteristics on music recommender user interfaces," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM, 2018, pp. 101–109.

[9] N. Duong-Trung, *Social Media Learning: Novel Text Analytics for Geolocation and Topic Modeling*. Cuvillier Verlag, 2017.

[10] A. Anandhan, L. Shuib, M. A. Ismail, and G. Mujtaba, "Social media recommender systems: review and open research issues," *IEEE Access*, vol. 6, pp. 15 608–15 628, 2018.

[11] N. Duong-Trung and L. Schmidt-Thieme, "On discovering the number of document topics via conceptual latent space," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2051–2054.

[12] N. Duong-Trung, N. Schilling, and L. Schmidt-Thieme, "Near real-time geolocation prediction in twitter streams via matrix factorization based regression," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 1973–1976.

[13] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, p. 13, 2016.

[14] H. Kim and H.-J. Kim, "A framework for tag-aware recommender systems," *Expert Systems with Applications*, vol. 41, no. 8, pp. 4000–4009, 2014.

[15] T. Bogers, "Tag-based recommendation," in *Social Information Access*. Springer, 2018, pp. 441–479.

[16] X. Shi, H. Huang, S. Zhao, P. Jian, and Y.-K. Tang, "Tag recommendation by word-level tag sequence modeling," in *International Conference on Database Systems for Advanced Applications*. Springer, 2019, pp. 420–424.

[17] S. Tang, Y. Yao, S. Zhang, F. Xu, T. Gu, H. Tong, X. Yan, and J. Lu, "An integral tag recommendation model for textual content," 2019.

[18] M. Rawashdeh, M. F. Alhamid, J. M. Alja'am, A. Alnusair, and A. El Saddik, "Tag-based personalized recommendation in social media services," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13 299–13 315, 2016.

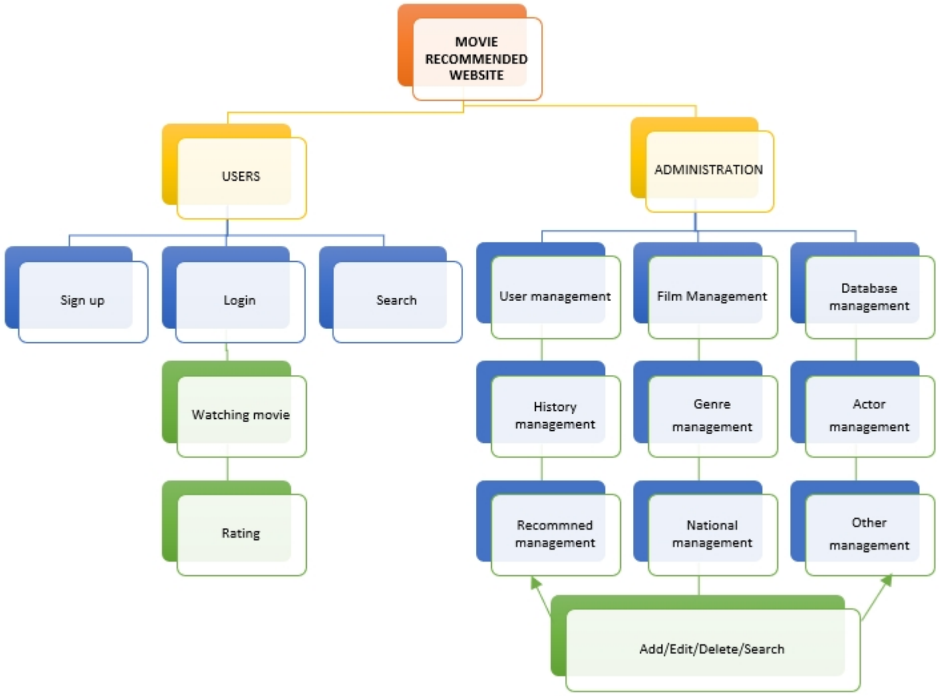


Fig. 2. The functionality of our deployed movie recommender system.

[19] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, and L. C. Jain, "Folksonomy and tag-based recommender systems in e-learning environments," in *E-Learning systems*. Springer, 2017, pp. 77–112.

[20] Y. Zheng, Y. Wang, L. Zhang, J. Wang, and Q. Qi, "A tag-based integrated diffusion model for personalized location recommendation," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 327–337.

[21] D. Lu, X. Liu, and X. Qian, "Tag-based image search by social reranking," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1628–1639, 2016.

[22] Y. Shen, P. Ai, Y. Xiao, W. Zheng, and W. Zhu, "A tag-based personalized news recommendation method," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2018, pp. 964–970.

[23] C. Musto, G. Semeraro, P. Lops, M. De Gemmis, and F. Narducci, "Leveraging social media sources to generate personalized music playlists," in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2012, pp. 112–123.

[24] J.-H. Su, W.-Y. Chang, and V. S. Tseng, "Personalized music recommendation by mining social media tags," *Procedia Computer Science*, vol. 22, pp. 303–312, 2013.

[25] S. Wei, X. Zheng, D. Chen, and C. Chen, "A hybrid approach for movie recommendation via tags and ratings," *Electronic Commerce Research and Applications*, vol. 18, pp. 83–94, 2016.

[26] K. N. Jain, V. Kumar, P. Kumar, and T. Choudhury, "Movie recommendation system: hybrid information filtering system," in *Intelligent Computing and Information and Communication*. Springer, 2018, pp. 677–686.

[27] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 239–249, 2019.

[28] M. F. Alhamid, M. Rawashdeh, M. A. Hossain, A. Alelaiwi, and A. El Saddik, "Towards context-aware media recommendation based on social tagging," *Journal of Intelligent Information Systems*, vol. 46, no. 3, pp. 499–516, 2016.

[29] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–71, 1992.

[30] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 447–456.

[31] P. Mathew, B. Kuriakose, and V. Hegde, "Book recommendation system through content based and collaborative filtering method," in *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. IEEE, 2016, pp. 47–52.

[32] M. Gong, J. Hong, and J. Choi, "Music information and musical propensity analysis, and music recommendation system using collaborative filtering," in *Proceedings of the Korean Institute of Information and Commucation Sciences Conference*. The Korea Institute of Information and Commucation Engineering, 2015, pp. 533–536.

[33] R. Bharti and D. Gupta, "Recommending top n movies using content-based filtering and collaborative filtering with hadoop and hive framework," in *Recent Developments in Machine Learning and Data Analytics*. Springer, 2019, pp. 109–118.

[34] R. Raina, G. Rajaram, H. Ge, J. Pan, and J. Hegeman, "Selecting advertisements for users of a social networking system using collaborative filtering," Jun. 20 2013, uS Patent App. 13/330,502.

[35] H. S. Moon, Y. U. Ryu, and J. K. Kim, "Enhanced collaborative filtering: A product life cycle approach," *Journal of Electronic Commerce Research*, vol. 20, no. 3, pp. 155–168, 2019.

[36] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, p. 19, 2016.

[37] P. Rinearson and E. Speckman, "Relevancy rating of tags," May 31 2011, uS Patent 7,953,736.

[38] A. Azaria, A. Hassidim, S. Kraus, A. Eshkol, O. Weintraub, and I. Netanely, "Movie recommender system for profit maximization," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 121–128.

[39] K. Wakil, R. Bakhtyar, K. Ali, and K. Alaadin, "Improving web movie recommender system based on emotions," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, pp. 218–226, 2015.

[40] Y. Deldjoo, M. G. Constantin, H. Eghbal-Zadeh, B. Ionescu, M. Schedl, and P. Cremonesi, "Audio-visual encoding of multimedia content for

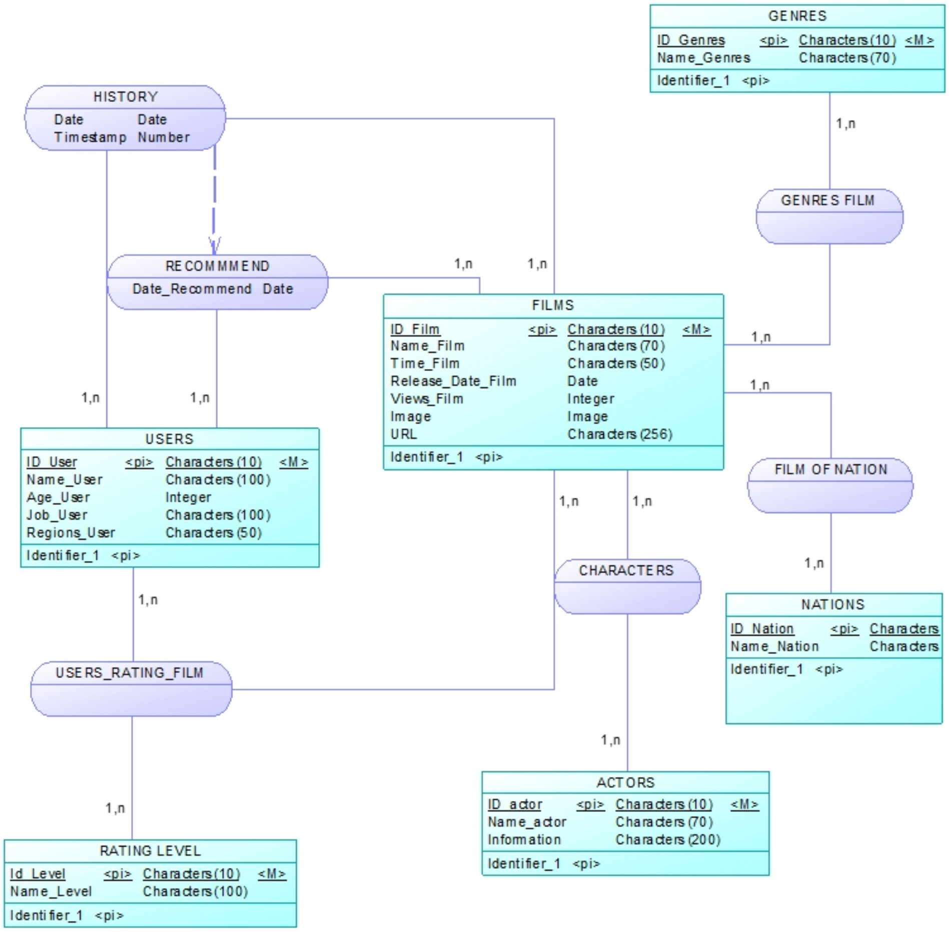

GENRES
ID_Genres <pi> Characters (10) <M>
Name_Genres Characters (70)
Identifier_1 <pi>
1,n
GENRES FILM
HISTORY
Date Date
Timestamp Number
RECOMMEND
Date_Recommend Date
1,n
1,n
FILMS
ID_Film <pi> Characters (10) <M>
Name_Film Characters (70)
Time_Film Characters (50)
Release_Date_Film Date
Views_Film Integer
Image Image
URL Characters (256)
Identifier_1 <pi>
1,n
1,n
1,n
1,n
USERS
ID_User <pi> Characters (10) <M>
Name_User Characters (100)
Age_User Integer
Job_User Characters (100)
Regions_User Characters (50)
Identifier_1 <pi>
FILM OF NATION
1,n
1,n
CHARACTERS
1,n
NATIONS
ID_Nation <pi> Characters
Name_Nation Characters
Identifier_1 <pi>
1,n
USERS_RATING_FILM
1,n
ACTORS
ID_actor <pi> Characters (10) <M>
Name_actor Characters (70)
Information Characters (200)
Identifier_1 <pi>
1,n
RATING LEVEL
Id_Level <pi> Characters (10) <M>
Name_Level Characters (100)
Identifier_1 <pi>

Fig. 3. The database design of our deployed movie recommender system.

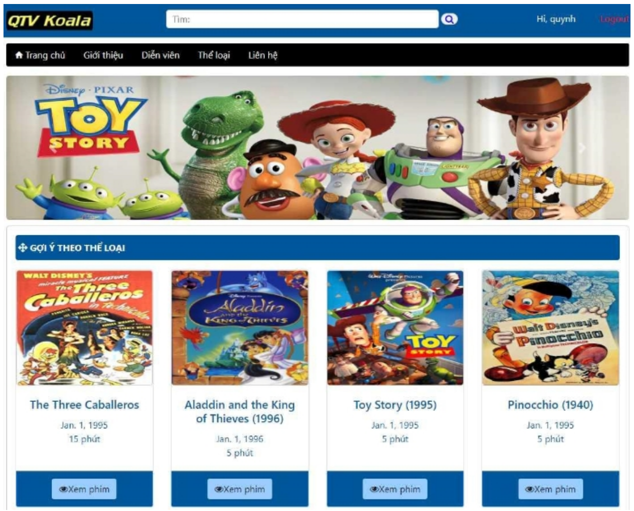
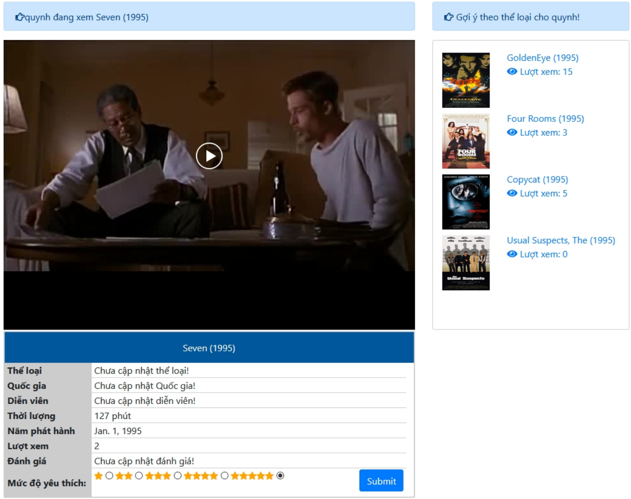


Fig. 4. A logged-in screenshot of our deployed movie recommender system.


quynh đang xem Seven (1995)

Gợi ý theo thể loại cho quynh!

GoldenEye (1995)
Lượt xem: 15

Four Rooms (1995)
Lượt xem: 3

Copycat (1995)
Lượt xem: 5

Usual Suspects, The (1995)
Lượt xem: 0

Seven (1995)

Thể loại	Chưa cập nhật thể loại!
Quốc gia	Chưa cập nhật Quốc gia!
Diễn viên	Chưa cập nhật diễn viên!
Thời lượng	127 phút
Năm phát hành	Jan. 1, 1995
Lượt xem	2
Đánh giá	Chưa cập nhật đánh giá!
Mức độ yêu thích:	★ ○ ★★ ○ ★★★ ○ ★★★★ ○ ★★★★★ ●

Fig. 5. A watched movie and its list of recommendation on the right side.

enhancing movie recommendations,” in *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018, pp. 455–459.

[41] Y. Deldjoo, M. Schedl, and M. Elahi, “Movie genome recommender: A novel recommender system based on multimedia content,” in *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2019, pp. 1–4.

[42] C. Zhang, K. Wang, E.-p. Lim, Q. Xu, J. Sun, and H. Yu, “Are features equally representative? a feature-centric recommendation,” in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[43] C. Burch, “Django, a web framework using python: tutorial presenta-

tion," *Journal of Computing Sciences in Colleges*, vol. 25, no. 5, pp. 154–155, 2010.

[44] N. Idris, C. F. M. Foozy, and P. Shamala, "A generic review of web technology: Django and flask," *International Journal of Engineering Information Computing and Application*, vol. 1, no. 1, 2019.

[45] S. Bhosale, T. Patil, and P. Patil, "Sqlite: Light database system," *International Journal of Computer Science and Mobile Computing*, pp. 882–885, 2015.

[46] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[47] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 448–456.

[48] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[49] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 19–28.