Enhanced Content-Based Recommendation Using Topic Modelling and Knowledge Graph

Nur Izyan Yasmin Saat[1,*], Shahrul Azman Mohd Noah[1], Masnizah Mohd[2]
[1]*Center for Artificial Intelligence and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia*
[2]*Center for Cyber Security, Universiti Kebangsaan Malaysia, Bangi, Malaysia*
[*]*p88930@siswa.ukm.edu.my; shahrul@ukm.edu.my; masnizah.mohd@ukm.edu.my*

***Abstract*—Content-based (CB) recommendation algorithms recommend items to a user based on items the user liked in the past. CB methodologies have gained attention due to their higher accuracy and transparency and the emergence of new technologies, such as knowledge graphs (KGs), advances in natural language processing (NLP), and sentiment analysis. While most previous studies have mainly focussed on the use of term frequency-inverse document frequency (TF-IDF) and other related enhancements, little work can be found on using KGs in CB recommendations. This paper presents an enhancement of the conventional CB recommendation by incorporating KGs for a movie domain. The graph is constructed using the MovieLens data set, which is augmented with additional features such as actors, directors, and genres. Furthermore, the graph is expanded by incorporating topics derived from latent dirichlet allocation (LDA) extraction. Using the KGs, the proposed approach enhances user profiles by leveraging the interconnected user-movie relationship within a graph structure. The results of the experiments showed that the proposed approach exceeded the tested baselines in terms of precision, recall, and F-score metrics.**

***Index Terms*—Content-based recommender system; Information filtering; Knowledge graph; Topic modelling.**

I. INTRODUCTION

Recommender systems (RSs) are software applications or algorithms designed to provide consumers with personalised suggestions or recommendations. Typically, recommendations are based on user preferences, behaviours, previous interactions, and preferences of other users with similar characteristics. The purpose of RSs is to assist users in discovering items or content that they may be interested in, thus augmenting their overall experience and enabling them to discover new things. A typical RS model consists of two sets and a utility function, with the *User set* containing all users and the *Item* set containing all items that can be recommended to users. The utility function calculates the suitability of a recommendation to a user $u \in User$ an item $i \in Item$, which is declared as $R: User \times Item \rightarrow R_0$, where R_0 is equal to either a real number or a positive integer within a specific range.

Manuscript received 17 November, 2023; accepted 27 January, 2024.
This research was supported by the Ministry of Higher Education under Grant No. FRGS/1/2020/ICT02/UKM/01/1.

RS approaches can be broadly classified into content-based (CB), collaborative filtering (CF), and hybrid [1]. Collaborative-based recommendation implies that the items recommended to a user are based on the preferences from similar user who have liked similar items previously [2]. It assumes that users have interests in content similar to that with which they have interacted with in the past [3]. CF recommendation works by using user feedback in the form of ratings for items in certain domains and taking advantage of similarities in how different users rate things to determine how to recommend an item. While CF recommenders use the user ratings matrix, CB approaches treat all users and items as atomic single units. It works based on the data provided by users, either explicitly or implicitly, which are then used to generate user profiles. In CB recommendation, items are recommended to a user based on the items the user liked in the past (stored as a user profile) [4]. CB filtering techniques rely largely on the information retrieval field, where the metadata and content of the documents are used to select documents relevant to the user's query.

CB algorithms have the main demerit of not being able to consider recommendations for unexpected items [5], while CF-based recommendations may suffer from data sparsity and cold-start problems [5], [6]. Hybrid RS alleviates these issues by unifying the interactions and similarities at the content level [7].

Most studies on RSs focus on CF approaches over CB recommendations due to their better accuracy results. On the other hand, the more recent approach exhibits better efficiency and transparency in relation to user applications [8]. CB methodologies have gained attention due to their higher accuracy and transparency and the emergence of new technologies, such as knowledge graphs (KGs), advances in natural language processing (NLP), and sentiment analysis [9]. Due to these new technologies and the availability of large external knowledge sources, this study focussed on CB recommendation approaches.

Descriptions of user items and profiles are the foundations of CB systems [10]. User reviews are a rich source of item descriptions, to which new and complementary unstructured data concerning an item can be added and updated over time by users interacting with the system. Several studies have focussed on obtaining relevant information from unstructured

data, such as reviews, to improve the quality of both justifications and recommendations. The primary benefits of CB approaches encompass its autonomy in constructing user profiles, as it does not rely on other users to generate recommendations. Additionally, it is conducive to incorporating new items and exhibits transparency. However, it has certain limitations, such as the cold-start problem for users, restricted content analysis, and potential overspecialisation [11], [12].

Applications of CB filtering approaches are mainly on items with significant textual content. Therefore, it is not surprising that CB recommendation algorithms have a strong relationship with information retrieval, since both fields involve techniques to process and present relevant information to users. They share similarities in using text analysis, feature extraction, and similarity measures to connect users with content that matches their preferences or information needs.

The classic term frequency-inverse document frequency (TF-IDF) measure and its variations are still favoured in some literature. However, CB approaches using TF-IDF may encounter a few limitations. One limitation is the inability to account for synonyms, as it does not consider any semantic relationship between words [13]. The TF-IDF algorithm assigns importance to individual terms in a set of words without prioritising any specific feature selection. However, since feature selection is not a fundamental aspect of TF-IDF, it requires one to adapt additional parameters, such as feature selection function and thresholds, to obtain the best possible results [14].

The use of the TF-IDF approach for a large document may also present certain difficulties. Research has demonstrated the inefficiency of TF-IDF for large data sets and its limited ability to account for semantic similarities in language [15]. The studies of the authors in [16] and [17] demonstrate that relying only on TF-IDF is insufficient for accurate recommendations. The limited amount of data available for processing significantly impacts the quality of the recommendation outcomes. The use of semantic techniques can contribute to enhancing the effectiveness of recommendation systems. Implementing TF-IDF can be improved by adopting or combining it with other methods or resources, such as the KG.

A KG is a specific type of graph that is specifically created to help with the comprehension of contextual information. KGs are complex networks consisting of interconnected facts, giving rise to a sophisticated information structure. KG intricate networks comprise interconnected facts, forming a complex structure of information. The information provided encompasses different facets of entities, events, or connections in the real world. It is specifically designed to be comprehensible for both humans and machines [18]. According to [19], the use of KGs offers several benefits. First, it facilitates the creation of semantic representations for items, as their formal and interconnected structure enables systems to retrieve more relevant items. Second, it contributes to improved search efficiency by leveraging advanced representations. Lastly, the analysis of correlations between queries based on the relationships connecting entities in KGs leads to more accurate retrieval results.

Currently, most research in CF-based recommendations addresses the applications of knowledge graphs (KGs) in different ways. For example, KGs have been used to gain a deeper understanding of user preferences [20], address challenges related to data sparsity [21]–[23], the cold-start problem, and improve recommendation coverage [24], [25]. However, the implementation of KGs in CB RSs is still limited.

Therefore, our proposed model aims to enhance CB RSs by incorporating KG. This integration allows for the enrichment of movie information within the graph, where each content element can serve as a feature that contributes to constructing user profiles. This study is similar to the work in [24], where a KG was constructed using an existing data set such as MovieLens. However, this study employs different methodologies and approaches. The proposed work for CF differs significantly from our approach, as we use CB in our work. The authors in [24] employ a spreading method with distinct steps for the recommendation process. This research focusses on improving coverage and addressing the cold-start problem. In our work, our aim is to address the issue of restricted content by employing a variety of features and additional knowledge, such as topic information, to effectively find relevant items for users by selecting features. According to [26], selecting features positively impacts the recommendation process, as it helps reduce the number of data dimensions, eliminate irrelevant data, improve learning accuracy, and improve recommendation results. Finally, the proposed method also aimed to discover whether it would improve precision compared to the traditional CB approach and improve the recommendation.

II. MATERIALS AND METHODS

As mentioned above, the main aim of this study is to exploit KG to enhance the performance of CB RSs. Throughout this study, we used the movie domain represented by the MovieLens data set [27]. The MovieLens data set, however, is mainly suitable for CF approaches and has limited textual content, which can affect the performance of CB approaches [28]. Thus, all items in the MovieLens data set are further enriched with plot synopses which are available in the movie plot synopses with tags (MPST) data set [29]. MPST is a corpus of ≈70 fine-grained tags and their associations with ≈14 K plot synopses of movies. For example, as shown in Fig. 1, the Toy Story movie is enriched with the plot synopsis from the MPST data set by mapping of movieId.

As illustrated in Fig. 2, the proposed recommender model consists of five phases: data preprocessing, topic extraction, user-feature mapping, KG representation, and recommendation. The following describes these phases.

A. Data Preprocessing

In the preprocessing phase, the aim is to construct the lexicon and corpus that encompass important words. The lexicon stores a comprehensive collection of words, each accompanied by a unique numerical identifier. A corpus is a data structure that consists of a list of lists. Each list within the corpus contains a tuple that represents a word ID and its corresponding frequency. Data preprocessing involves tokenisation, stopword removal, lowercase conversion, and lemmatisation.

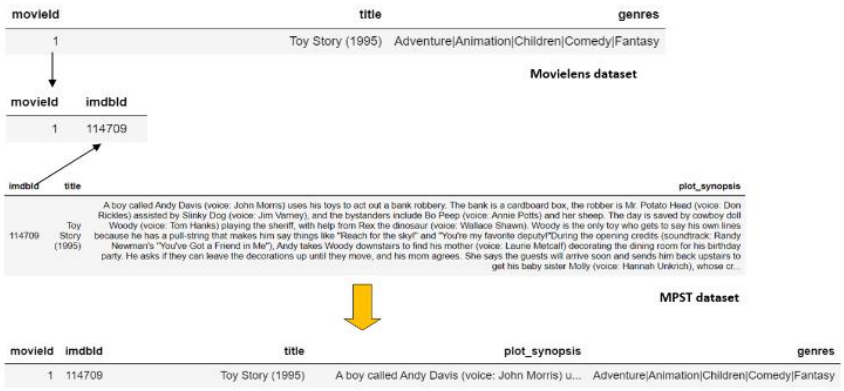

Fig. 1. Example of movie mapping from the MovieLens and MPST data set.

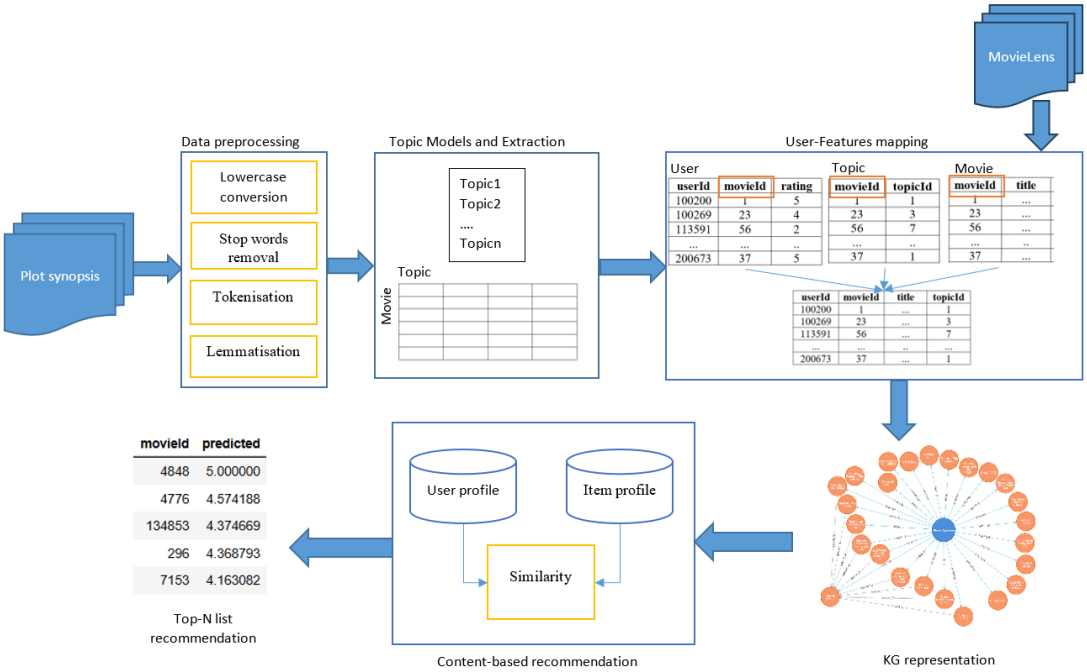

Fig. 2. Proposed recommender system.

Tokenisation is a process that involves splitting sentences into individual words. In addition, all words are converted to lowercase, punctuation marks are removed, and stop words are eliminated. We use the bigram and trigram methods to extract a sequence of *n* words frequently occurring in the corpus. These methods involve referencing two or three words in a sequence before performing lemmatisation. Careful measures have been taken to facilitate the synopsis cleaning process, aiming to minimise noise to avoid data sparseness and obtain high-quality data.

To ensure the quality of the lexicon that only relevant words describing each movie are included, we only consider words where their occurrences in the corpus are larger than 70. We also remove any words that appear in less than ten documents. This stage is crucial because it enables us to discard words deemed irrelevant within the framework of our research.

B. Topic Modelling and Extraction

Topic modelling plays an important role in improving CB RSs, as it helps to understand the main themes and topics within items, whether they are articles, movies, products, or any other form of content. This understanding goes beyond simple metadata such as keywords, enabling a more nuanced representation of content. This study uses the latent dirichlet allocation (LDA) mallet technique to extract topics. The reason for choosing LDA mallet is its use of an optimised Gibbs sampling algorithm, which ensures the generation of efficient topics. The LDA algorithm generates many topics; however, not all are considered relevant for practical use. Thus, we only chose the eight most relevant topics based on the finding of the heuristic value that yields the maximum performance using the LDA approach. Figure 3 shows terms for each topic.

	Terms per Topic
Topic1	family, father, mother, child, love, home, life, wife, daughter, young
Topic2	police, shoot, money, murder, car, gun, man, gang, arrive, drug
Topic3	school, friend, frank, home, boy, game, father, end, student, girl
Topic4	film, show, work, play, love, end, bill, woman, scene, movie
Topic5	order, ship, force, war, team, crew, soldier, agent, destroy, captain
Topic6	attack, body, group, escape, fall, water, head, discover, creature, boat
Topic7	fight, king, death, arrive, village, order, son, lead, father, send
Topic8	room, man, walk, door, car, house, open, start, hear, talk

Fig. 3. Terms for each topic.

As can be seen, most of the terms are related and represent specific topics. For example, the terms in *Topic1* and *Topic2* relate to family and crime, respectively.

C. User-Features Mapping

The user-features mapping phase involves mapping three distinct data sets: a user-rating data set, a data set containing detailed information on movies, and a data set containing topic data associated with each movie ID (movieID). Both the user-rating and movie data sets are sourced from the MovieLens data set. The mapping process merges all three data sets into a single representation that illustrates the relationships between users, movies, and topics. The mapping is performed by matching the *movieId* from each data set.

D. Knowledge Graph Representation

The mapped data set from the previous process is then transformed into a KG, where instances and features related to users, movies and topics are represented as nodes, and the relationships between them are represented as edges.

One of the key advantages of KGs is their ability to facilitate the exploration of relationships between nodes, edges, and their properties. Every node and edge within the network symbolises a unique connection between the characteristics of the user and the object, which in turn serves as a measure of the user's interests.

As illustrated in Fig. 4, the nodes represent entities for Movie, User, Actor, Director, Genre, and Topics. The nodes Actor, Director, Topic, and Genre represent the features of the movies. The edges represent the relation between entities in the KG, where the relation DIRECTED refers to directors who direct the movies, HAS_GENRES refers to genres for the movies, ACTED_IN refers to actors who acted in the movies, RATED contains user ratings of each movie, and HAS_TOPIC refers to the topic for the movie.

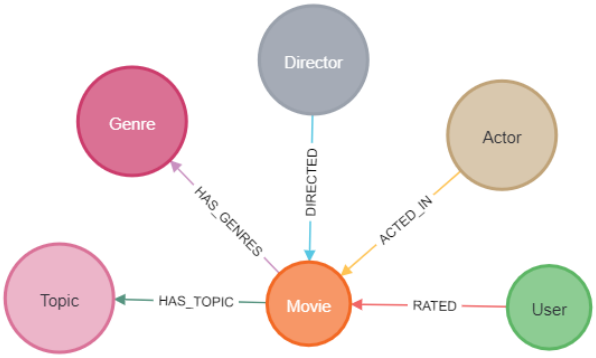

Fig. 4. Detailed view of the data set represented as a knowledge graph.

E. Recommendation

When using traditional CB RSs, the recommendation phase matches user interests with item attributes by utilising user profiles and item representations to recommend appropriate items. A prediction model is developed and used to generate a relevancy score by means of some similarity measures for each item for each user. This score is used to rank and order items to recommend to the user. When using KG, such values are stored back in the graph, which is, in this way, enriched with other data inferred from the item profiles.

The features used to represent movies and user profiles are genres, actors, directors, and movie topics. The movie genres are extracted from the MovieLens data sets, whereas actors and directors are extracted based on the given links in MovieLens data set, and topics are additional data that are extracted and processed from the synopsis provided by the MPST data set. The vector space model (VSM) is used to represent movies and user profiles.

To represent the users' preferences for movies, a relationship called INTERESTED_IN was constructed within the graph. In this case, we assumed that a user is interested in the movie if he/she gives a rating of 3.5 or higher. The features that represent user preferences are determined by their assigned weights. The weight defined for each feature is determined by its frequency of occurrence in the user profiles.

Figure 5 illustrates the result with the weight assigned to each feature in a user profile 113591. The weight is a numerical representation of the level of interest of the user in a particular feature to model their profiles. The approach is used to assess the strength of the relations between user-feature pairs based on their weight. The higher the weight, the stronger the strength relation.

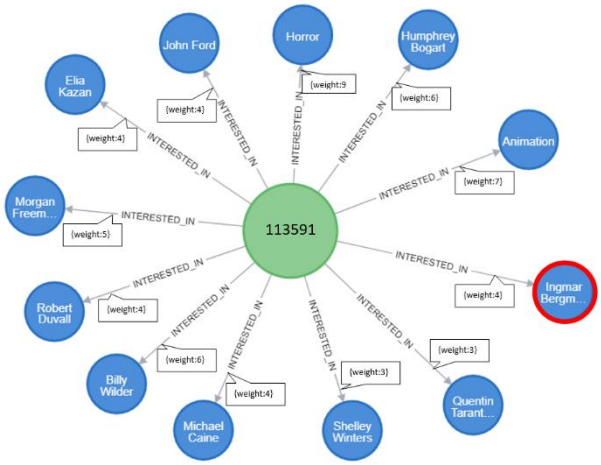

Fig. 5. User profile for user 113591 with features weight representation.

Figure 5 also shows the interest of user 113591 in four films directed by Ingmar Bergman. Consequently, Ingmar Bergman is included as one of the features in the user profile, with a designated weight of 4.

To create the user profile, we set a threshold value of 2 or higher for the appearance of features. Any features that exceed this value will be mapped to user-interest relations on the graph, and their occurrence will reflect their weight. The cypher code below refers to the process of creating the relationship INTERESTED_IN.

```
To create a new relation to construct user profiles
MATCH (user:User)-[r:RATED]->(movie:Movie)-
[:ACTED_IN|DIRECTED|HAS_TOPIC|HAS_GENRES]-(feature)
WHERE r.rating > = 3.5
WITH user, feature, count(feature) as occurrences
WHERE occurrences > 2
MERGE (user)-[:INTERESTED_IN]->(feature)
SET r.weight = occurrences
```

To proceed with the recommendation, another threshold is established to narrow down the weight range for user preferences, which helps the graph to find relevant movies that reflect the user profile. The following is the cypher code

applied to extract movies based on specified features.

```
To get a movie based on features
MATCH (m:Movie)-
[r:HAS_TOPIC|ACTED_IN|DIRECTED|HAS_GENRES]-(feature<-
[i:INTERESTED_IN]-(u:User {userId: $userId})
WHERE NOT EXISTS((u:User)-[]->(m)) AND EXISTS((u)-[]-
>(feature))
WITH m, count(i) as featuresCount, count(r) as relation
WHERE featuresCount > 10
RETURN m.movieId as movieId, m.title as title
```

Figure 6 shows the similarity measuring between users and items, where $feature_n$ refers to movie features that represent actor, director, genres, and topic, while $feature_a$ represents user interest.

Procedure user-item similarity on KG content based recommender system

1. Set $feature_n$ for actor, director, genres and topic
2. For all $alluser_u, movies_m$ which $user_u, rated_r$ >=3.5 **do**
3. For $alluser_u$ if feature > 2, set feature as $feature_a$ **do**
4. Compute cosine similarity for each $user_u$ between $feature_a$ and $feature_n$
5. **end for**
6. Add user's preference for $movies_m$ weighted by feature, if $feature_n$ count >=max featuresCount
7. **end for**
8. **return** top movies, ranked by top user x movie score

Fig. 6. Procedure on user-item similarity of the proposed method.

To calculate the similarity between user-movie pairs, we normalise the weight of each feature. The similarity scores are then calculated using the cosine measure. The equation of cosine similarity is shown in (1), where f_a refers to feature a and f_n refers to feature n

$$\text{cosine\_}similarity(f_a, f_n) = \frac{f_a \cdot f_n}{\|f_a\| \times \|f_n\|}. \quad (1)$$

The top N items that are most similar to the user profile are then submitted for recommendation.

III. RESULTS

To test the effectiveness of the proposed approach, we conducted a series of experiments and benchmarked the results against the conventional TF-IDF weighting scheme and LDA. In this study, we used a MovieLens data set. This data set encompasses approximately 330,975 unique users and roughly 33,832,162 rating entries across a collection of 86,537 movies. For this research purpose, we extended the MovieLens data set using another set of movies from MPST that contains plot synopsis data. We mapped the MovieLens and MPST movie based on the IMDb ID. As a final result, the match movies comprise approximately 12,008 movies with complete synopsis. Figure 7 represents a snippet extracted from the mapping of both data sets.

We use the same plot synopsis throughout all the experiments, but with different approaches. We use the synopsis only to plot the TF-IDF approach. In contrast, in the LDA approach, we used the synopsis to extract it as a meaningful topic and recommend an item based on it. In contrast, our proposed approach uses the topic extracted from the LDA approach with other additional features, such as actor, director, and genre, to perform the recommendation.

	movieId	title	directedBy	starring	genres	plot_synopsis
0	1	Toy Story (1995)	John Lasseter	Tim Allen, Tom Hanks, Don Rickles, Jim Varney,...	Adventure,Animation,Children,Comedy,Fantasy	A boy called Andy Davis (voice: John Morris) u...
1	2	Jumanji (1995)	Joe Johnston	Jonathan Hyde, Bradley Pierce, Robin Williams,...	Adventure,Children,Fantasy	The film begins in 1869 in the town of Brantfo...
2	3	Grumpier Old Men (1995)	Howard Deutch	Jack Lemmon, Walter Matthau, Ann-Margret , Sop...	Comedy,Romance	The feud between Max (Walter Matthau) and John...
3	4	Waiting to Exhale (1995)	Forest Whitaker	Angela Bassett, Loretta Devine, Whitney Housto...	Comedy,Drama,Romance	"Friends are the People who let you be yoursel...
4	5	Father of the Bride Part II (1995)	Charles Shyer	Steve Martin, Martin Short, Diane Keaton, Kimb...	Comedy	The film begins five years after the events of...
...	...	...	...	...	...	...
12470	808	Alaska (1996)	Fraser Clarke Heston	Thora Birch,Vincent Kartheiser,Dirk Benedict,C...	Adventure,Children	Jake Barnes (Dirk Benedict) is flying a plane ...
12471	2883	Mumford (1999)	Lawrence Kasdan	Loren Dean, Jason Lee, Hope Davis, Alfre Wood...	Comedy,Drama	As a relative newcomer to an Oregon town that ...
12472	53574	TV Set, The (2006)	Jake Kasdan	David Duchovny, Sigourney Weaver, Ioan Gruffud...	Comedy,Drama	Idealistic scriptwriter Mike Klein (Duchovny) ...
12473	56012	Evening with Kevin Smith 2: Evening Harder, An...	J.M. Kenny	Kevin Smith, Jason Mewes	Comedy,Documentary	In this second Q&A with Kevin Smith he now ent...
12474	116897	Wild Tales (2014)	Damián Szifrón	Ricardo Darín,Leonardo Sbaraglia,Darío Grandin...	Comedy,Drama,Thriller	The film is divided into six segments. (1) "Pa...

12475 rows × 6 columns

Fig. 7. Movie data with plot synopsis.

TABLE I. RESULTS OF EXPERIMENTS.

Approaches	Precision			Recall			F score		
	@5	@10	@15	@5	@10	@15	@5	@10	@15
TF-IDF	0.723	0.726	0.725	0.036	0.093	0.109	0.069	0.165	0.190
LDA	0.836	0.800	0.769	0.042	0.077	0.116	0.080	0.140	0.202
KG	0.990	0.938	0.871	0.050	0.094	0.133	0.095	0.171	0.231

To evaluate the proposed approach, the data were divided into training and testing in the 80:20, 70:30, and 60:40 ratio. We used the standard metrics of *Precision@k*, *Recall@k*, and *F-measure@k* in all of the experiments, where the equations are as follows, respectively:

$$Precision@k = \frac{relevant}{k}, \quad (2)$$

$$Recall@k = \frac{relevant}{total\_recommended\_item}, \quad (3)$$

$$F-measure@k = \frac{2\times(p@k\times r@k)}{(p@k+r@k)}, \quad (4)$$

where k refers to the k top recommendations. We observed at k = 5, 10, and 15 throughout the evaluation.

The results shown in Table I demonstrate that the proposed approach achieves better results compared to the traditional CB method using TF-IDF and LDA. Our approach demonstrates better precision than other approaches, with an increase of 0.26 and 0.15 compared to the TF-IDF and LDA approaches, respectively.

The experiment used a cut-off point of 100 movies. Recall values indicate that there is a minimal disparity between the Recall@10 for TF-IDF and our proposed approach. This difference arises because both approaches are able to recommend relevant items in the top 10, but our proposed approach has better ranking performance, as evidenced by the precision values.

IV. DISCUSSION

As mentioned earlier, all experiments use the same plot synopsis, but with different approaches. The TF-IDF approach uses traditional term weighting techniques, while the LDA employs the process of extracting terms based on specific topic distributions, while our proposed approach uses the extracted topics from LDA and other features using the KG. On the basis of the result, our proposed approach outperforms the other two approaches in all performance metrics. We can assume that relying solely on the plot synopsis cannot give a good prediction. Certain plots may include noise that can lead to the term "weighting" to focus on unimportant keywords. The outcome indicates the importance of enhancing user features that impact the result of suggestions. The inclusion of crucial information, such as the actor, director, and other relevant features that represent the content of movies, highlights the importance of creating an improved profile preference on the KG.

Figure 8 shows a comparison between the LDA approach and the proposed KG approach when using the MovieLens data set and the topics extracted from the MPST data set for $user_{113519}$.

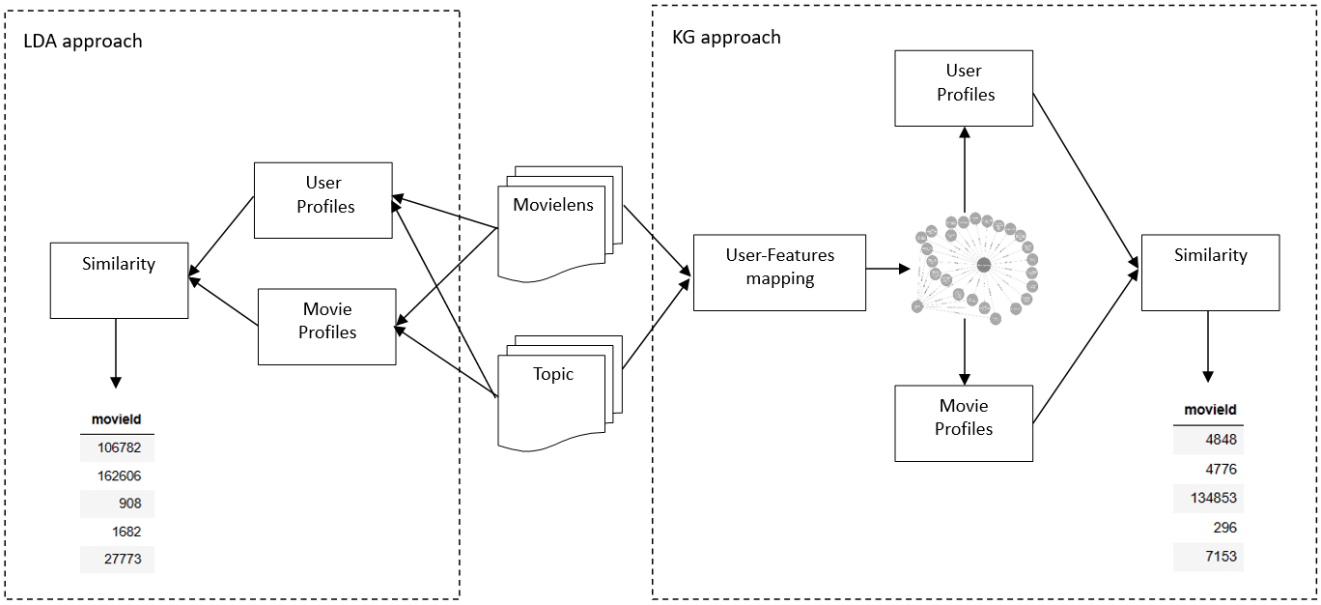

Fig. 8. LDA approach vs. KG approach.

The user-features item mapping constructs a KG that links items and features from the MovieLens data set and topics. The KG is then used to construct the user profile and the movie profile. In this case, the KG has the advantage of utilising all available features to establish connections with other movies based on the relations within the graph.

In the case of LDA, due to its reliance on topic distribution, focussing on the topic might suggest items that the user does not prefer. On the contrary, graphs have the ability to traverse paths to locate additional information. The graph enables the identification of relationships between features that may not be available using conventional CB methods. Our proposed approach uses extracted topics from LDA and features such as actors, directors, and genres. The threshold we set based on the feature occurrences justifies their strong relation within the user features.

V. CONCLUSIONS

This study presents an approach to enhance traditional CB RSs by incorporating KG. The use of the LDA methodology within the framework of the RSs under consideration facilitates the process of extracting relevant topics. Additionally, we incorporate KGs by using the available data set from MovieLens and enhancing it with topic information derived from LDA. The KG enhances user preferences by leveraging data connections derived from attribute relationships within the graph.

The proposed method demonstrates superior performance compared to baselines due to its utilisation of supplementary knowledge sources. It can be concluded that the features play a crucial role in determining the appropriate and precise recommendations based on the preferences of the users. In

our work, we utilise the features that are observed more frequently in the profiles. The more robust the features built into the user profiles, the more accurate the recommendation will be.

However, the effectiveness of the proposed approach is constrained by the accessibility of MovieLens data set and the topics extracted using LDA. To enhance data linking and expand the exploration of items, it is important to consider leveraging existing KGs, such as DBpedia, Wikipedia, or Freebase, in future research works. Additionally, the applications of KGs to overcome the well-known problems of overspecialisation and serendipity in CB recommendations have the potential for further exploration.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] N. A. Osman, S. A. M. Noah, M. Darwich, and M. Mohd, "Integrating contextual sentiment analysis in collaborative recommender systems", *PLoS ONE*, vol. 16, no. 3, p. e0248695, 2021. DOI: 10.1371/journal.pone.0248695.

[2] S. M. Al-Ghuribi and S. A. M. Noah, "A comprehensive overview of recommender system and sentiment analysis", 2021. arXiv: 2109.08794.

[3] N. Jamil, S. A. Noah, and M. Mohd, "Collaborative item recommendations based on friendship strength in social network", *International Journal of Machine Learning and Computing*, vol. 10, no. 3, pp. 437–443, 2020. DOI: 10.18178/ijmlc.2020.10.3.954.

[4] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends", in *Recommender Systems Handbook*. Springer, Boston, MA, 2011, pp. 73–105. DOI: 10.1007/978-0-387-85820-3_3.

[5] M. Marcuzzo, A. Zangari, A. Albarelli, and A. Gasparetto, "Recommendation systems: An insight into current development and future research challenges", *IEEE Access*, vol. 10, pp. 86578–86623, 2022. DOI: 10.1109/ACCESS.2022.3194536.

[6] Z. Sun *et al.*, "Research commentary on recommendations with side information: A survey and research directions", *Electronic Commerce Research and Applications*, vol. 37, art. 100879, 2019. DOI: 10.1016/j.elerap.2019.100879.

[7] E. Cano and M. Morisio, "Hybrid recommender systems: A systematic literature review", *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, 2017. DOI: 10.3233/IDA-163209.

[8] A. L. Zanon, L. Souza, D. Pressato, and M. G. Manzato, "A user study with aspect-based sentiment analysis for similarity of items in content-based recommendations", *Expert Systems*, vol. 39, no. 8, p. e12991, 2022. DOI: 10.1111/exsy.12991.

[9] M. de Gemmis, P. Lops, G. Semeraro, and C. Musto, "An investigation on the serendipity problem in recommender systems", *Information Processing & Management*, vol. 51, no. 5, pp. 695–717, 2015. DOI: 10.1016/j.ipm.2015.06.008.

[10] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. Cambridge University Press, 2010, pp. 51–79. DOI: 10.1017/CBO9780511763113.

[11] U. Javed, K. A. Shaukat, I. A. Hameed, F. Iqbal, T. Alam, T. and S. Luo, "A review of content-based and context-based recommendation systems", *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 3, pp. 274–306, 2021. DOI: 10.3991/ijet.v16i03.18851.

[12] N. I. Y. Saat, S. A. M. Noah, and M. Mohd, "Towards serendipity for content-based recommender systems", *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, nos. 4–2, pp. 1762–1769, 2018. DOI: 10.18517/ijaseit.8.4-2.6807.

[13] J. E. Ramos, "Using TF-IDF to determine word relevance in document queries", in *Proc. of the First Instructional Conference on Machine Learning*, 2003, pp. 1–4.

[14] P. Soucy and G. W. Mineau, "Beyond TFIDF weighting for text categorisation in the vector space model", in *Proc. of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 1130–1135.

[15] K. Shrestha, "Comparative analysis of TF-IDF and word2vec algorithm for content-based job recommendation system", M.S. thesis, Department of Computer Science and Information Technology, Institute of Science & Technology, Tribhuvan University, 2020.

[16] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "Netflix recommendation system based on TF-IDF and Cosine similarity algorithms", in *Proc. of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021)*, 2021, pp. 15–20. DOI: 10.5220/0010727500003101.

[17] R. Huang and R. Lu, "Research on content-based MOOC recommender model", in *Proc. of 2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018, pp. 676–681. DOI: 10.1109/ICSAI.2018.8599503.

[18] A. Barrasa, A. E. Hoddler, and J. Webber, *Knowledge Graphs*. O'Reilly Media, Inc., 2021.

[19] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, "Knowledge graphs: Opportunities and challenges", *Artif. Intell. Rev.*, vol. 56, pp. 13071–13102, 2023. DOI: 10.1007/s10462-023-10465-9.

[20] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences", in *Proc. of the 2019 World Wide Web Conference (WWW'19)*, 2019, pp. 151–161. DOI: 10.1145/3308558.3313705.

[21] Y. Chen, S. Mensah, F. Ma, H. Wang, and Z. Jiang, "Collaborative filtering grounded on knowledge graphs", *Pattern Recognition Letters*, vol. 151, pp. 55–61, 2021. DOI: 10.1016/j.patrec.2021.07.022.

[22] L. Sang, M. Xu, S. Qian, and X. Wu, "Knowledge graph enhanced neural collaborative recommendation", *Expert Systems with Applications*, vol. 164, art. 113992, 2021. DOI: 10.1016/j.eswa.2020.113992.

[23] R. M. Nawi, S. A. M. Noah, and L. Q. Zakaria, "Integration of linked open data in collaborative group recommender systems", *IEEE Access*, vol. 9, pp. 150753–150767, 2021. DOI: 10.1109/ACCESS.2021.3124939.

[24] L. Grad-Gyenge, P. Filzmoser, and H. Werthner, "Recommendations on a knowledge graph", in *Proc. of 1st International Workshop on Machine Learning Methods for Recommender Systems*, 2015, pp. 13–20.

[25] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems", in *Proc. of the 2019 World Wide Web Conference (WWW'19)*, 2019, pp. 3307–3313. DOI: 10.1145/3308558.3313417.

[26] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Impact of feature selection on content-based recommendation system", in *Proc. of 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1–6. DOI: 10.1109/WITS.2019.8723706.

[27] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context", *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, art. no. 19, pp. 1–19, 2015. DOI: 10.1145/2827872.

[28] C. C. Aggarwal, "Content-based recommender system", in *Recommender Systems*. Springer, Cham, 2016, pp. 139–166. DOI: 10.1007/978-3-319-29659-3_4.

[29] S. Kar, S. Maharjan, A. P. López-Monroy, and T. Solorio, "MPST: A corpus of movie plot synopses with tags", in *Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018, pp. 1734–1741.