C2P: FROM COMMUNITY TO PERSONALITY RECOMMENDATION SYSTEM WITH APPLICATION TO MOVIE RECOMMENDATION

Dat Ngo-Thanh Nguyen, Thang Phuoc Nguyen, Luong Ngoc Nguyen, Kiet Van Nguyen

Faculty of Information Science and Engineering, University of Information Technology,
Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
{21522923,21522590, 21522311}gm.uit.edu.vn
kietnv@uit.edu.vn

ABSTRACT

With the continuous development in the field of recommendation systems, the provision of personalized product recommendations tailored to individual users' preferences has become increasingly accurate. However, limitations arise when users rate too few items or are new users. At this point, we lack sufficient information to personalize recommendations for these users, as well as any connections to anchor them to a specific community. To overcome this issue, we proposed the architecture **C2P** (From **C**ommunity **t**o **P**ersonality Recommendation System). This architecture provides a way to tackle the above problem, comprising two main stages. Firstly, aggregating the average ratings of all users to create a Community Model through fine-tuning. Subsequently, parallel fine-tuning is conducted for each user using their labeled ratings from previously rated items, thus uncovering their unique Personality traits. This architecture utilizes low-resource techniques, representing users solely based on item features rather than user-specific information. C2P surpasses several existing outstanding models, achieving the best results with MSE = 0.893 and RMSE = 0.940 on The Movies Dataset. Our architecture is available publicly for research purposes[1].

1 INTRODUCTION

The data explosion in the digital age has created ample opportunities for recommendation systems, aiding users in efficiently searching and discovering information. These systems not only facilitate exploration of new products/services but also enhance user interaction with the platform. Through personalization, recommendation systems elevate user experience, drive conversion rates, and increase sales volume.

Recommendation systems play a pivotal role in delivering a tailored digital experience, benefiting both users and businesses across diverse domains. For instance, in music Song et al. (2012); Chen and Chen (2001), they offer personalized listening experiences and aid in discovering new music based on user preferences, while in movies Agrawal and Jain (2017); Sharma and Dutta (2020), they streamline the selection process by suggesting films aligned with user preferences.

Throughout their evolution, recommendation systems have challenged researchers to develop increasingly sophisticated models, ranging from simple to complex. Collaborative Filtering (CF) and Content-Based Filtering (CBF) recommender systems have seen notable advancements, as evidenced by seminal works such as those by Sarwar et al. (2001), Pazzani and Billsus (2007), and Lops et al. (2011). Hybrid approaches, like those proposed by Strub et al. (2016), have emerged to harness the strengths of multiple techniques and address their respective limitations.

[1]Code available at https://github.com/Sonny-Inkai/C2P-FROM-COMMUNITY-TO-PERSONALITY-RECOMMENDATION-SYSTEM

The integration of deep learning techniques into recommendation systems, as demonstrated by Zhang et al. (2019), has ushered in a new era of breakthroughs. Leveraging neural networks, these methods extract intricate patterns from vast and diverse data sources, leading to personalized and context-aware recommendations. Architectures such as Neural Collaborative Filtering (NCF) He et al. (2017), Factorization Machines (FM), and Convolutional Neural Networks (CNN) Han et al. (2021) have become indispensable tools for creating robust and adaptable recommendation models.

Although there are several prominent methods that utilize the relationship between users and their surrounding community Mao et al. (2021); Darban and Valipour (2022), or personalize users based on their history or behavior Cui et al. (2020); Gupta et al. (2020); Dhelim et al. (2020), limitations arise when users rate too few items or are new users. At this point, we lack sufficient information to personalize recommendations for these users, as well as any connections to anchor them to a specific community. This serves as the inspiration and rationale for this paper.

The idea is to first aggregate the average ratings of all users to create a Community Model through specific fine-tuning. Subsequently, parallel fine-tuning is performed for each user using their labeled ratings from previously rated items, thereby revealing their unique personality. This combination is expected to overcome the aforementioned challenges.

This architecture can be described as "Making the most of what you have" compared to approaches by Rashed et al. (2019); Ugla et al. (2020); Darban and Valipour (2022). Alternatively, these methods utilize user-side information to establish connections between them. However, in practice, obtaining such data can be difficult, or we may only have access to synthetic data during the training process. Instead, the C2P architecture focuses on leveraging item information to represent users. Without using any additional information, our C2P achieves the smallest RMSEs, even outperforming models augmented by side information.

Based on the above points, we organize the information of this paper as follows: Section 2 covers Related Works, our proposed model architecture is described in Section 3, Section 4 discusses the Dataset utilized in this paper. Section 5 details Experimental Results, and Section 6 Conclusion of the paper.

2 Related Works

This section introduces how a content-based recommendation system is applied and focuses on some techniques to learn textual information of an item to predict its rating.

2.1 Content-Based Recommendation System

Content-based recommendation system Javed et al. (2021); Pérez-Almaguer et al. (2021) focus on recommending items to users based on how similar those items are to things the user has liked before. These work by analyzing the features and descriptions of items, allowing them to tailor recommendations to each user's unique preferences, and use that to represent for a user, see that in Figure.

The core idea is to build a user profile by identifying common characteristics in the items a user enjoys. This profile is then used to find other items with similar features. A key strength of content-based recommenders is their ability to capture specific user interests and suggest unique items that might not be popular with everyone.

Additionally, unlike collaborative filtering which recommends items based on similar users, content-based filtering uses item attributes, making it suitable for various data types, especially text.

After encoding the information of an item into a vector, we can use these vectors to calculate similarity between items, which allows us to estimate ratings for unrated items based on their similarity to rated ones. One commonly used measure of similarity is cosine similarity. Cosine similarity quantifies the similarity between two items regardless of their sizes. Mathematically, each item is represented as a vector in a multi-dimensional space, and then the cosine angle is computed between the two vectors. Smaller angles indicate higher similarity:

$$\cos(a, b) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2} \times \sqrt{\sum_{i=1}^{n}(b_i)^2}}$$

In content-based recommendation systems, cosine similarity is often applied to a feature vector to identify the most similar item. This approach is widely used in various domains, such as determining profile similarities for recommending social tags Cantador et al. (2010) or recommending news articles Ahn et al. (2007).

Furthermore, we explore several deep learning-based techniques that learn latent features of an item and utilize them to predict ratings for other items.

2.2 Modeling Textual Content

Content-based information is associated with users and items, such as textual descriptions, multimedia descriptions, and user social networks. This information can be used to improve the recommendation system's understanding of users' preferences and items' characteristics.

One technique for modeling textual content is word embeddings, which was proposed to leverage word embedding techniques for better content recommendation Catherine and Cohen (2017); Fan et al. (2019); Liu et al. (2020); Zheng et al. (2017); Lee et al. (2016); Kim et al. (2016), which are numerical representations of words that capture their meaning and relationships to other words. These embeddings can be used as input to neural network models that learn to predict user ratings for items.

Another technique is attention models. Attention mechanism has also been widely used in content enriched recommender systems. Given textual descriptions of an item, attention based models have been proposed to assign attentive weights to different pieces of content, such that informative elements are automatically selected for item content representation Cheng et al. (2018); Gong and Zhang (2016); Wang et al. (2020); Seo et al. (2017); Li et al. (2016); Qi et al. (2020); Lee et al. (2020); Qin et al. (2019). For example, given a tweet, the attention based CNN learns the trigger words in the tweet for better hashtag recommendation Gong and Zhang (2016). With the historical rated items of a user, an attention model is proposed to selectively aggregate content representations of each historical item for user content preference embedding modeling Zhu et al. (2019); Wu et al. (2019); Chen et al. (2019).

Overall, content-enriched recommendation systems aim to improve the accuracy of recommendations by using a wider range of information about users and items.

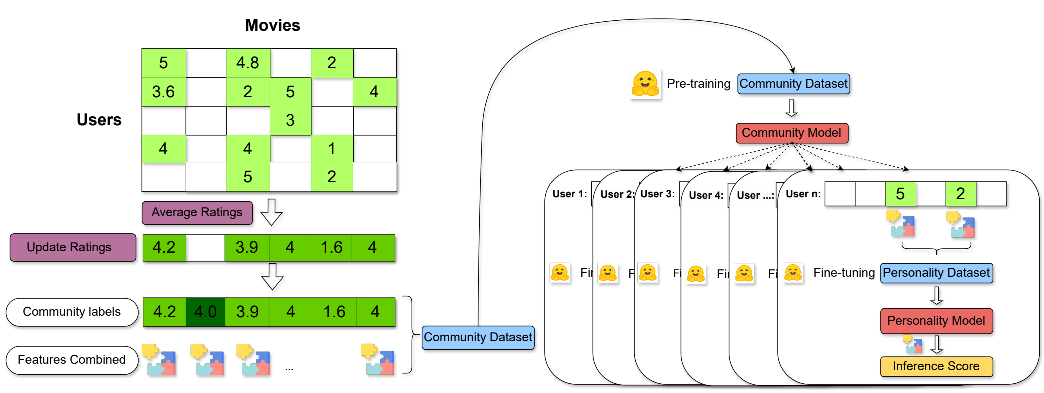


Figure 1: The architecture of the proposed recommendation system involves utilizing Hugging Face's models for a regression task trained on the Community dataset to establish the Community model. Then, this model is fine-tuned to create the Personality model. Predicted ratings are generated by passing features combined of the movie, to the user's personality model.

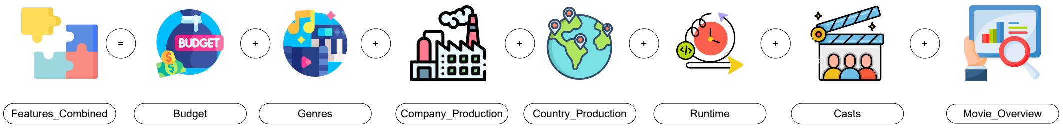


Figure 2: Features represent a movie created by combine multi factors such as budget, genres, production company, production country , runtime, casts, and movie overview.

No.	UserId	MovieId	TmdbId	Features Combined
1	1	1029	11360	812000. Animation. Family. RKO Radio Pictures. Walt Disney Productions. United States of America. 64.0. Sterling Holloway ...
2	3	1378	11967	13000000. Crime. Action. Adventure. Drama. Western. Twentieth Century Fox Film Corporation. Morgan Creek Productions ...

Table 1: Some value about Features Combined of The Movies DataSet.

3 The Proposed Architecture

3.1 Features Combined

The success or failure of a movie depends not only on a single factor but also on a complex network of factors, from human decisions to the constraints of the material world. With the philosophy of 'Making the most of what's available,' we can see that a movie is created by combining a variety of factors such as budget, genre, production company, country of production, duration, cast, and plot summary.

All these factors are combined into a Combined Feature, which can be see in Figure 2, representing the complexity and diversity in the process of producing and marketing a movie. Below are some specific examples of these combined features, illustrating the multidimensionality and creativity in the process of creating a cinematic work, as presented in Table 1. We can see that each movie has its own story, built from different factors, making the world of cinema more diverse and captivating than ever before.

3.2 Community Dataset

Before deciding to watch a movie, many people often want to know the community's opinion about it, ensuring they don't "fall into the trap" of watching a movie that isn't worth their time. This not only helps them save money on tickets and time but also gives them an overall view of the movie's quality and the general consensus of others.

However, there are cases where the number of people participating in the ratings is too large, causing some individuals to hesitate to express their personal views. This is particularly common when the crowd's ratings seem impressive and have a significant influence. Studies such as "The multidimensional wisdom of crowds" Welinder et al. (2010), "The effects of social influence on user acceptance of online social networks" Qin et al. (2011), and "Measuring peer group effects: A study of teenage behavior" Evans et al. (1992) are important references that help us understand this phenomenon and how the community influences the decisions of individuals.

In our architecture, we incorporate the community aspect by utilizing the average rating of all users to evaluate the quality of each movie, which will be used as labels for the movies. The Feature Combined will be used to represent each movie, serving as input. This approach assists us in creating a Community DataSet, which reflects the connections and influence of the community on movies, thereby providing crucial information for each viewer's decision-making process.

3.3 Community and Personality Fine-tuning

After collecting the Community Dataset, we proceeded to fine-tune the model pretrained checkpoints with regression task. To tokenize the Community Dataset, we applied the Tokenizer from the Hugging Face library[2], along with the corresponding model checkpoints. As a result, we obtained a fine-tuned model, with weights updated appropriately according to the Community data. We refer to this model as the Community Model, a significant step in this process. With the Community

[2]https://huggingface.co/docs/transformers/index

Model, we can accurately predict ratings for a movie. However, this is just part of the story. To achieve personalization and accurate predictions for a specific individual, we need to take some further steps.

Similar to processing the Community Dataset, to construct the Personality dataset, we need to gather information about the movies that each user has rated. This information is then used to generate Personality labels, and from there, we can create corresponding Feature Combined. The Tokenizer process remains the same as before Fine-tuning the Community Model. Each user will be assigned a copy of the Community Model and Personality Dataset, ensuring that each user has an independent model, and we call it the Personality models.

Once these tasks are completed, we can observe a specific example. For instance, a romantic movie highly rated by the community, but recently, a user has rated similar movies poorly, perhaps because they are currently single. Therefore, they might be suggested a more cheerful movie, such as "Home Alone". This is an illustrative example of how the combination of two important factors, Community and Personality, works.

And that concludes the entire C2P architecture, you can delve into more details in Figure 1.

4 DATASET DESCRIPTION

4.1 THE MOVIES DATASET

To implement the C2P architecture, we utilize one of the most renowned datasets in the field, namely the MovieLens dataset[3]. This dataset contains information on 45,000 movies listed in the full MovieLens dataset. This information includes actors, production crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, vote counts, and vote averages on TMDB.

Additionally, this dataset includes files containing 26 million ratings from 270,000 users for all 45,000 movies. These ratings range from 1 to 5 and were collected from the official GroupLens website.

Dataset Contents This dataset comprises the following files:

- **movies_metadata.csv**: The main data file containing information on the 45,000 movies in the full MovieLens dataset. Information includes posters, backdrops, budget, revenue, release dates, languages, production countries, and companies.

- **keywords.csv**: Contains plot keywords for MovieLens movies, displayed as a JSON object that has been serialized.

- **credits.csv**: Contains information on actors and production crew for all movies, displayed as a serialized JSON object.

- **links.csv**: File containing TMDB and IMDB IDs for all movies in the full MovieLens dataset.

- **links_small.csv**: Contains TMDB and IMDB IDs for a small subset of 9,000 movies from the full MovieLens dataset.

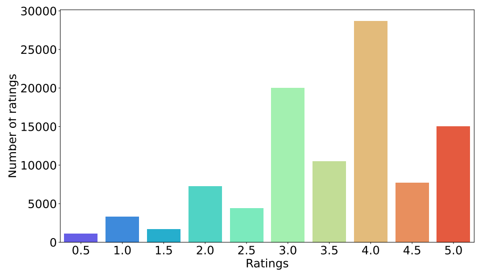
- **ratings_small.csv**: Subset consisting of 100,000 ratings from 700 users for 9,000 movies.

The full MovieLens dataset comprises 26 million ratings and 750,000 tag applications from 270,000 users across all 45,000 movies in this dataset.

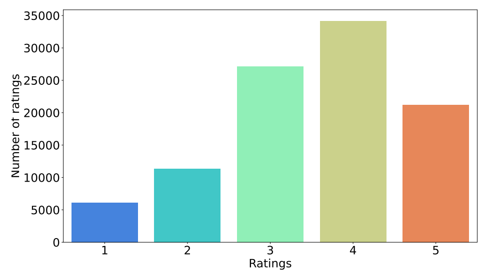
Dataset	Users	Items	Ratings	Density
The Movies DataSet	**671**	**9,025**	**99,810**	**1,648%**
MovieLens 100K	943	1,682	100,000	6,304%

Table 2: The comparison table of data metrics between The Movies DataSet and MovieLens 100K DataSet

[3]https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset



(a) The Movies DataSet



(b) MovieLens 100K

Figure 3: Illustration of rating ranges: (a) The Movies DataSet captures a wider range with 10 rating labels compared to (b) MovieLens 100K, which only has 5 rating labels.

4.2 Data prep-processing

The performance of an architecture depends not only on the model but also on the data. Below are the steps taken to clean this dataset. Our goal is to use the ratings_small.csv file containing 100,000 rows of data for training, evaluation, and testing for the C2P architecture. We need to supplement information such as actors, production crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, vote counts, and vote averages on TMDB from the movie_overview.csv and credits.csv files, through linking movieId and TmdbId in the links.csv file as illustrated in Figure . However, due to reasons such as missing information or film copyright, some tmdbIds have been lost in the movie_overview.csv file. Therefore, we performed the following steps to clean the data:

Step 1: Remove movieIds for which the corresponding tmdbId is missing in the links.csv file and remove the rows containing those movieIds in the rating_small.csv file.

Step 2: Remove rows in the ratings_small.csv file containing movieIds corresponding to ImdbIds without overview information in the movie_overview.csv file.

Step 3: From the ratings_small.csv file, group the data by userId and randomly select 5 movieIds for the test set and 5 movieIds for the evaluation set from each user, while removing these movieIds from the ratings_small.csv file to create the training set.

These steps help us obtain more necessary and higher-quality data. After preprocessing the data, we obtained 671 users and 9,025 movies with a total number of ratings is 99,810 observations, on average, each user rated 148 movies, each user rates at least 20 movie ratings.

In Table 2, we can compare the density of the dataset we created with the MovieLens 100k dataset. Our dataset is less dense but reflects the real-world scenario more accurately, where companies often do not have much available data to use.

5 Experiments and Analysis

5.1 Evaluation metrics

In the field of recommendation systems, evaluating the performance of a model is crucial to ensure that the system can provide accurate recommendations to users. For data-driven models, the two most common metrics used to evaluate performance are Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

MSE measures the average squared difference between the model's predictions and the actual values. For recommendation systems, MSE can be understood as the average discrepancy between the predicted rankings and the actual rankings of the recommended items. A lower MSE value indicates that the model can make more accurate predictions.

$$MSE = \frac{\sum_{(u,i)}(\hat{r}_{u,i} - r_{u,i})^2}{n}$$

RMSE, calculated by taking the square root of MSE, provides a more easily interpretable figure of the model's error magnitude. It indicates the average discrepancy between predictions and actual values in the unit of ranking. RMSE is often preferred because it can be easily interpreted and reflects the degree of prediction bias intuitively.

$$RMSE = \sqrt{MSE}$$

Using MSE and RMSE in model evaluation helps researchers and developers of recommendation systems gain a clear and comprehensive insight into the system's performance, thereby enabling them to adjust and improve the model to ensure that users receive the best possible recommendations.

Models	**MSE**	**RMSE**
WMLFF - Rodriguez and Tommasel (2023)	0.971	0.985
GraphRec - Rashed et al. (2019)	0.961	0.980
IGMC - Zhang and Chen (2019)	0.964	0.981
MG-GAT - Leng et al. (2020)	0.959	0.979
GLocal-K - Han et al. (2021)	0.952	0.976
C2P (ours)	**0.885**	**0.940**

Table 3: Model performances on The Movies Dataset, each experiment was performed 5 times and agveraged.

5.2 Experimental Results

In this section, we focus on the experimental results we have obtained and presented in Table 3. DistilRoberta$_{Base}$ and DistilBert$_{Base-Cased}$ were trained for the Community Model with identical parameters, including *learning_rate = 2e-5*, *weight_decay = 0.01, per_device_train_batch_size = 32*, *num_train_epochs = 6*. For training the Personality, we still used *learning_rate = 2e-5, weight_decay = 0.01, per_device_train_batch_size = 32*, but the difference lies in the number of training personality epochs.

In addition to our own model, we conducted a thorough comparative analysis with several prominent existing models renowned for their performance across various datasets. Noteworthy among these are Glocal-K Han et al. (2021),which has exhibited superior performance on well-established datasets such as MovieLens 1M and Douban Monti[4]. Similarly, MG-GAT propose by Leng et al. (2020) has demonstrated leading performance on the YahooMusic dataset Monti[5], while IGMC developed by Zhang and Chen (2019) has excelled on the Flixster dataset Monti[6]. Additionally, we evaluated GraphRec introduced by Rashed et al. (2019) and WMLFF by Rodriguez and Tommasel (2023) for a comprehensive comparison.

Our experimental results, as illustrated in Table 3, clearly indicate that our model, C2P, equipped with a pretrained DistilRoberta$_{Base}$ model with 9 epochs of Personality Fine-tuning, surpass these existing outstanding models. Notably, C2P achieves impressive results with an MSE = 0.893 and an RMSE = 0.940 on the test set, establishing itself as the top-performing model in this evaluation.

5.3 Features ablation

To analyze the effectiveness of features combined in detail, we conducted ablation experiments where each feature was removed in turn, observing its impact on the MSE metric. Each experiment was performed 5 times, and the average MSE values were calculated. The experiments were conducted on both the train and test sets of The Movies Dataset, using the C2P architecture with

[4]https://paperswithcode.com/paper/glocal-k-global-and-local-kernels-for

[5]https://paperswithcode.com/paper/interpretable-recommender-system-with

[6]https://paperswithcode.com/paper/inductive-graph-pattern-learning-for

DistilRoberta$_{Base}$ employed for the regression task, with 6 personality epochs. Looking at the Figure 4, it can be observed that Genre is used to represent movies well, so when it is removed, MSE increases.

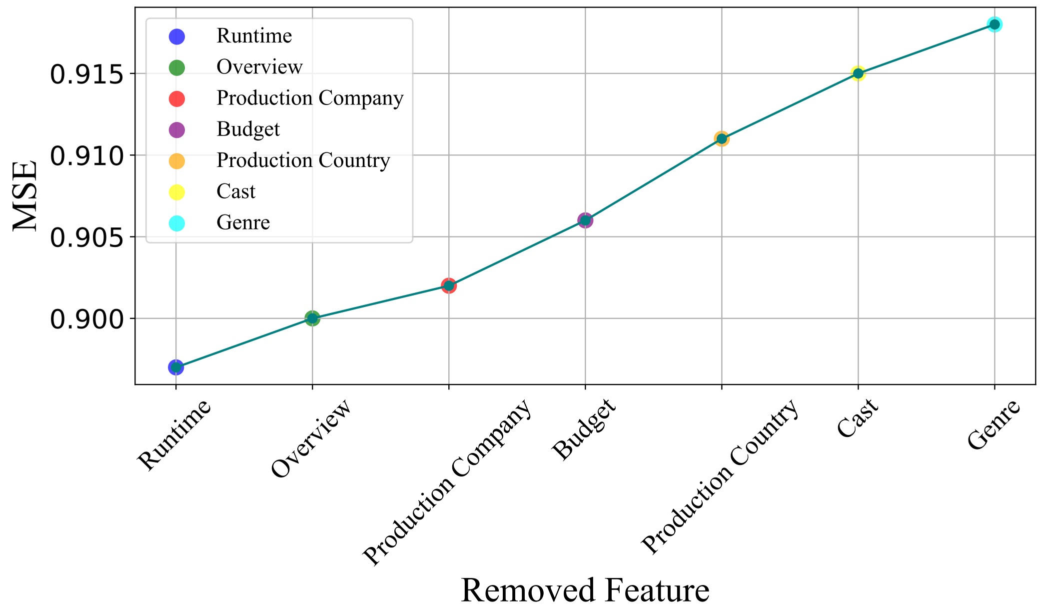


Figure 4: The impact of Number of Personality Epochs on the MSE metric.

5.4 RESULT ANALYSIS

In order to gain deeper insights, we conducted an analysis on the best performing model, the C2P architecture with DistilRoberta$_{Base}$ respectively to regression task with *Personality_epochs = 9*.

How does the number of ratings per user affect performance?

Table 4 presents the analysis of accuracy using the RMSE metric. We compared two users, with userId are 547 and 668: one with the lowest number of ratings at 9 instances, and another with the highest number of ratings at 2376 instances respectively. These ratings were collected from the training set of The Movies Dataset.

UserId	Number of ratings	RMSE
547	2376	1.202
668	9	1.351

Table 4: Model performance with respect to the lowest and highest number of ratings per user.

User with id = 547 significantly outperformed user with id = 668 in terms of the RMSE metric. The test sets for each user were derived from the test set of The Movies Dataset. This indicates that the higher the number of ratings per user, the more the predicted ratings tend to align with the personality of that user, thereby improving the accuracy of recommendations.

For userId = 668, despite having a very low number of ratings, the accuracy remains decent as it was trained on the Community dataset beforehand. Users will be recommended either top-rated products that they have previously evaluated or the best-rated movies from the Community.

6 Conclusion

In this paper, we introduce the C2P architecture. With the idea that we are largely influenced by the community around us, yet each individual has their own opinions and decisions, the C2P architecture is born to improve the accuracy of recommendations for each user. The best results of this architecture also achieve competitive results with other outstanding models. Moreover, this architecture utilizes low-resource, representing users based on their interactions rather than using user side-information, which is often difficult to obtain in practice. This opens up new directions for future research in the field of recommendation systems.

Acknowledgements

We extend our heartfelt appreciation to the anonymous reviewers whose insightful comments and constructive feedback greatly contributed to the refinement and enhancement of this paper.

References

Shreya Agrawal and Pooja Jain. 2017. An improved approach for movie recommendation system. In *2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*, pages 336–342. IEEE.

Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. 2007. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20.

Iván Cantador, Alejandro Bellogín, and David Vallet. 2010. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 237–240.

Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 288–296.

Hung-Chen Chen and Arbee LP Chen. 2001. A music recommendation system based on music data grouping and user interests. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 231–238.

Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 53–60.

Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. 2018. Aˆ 3ncf: An adaptive aspect attention model for rating prediction. In *IJCAI*, pages 3748–3754.

Zhihua Cui, Xianghua Xu, XUE Fei, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. 2020. Personalized recommendation system based on collaborative filtering for iot scenarios. *IEEE Transactions on Services Computing*, 13(4):685–695.

Zahra Zamanzadeh Darban and Mohammad Hadi Valipour. 2022. Ghrs: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*, 200:116850.

Sahraoui Dhelim, Huansheng Ning, Nyothiri Aung, Runhe Huang, and Jianhua Ma. 2020. Personality-aware product recommendation system based on user interests mining and metapath discovery. *IEEE Transactions on Computational Social Systems*, 8(1):86–98.

William N Evans, Wallace E Oates, and Robert M Schwab. 1992. Measuring peer group effects: A study of teenage behavior. *Journal of Political Economy*, 100(5):966–991.

Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2478–2486.

Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *IJCAI*, volume 16, pages 2782–2788.

Udit Gupta, Carole-Jean Wu, Xiaodong Wang, Maxim Naumov, Brandon Reagen, David Brooks, Bradford Cottel, Kim Hazelwood, Mark Hempstead, Bill Jia, et al. 2020. The architectural implications of facebook's dnn-based personalized recommendation. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 488–501. IEEE.

Soyeon Caren Han, Taejun Lim, Siqu Long, Bernd Burgstaller, and Josiah Poon. 2021. Glocal-k: Global and local kernels for recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3063–3067.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Umair Javed, Kamran Shaukat, Ibrahim A Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET)*, 16(3):274–306.

Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM conference on recommender systems*, pages 233–240.

Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2020. News recommendation with topic-enriched knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 695–704.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 957–960.

Yan Leng, Rodrigo Ruiz, Xiaowen Dong, and Alex Pentland. 2020. Interpretable recommender system with heterogeneous information: A geometric deep learning perspective. *SSRN Electron. J*, 10:2411–2430.

Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. 2016. Hashtag recommendation with topical attention-based lstm. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3019–3029.

Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. Kred: Knowledge-aware document representation for news recommendations. In *Proceedings of the 14th ACM conference on recommender systems*, pages 200–209.

Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pages 73–105.

Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. Ultragcn: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1253–1262.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer.

Yilena Pérez-Almaguer, Raciel Yera, Ahmad A Alzahrani, and Luis Martínez. 2021. Content-based group recommender systems: A general taxonomy and further improvements. *Expert Systems with Applications*, 184:115444.

Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. *arXiv preprint arXiv:2003.09592*.

Chuan Qin, Hengshu Zhu, Chen Zhu, Tong Xu, Fuzhen Zhuang, Chao Ma, Jingshuai Zhang, and Hui Xiong. 2019. Duerquiz: A personalized question recommender system for intelligent job interview. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2165–2173.

Li Qin, Yongbeom Kim, Jeffrey Hsu, and Xin Tan. 2011. The effects of social influence on user acceptance of online social networks. *International Journal of Human-Computer Interaction*, 27(9):885–899.

Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Attribute-aware non-linear co-embeddings of graph features. In *Proceedings of the 13th ACM conference on recommender systems*, pages 314–321.

Juan Manuel Rodriguez and Antonela Tommasel. 2023. Weighted multi-level feature factorization for app ads ctr and installation prediction. *arXiv preprint arXiv:2308.02568*.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 297–305.

Nisha Sharma and Mala Dutta. 2020. Movie recommendation systems: a brief overview. In *Proceedings of the 8th international conference on computer and communications management*, pages 59–62.

Yading Song, Simon Dixon, and Marcus Pearce. 2012. A survey of music recommendation systems and future perspectives. In *9th international symposium on computer music modeling and retrieval*, volume 4, pages 395–410. Citeseer.

Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 11–16.

A Ugla, Dhuha J Kamil, and J Khaudair Hassan. 2020. Interpretable recommender system with heterogeneous information: A geometric deep learning perspective. *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)*, 10(3):2411–2430.

Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 836–845.

Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. 2010. The multidimensional wisdom of crowds. *Advances in neural information processing systems*, 23.

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2576–2584.

Muhan Zhang and Yixin Chen. 2019. Inductive matrix completion based on graph neural networks. *arXiv preprint arXiv:1904.12058*.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 425–434.

Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5973–5980.