

Evaluating Recommender Systems for Digital Library Datasets

ÁKOS LÉVÁRDY*, Slovak Technical University, Slovakia

Recommender systems play a vital role in navigating the growing volume of digital content, especially in academic and digital library environments. This work presents a description of different recommendation methods and their characteristics along with a comprehensive evaluation of the content-based recommender systems applied to book datasets. We compare seven text representation methods—TF-IDF, BoW, LSA, GloVe, FastText, BERT, and E5—through offline experiments using datasets composed of academic book paragraphs and Goodreads descriptions. The work describes the various evaluation metrics that are used for recommendation comparison. The evaluation on the tested algorithms and datasets includes similarity, coverage, diversity, and confidence. Our results highlight trade-offs in effectiveness and diversity across models. This paper provides actionable insights for selecting appropriate recommendation strategies for digital library use cases.

CCS Concepts: • **Information systems** → **Content-based recommendation**; *Evaluation of retrieval results*; • **Computing methodologies** → *Natural language processing*; *Learning latent representations*.

Additional Key Words and Phrases: Recommender Systems, Content-Based Filtering, Text Embeddings, Evaluation Metrics, Digital Libraries, Natural Language Processing, Semantic Similarity, Text Representation, Information Retrieval, Offline Evaluation

ACM Reference Format:

Ákos Lévárđy. 2025. Evaluating Recommender Systems for Digital Library Datasets. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference '25)*. ACM, New York, NY, USA, Article 1, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

As Internet and Web technologies continue to evolve rapidly, the amount of information available online has expanded excessively across sections such as e-commerce, e-government or e-learning. To help users navigate this vast sea of content, Recommender Systems (RS) have become fundamental. These systems are not designed just for saving time for users, but they are enhancing the users experience when using the said system, by anticipating their needs and relevant items or topics to discover. They are very effective tools for filtering out the most appropriate information any user would like to find. The primary focus of these recommendations is to predict if a specific user will be interested in the distinct items.

“Item” is the general term used to refer to what the system recommends to users. A RS normally focuses on a specific type of item (e.g., movies, books or news) and accordingly its design, its graphical user interface, and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item. [17]

Author’s Contact Information: Ákos Lévárđy, levardyakos@gmail.com, Slovak Technical University, Bratislava, Slovakia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference '25, Woodstock, NY

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM

<https://doi.org/XXXXXXX.XXXXXXX>

The basic principle of recommendations is that significant dependencies exist between user- and item-centric activity. For example, a user who is interested in a historical documentary is more likely to be interested in another historical documentary or an educational program, rather than in an action movie. [1]

Making decisions is not always easy. People are frequently presented with an overwhelming number of options when picking a product, a movie, or a destination to travel to, and each option comes with different levels of information and trustworthiness.

While there are many situations in which users know exactly what they are looking for and would like immediate answers, in other cases they are willing to explore and extend their knowledge [6].

As RS become increasingly integral to user engagement, the evaluation of their effectiveness has emerged as a critical area of research and practice. Evaluating a RS is inherently complex, involving a multitude of facets and stakeholder perspectives. Evaluations can be broadly categorized into system-centric (focusing on algorithmic performance such as accuracy and ranking quality) and user-centric (concerning user experience and satisfaction). It has been shown that high algorithmic performance does not necessarily translate to user satisfaction, which underscores the need for comprehensive evaluations [22].

Recent frameworks like FEVR (Framework for Evaluating Recommender Systems) categorize the evaluation space by highlighting crucial components such as evaluation objectives, methods (offline testing, user studies, online field experiments), and aspects like novelty, diversity, and fairness. FEVR emphasizes that effective evaluation requires multi-dimensional analysis—from reproducibility and generalization power to business impact and user perception. As recommender systems are deployed in dynamic, real-world environments, evaluations must evolve beyond traditional accuracy metrics to consider real-world usability, context-awareness, and stakeholder equity [22].

2 Understanding Recommendation Systems

The core objective of a **Recommender System** (RS) is to identify and suggest items that are most likely to be relevant or useful to a user. These systems filter and prioritize information by analyzing either the user's own behavior or the preferences of others. In today's data-driven world, where users face an overwhelming amount of content, recommender systems are essential for efficiently guiding users toward desirable options [11]. Particularly in domains such as digital libraries, e-commerce, and media platforms, RS must process large volumes of textual and behavioral data to accurately model user preferences and generate relevant suggestions [21].

Search Engines serve a complementary but distinct function in information retrieval. While RS proactively suggest content tailored to a user's past behavior or inferred interests, search engines respond reactively to explicit queries. They depend on keyword-based retrieval to provide ranked results from a vast index of web resources. Despite differences in operation, both systems are crucial for navigating digital content. Performance speed, relevance, and result quality are essential for user satisfaction in both approaches [3].

A key distinction lies in the personalization strategies: RS adapt to individual behavior, often predicting interests before the user formulates a query, whereas search engines are more general-purpose. Moreover, RS must address challenges like *concept drift*, where user preferences evolve over time, requiring the system to dynamically adjust its predictions [20]. Accurate and timely recommendations can enhance user engagement, satisfaction, and discovery in both academic and commercial contexts [15].

When users are presented with a wide range of choices—such as selecting a movie, book, or product—the lack of initial guidance can lead to choice overload. RS provide a starting point, increasing the likelihood of user interaction and satisfaction by narrowing down options through intelligent filtering.

2.1 Recommendation System Approaches

Recommender systems can be categorized into several methods, each have distinct principles and are suited to different data scenarios, goals and domains. Below, we discuss the major types of recommendation approaches along with their underlying mechanisms, strengths, and limitations.

- Content-Based Filtering:** This approach recommends items that share similar attributes with those a user has previously liked or interacted with. It builds a profile for each user based on their preferences—typically derived from item features such as genre, author, or descriptive keywords—and uses these profiles to match against item metadata ¹. Techniques often involve vector space models, TF-IDF weighting, or embedding-based similarity computation. Content-based filtering performs well in cold-start situations for users with little interaction history but may suffer from limited novelty or over-specialization, often recommending items that are too similar to past preferences [16].

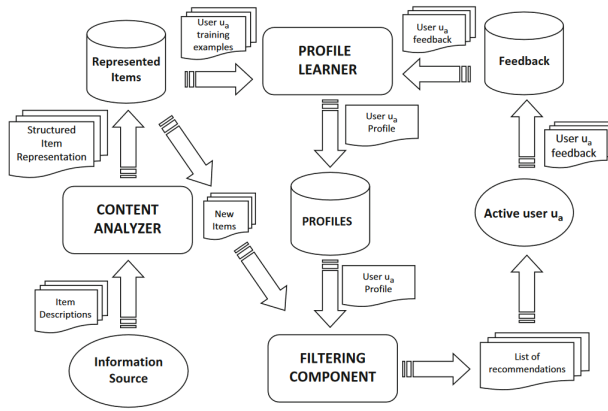


Fig. 1. High level architecture of a content-based recommender [13]

- Collaborative Filtering:** Instead of analyzing item features, collaborative filtering leverages patterns in user-item interactions (e.g., ratings, clicks, purchases) to identify similar users or items. The two primary subtypes are user-based and item-based filtering. User-based filtering recommends items that like-minded users have enjoyed, while item-based filtering finds items that are often co-preferred ². This method is effective for capturing complex user behavior without needing content metadata. However, it requires sufficient interaction data and typically struggles with the cold-start problem for new users or items [14].
- Context-Aware Approaches:** Contextual recommendation enhances traditional models by adding situational variables such as time of day, location, device used or user mood. This enables more adaptive and relevant suggestions. For instance, a travel app might prioritize warm destinations in winter or suggest restaurants near the user's current GPS location. Context can be explicit (manually provided) or implicit (inferred from usage patterns)[11].
- Knowledge Graph-Based Methods:** Knowledge graphs represent entities (items, authors, topics) and their relationships in a structured graph format. Recommender systems can traverse these graphs to discover paths and infer user-item relevance based on semantic proximity ³. These approaches are particularly valuable in domains with rich metadata or ontologies—such as academic libraries or product catalogs—enabling explainable and diverse recommendations. Graph embeddings or path-based reasoning techniques (e.g., TransE, path ranking) are commonly employed to model semantic similarity [12].

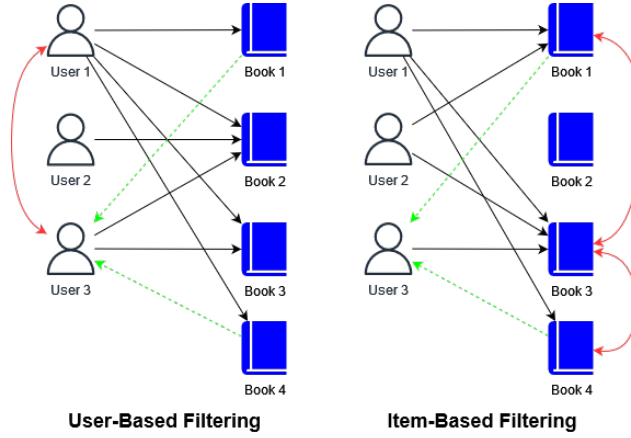


Fig. 2. Illustration of Memory-based CF recommendation

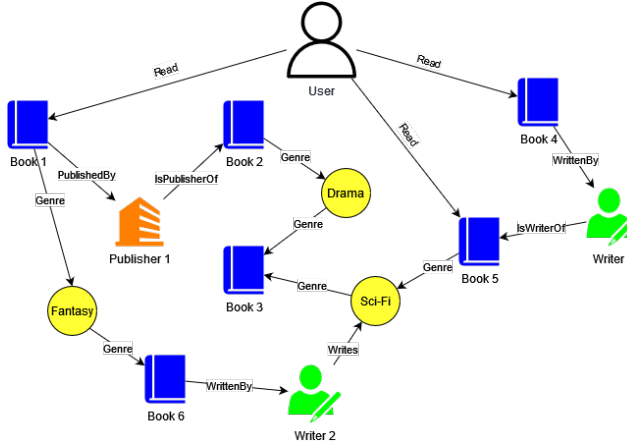


Fig. 3. Illustration of KG-aware recommendation

- **Popularity-Based Models:** These systems recommend items that are globally popular or trending across the user base. They are simple, highly scalable, and often used as default or fallback strategies, especially in the absence of personalization data. While effective in surfacing mainstream content, they ignore individual preferences and may use popularity bias by continually recommending well-known items.
- **Demographic-Based Filtering:** This method relies on user segmentation based on demographic attributes such as age, gender, location, or occupation. Recommendations are generated by identifying patterns or preferences within each group. For example, a platform might recommend young adult fiction to teenage readers or technical manuals to engineering professionals. Though useful in scenarios with limited interaction data, this approach risks stereotyping users and often lacks personalization depth [8].
- **Utility-Based Models:** Utility-based systems estimate the subjective value or utility of items for a user, often combining multiple criteria—such as cost, quality, and relevance—into a scoring function. These models are decision-theoretic in nature and are well-suited to

domains where user decisions depend on multi-attribute trade-offs, such as travel booking or product selection scenarios. The main challenge is defining appropriate utility functions that reflect diverse user preferences [8].

- **Deep Learning-Based Methods:** Deep learning models offer powerful means for capturing intricate, non-linear patterns in user behavior and item characteristics. Architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers, and autoencoders have been applied to recommendation tasks. These methods can process rich inputs such as text, images, or sequences of user interactions. While often achieving state-of-the-art performance, they typically require large-scale data and substantial computational resources for training and inference.
- **Hybrid Approaches:** Hybrid systems combine two or more of the aforementioned techniques to leverage their complementary strengths. Common combinations include blending collaborative filtering with content-based filtering, or incorporating knowledge graphs into deep learning models. Strategies for hybridization include weighted ensembles, cascading (where one model filters candidates for another), or feature augmentation (where the output of one model becomes input for another). Hybrid models are often more robust, especially in complex recommendation scenarios involving cold-start users, sparse data, or diverse content types.

2.2 System Roles and Stakeholder Objectives

The role of a recommender system varies depending on whether it is viewed from the perspective of the service provider or the end-user. For example, a music streaming platform may use RS to boost user engagement, thereby increasing subscriptions or ad revenue, while the user simply seeks enjoyable or novel content.

From the provider's perspective, common goals include:

- Increasing sales through cross-selling and upselling strategies,
- Promoting less popular or long-tail items,
- Enhancing overall user satisfaction and retention,
- Gaining insights into user preferences to improve inventory management or marketing campaigns [17].

From the user's standpoint, recommender systems assist with:

- Discovering high-quality or relevant items,
- Browsing unfamiliar content with minimal effort,
- Exploring item bundles or sequences (e.g., course modules, book series),
- Understanding recommendations through contextual cues,
- Sharing opinions and assisting other users via feedback,
- Evaluating the system's credibility and refining their own profiles [18].

By aligning system functionality with both user experience and business strategy, recommender systems continue to evolve as important tools across numerous digital platforms.

2.3 Challenges in Recommendation Systems

Despite their widespread use and effectiveness, recommendation systems face several critical challenges that affect their performance, accuracy, and user satisfaction. These challenges arise due to the complex and dynamic nature of user behavior, data limitations, and system constraints. Below are some of the most significant difficulties encountered across various types of recommendation systems:

- **Cold-Start Problem:** One of the most common and well-documented issues is the cold-start problem, which occurs when a system must generate recommendations for new users or new items that have little to no historical data. Since most recommendation algorithms rely on past interactions to make predictions, this lack of data can lead to poor or irrelevant suggestions. Solutions often involve hybrid methods that incorporate content-based or demographic data to address the information gap [2].
- **Data Sparsity:** In systems with a vast number of users and items, the user-item interaction matrix—used for collaborative filtering—is typically very sparse. This means that only a small fraction of possible interactions are known, making it difficult to identify patterns or similarities. Data sparsity reduces the effectiveness of similarity-based methods and can degrade overall recommendation accuracy.
- **Scalability:** As recommendation platforms grow to serve millions of users and items, scalability becomes a major concern. Algorithms must be designed to handle large-scale data efficiently, both in terms of computation and memory. Real-time recommendations further increase the demand for scalable, fast-performing systems.
- **Diversity:** Providing a diverse set of recommendations is important for user engagement and product discovery. However, some algorithms tend to recommend only highly similar or popular items, leading to a lack of novelty. This "filter bubble" effect can reduce user satisfaction over time. On the other hand, increasing diversity too aggressively may reduce relevance or precision, creating a trade-off that must be balanced carefully [7].
- **Privacy Concerns:** Personalized recommendations typically require collecting user data such as browsing history, preferences, or even location. While this data enhances prediction quality, it also raises privacy concerns. Users may feel uneasy or distrustful if the system appears to know too much about them, especially when data handling practices are not transparent. Ensuring data protection through anonymization and responsible data use is essential.
- **Serendipity:** Serendipity refers to the ability of a system to surprise users with unexpected but interesting recommendations. While this can enhance user experience, it must be balanced carefully. If too many recommendations are serendipitous and lack clear relevance, users might perceive the system as random or unreliable. A good recommender system must strike a balance between expected results and delightful surprises [5].

These challenges highlight the need for continued innovation in algorithm design, evaluation methods, and user experience optimization. Addressing them effectively requires a combination of advanced techniques, domain-specific adaptations, and a deep understanding of user needs.

3 Experimental Framework for Book Recommendation Systems

To bridge the gap between theoretical foundations and practical application, this section outlines the framework developed to evaluate content-based recommendation systems on digital library datasets. The proposed framework supports a comparative analysis of multiple text representation algorithms and enables systematic experimentation and benchmarking, including preprocessing, model implementation and evaluation methodology.

The primary reason for selecting a content-based filtering approach in this work comes from the characteristics of the datasets involved. The evaluation in this work is based on two distinct datasets: one composed of academic book content segmented at the paragraph level, and another consisting of Goodreads entries containing short textual summaries of general fiction and non-fiction books. Both datasets focus exclusively on short-form text—paragraphs and summaries—rather than full documents. The academic dataset does not contain any form of user feedback or interaction

data, making it unsuitable for collaborative filtering approaches that rely on user-item behavior. While the Goodreads dataset includes some aggregated feedback such as average ratings and number of reviews, it lacks any user-specific interaction history or behavioral traces. Because collaborative filtering techniques require explicit or implicit user-level feedback (e.g., clicks, ratings, co-consumption), they are not feasible in this scenario. For this reason, a content-based filtering approach was adopted, as it relies solely on the textual attributes of items and is well-suited for domains where user data is limited or unavailable.

To enable such content-based recommendations, a range of text representation models were selected, each offering a different level of linguistic or semantic abstraction. This diversity ensures that the evaluation captures performance variations across models that range from shallow frequency-based techniques to deep, contextual embeddings. Classical models like TF-IDF and Bag-of-Words provide interpretable and efficient baselines, capturing the statistical importance of terms within documents. Latent Semantic Analysis was included to explore the benefits of dimensionality reduction in uncovering hidden thematic relationships. Meanwhile, word embedding methods such as GloVe and FastText introduce pretrained semantic spaces that better capture word similarity and context independence, with FastText extending this to subword structures for handling rare terms. Finally, deep transformer-based encoders like BERT and E5 were integrated to evaluate the value of contextual understanding, where sentence-level meaning significantly influences recommendations.

The datasets were preprocessed using standard natural language techniques, including lowercasing, tokenization, and removal of punctuation and stopwords. Paragraphs and summaries were represented as fixed-length vectors using the chosen models, and recommendations were generated by computing cosine similarity between the input text and the candidate corpus. This vector-based similarity approach allowed for consistent comparison across models.

The evaluation was conducted through offline experiments and focused on measuring key system-centric metrics such as similarity, coverage, confidence, and diversity, as well as resource consumption. By combining multiple representation techniques with a unified recommendation pipeline, the framework not only enables reproducible benchmarking but also illustrates how the richness and type of text representation influence the effectiveness of recommendations when user data is absent.

4 Evaluation of Recommender Systems

Recommendation algorithms can be evaluated using various metrics depending on the system's intended purpose and available data. Evaluation methods are broadly classified into two categories: **system-centric** and **user-centric** evaluations [10, 22].

- **System-Centric Evaluation** examines the algorithm's internal performance, using quantitative metrics such as similarity, diversity, and coverage. It is suitable for offline experiments and content-based evaluation.
- **User-Centric Evaluation** involves interaction studies (e.g., surveys, A/B tests) to assess user satisfaction, perceived quality, and usability.

In scenarios where user-item interaction data is unavailable—such as digital libraries or offline textual datasets—traditional metrics like precision or recall are not applicable. In these cases, system-centric evaluation enables performance benchmarking using metrics derived from content similarity and catalog structure.

4.0.1 Overview of Evaluation Metrics. Table 1 provides a consolidated view of recommendation metrics, grouped by purpose and annotated with their data requirements. This overview informs the rationale for the specific evaluation strategy used in this work.

Table 1. Overview of Recommendation Evaluation Metrics

Metric Category	Description	Requires User Feedback?
Rating Accuracy	Measures the difference between predicted and true user ratings (e.g., RMSE, MAE)	Yes
Usage Prediction	Measures success in recommending used items (e.g., Precision, Recall, F1, AUC)	Yes
Ranking Quality	Evaluates ordering of relevant items (e.g., NDCG, MRR, ARHR, Spearman’s ρ)	Yes
Similarity	Measures cosine similarity between input and recommended items	No
Diversity	Captures dissimilarity among recommended items (via score variance or pairwise distance)	No
Confidence	Reflects consistency in similarity scores across top-N recommendations	No
Coverage	Proportion of catalog items that are recommended at least once	No
Novelty	Degree to which recommended items are unfamiliar to users (e.g., based on popularity thresholds)	Partial
Popularity	Proportion of well-known or high-rating items in the recommendations	Partial
Serendipity	Surprise factor—items both unexpected and useful to the user	Yes
Trust	Perceived credibility and reliability of recommendations	Yes
Utility	Value of recommended items to the user in practical scenarios	Yes
Scalability	Ability of the system to perform under growing data volume	No
Robustness	Resistance to noise or malicious input	No
Privacy	Ability to preserve user anonymity or minimize data exposure	No
Risk	Cost or danger of a wrong recommendation (e.g., in health-care)	Yes
Adaptability	System’s responsiveness to changes in user behavior or data	Yes

4.0.2 *Selected Metrics for This Study.* Due to the lack of user-specific interaction data in the datasets used (academic paragraphs and Goodreads summaries without user histories), this study adopts a **system-centric** and **content-level** evaluation strategy. The goal is to analyze the structural quality of the top-N recommendations produced by different text representation models.

The following metrics were selected and computed based on top-10 recommendations per input item:

- **Similarity:** Measures the average cosine similarity between the input and recommended items. A higher score indicates semantic closeness.
- **Diversity:** Quantifies the range of content in recommendations using:
 - *Similarity Variance:* Variance among similarity scores.

– *Pairwise Dissimilarity*: Average of $1 - \text{cosine similarity}$ across pairs.

Higher diversity increases the recommendation breadth and avoids redundancy [19].

- **Confidence**: Indicates stability of recommendations, calculated as:

$$\text{Confidence} = \frac{1}{1 + \text{std_dev}(\text{similarities})} \quad (1)$$

A lower standard deviation in similarity values leads to higher confidence.

- **Coverage**: Assesses how many unique items from the dataset appear in recommendations. A dynamic similarity threshold—based on model-specific behavior—was applied to filter out weak matches.
- **Novelty**: Measures exposure to lesser-known items. Recommendations falling below the 25th percentile in rating, rating count, and review count were classified as novel [4].
- **Popularity**: Evaluates the tendency to recommend mainstream items. Items exceeding dataset averages for rating and in the top quartile for review metrics were labeled popular.

These metrics were chosen for their ability to capture both the quality and the variety of content-based recommendations, especially in domains like digital libraries where personalization data is often sparse. They collectively offer insight into the semantic relevance, range, and uniqueness of each model's outputs, as well as their computational and practical applicability [9].

4.1 Results on Book Descriptions

Figure 4 visualizes each algorithm's normalized performance across the evaluation metrics for the Goodreads book description dataset.

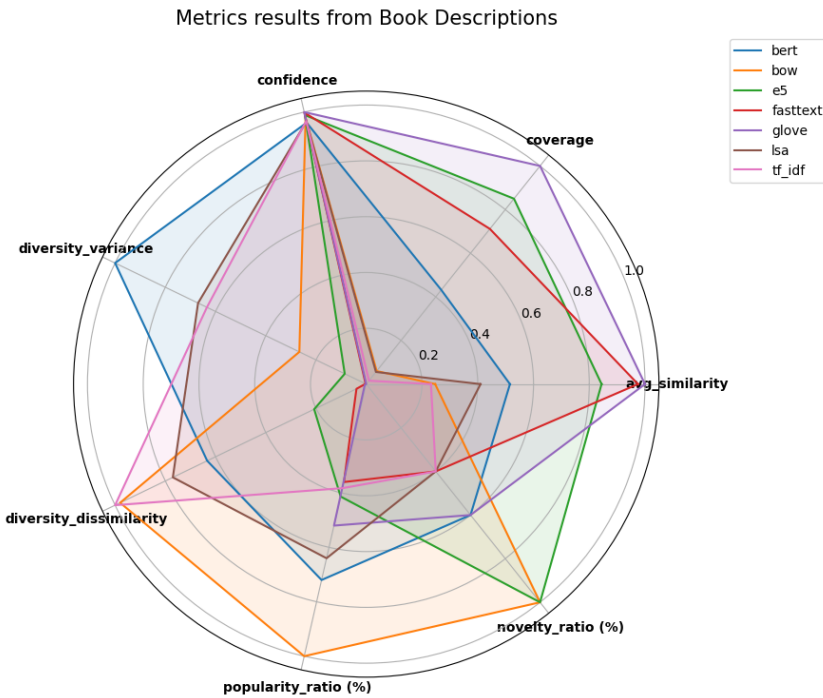


Fig. 4. Metrics Results of Descriptions Visualized

Key observations from the results:

- **FastText**, **E5**, and **GloVe** led in **coverage** and **confidence**, suggesting robust recommendations across a wide item range.
- **BERT** showed highest variance in diversity, while **BoW** and **TF-IDF** excelled in dissimilarity-based diversity.
- **BoW** and **E5** achieved top scores in both **popularity** and **novelty**, indicating an ability to surface both mainstream and niche items.

4.2 Results on Academic Paragraphs

Figure 5 summarizes the evaluation of models on paragraph-level academic book data.

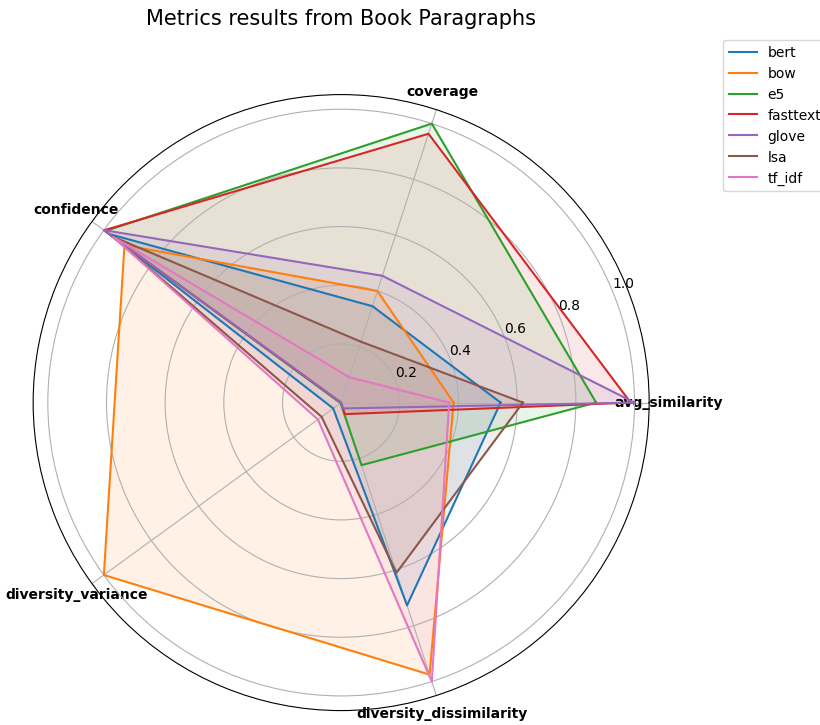


Fig. 5. Metrics Results of Paragraphs Visualized

Key observations:

- **GloVe**, **FastText**, and **E5** scored highest in **similarity** and **coverage**, capturing semantic meaning effectively.
- **BoW** and **TF-IDF** again delivered strong results in diversity, showing their ability to produce heterogeneous recommendations.
- All models had consistently high confidence, indicating stable performance even in more technical or domain-specific content.

5 Conclusion

This study investigated the effectiveness of various content-based recommendation algorithms applied to digital library datasets, with a focus on textual representations. Two distinct datasets were employed: one derived from academic book paragraphs and another from Goodreads, containing

book summaries and associated metadata. The experimental framework compared seven text representation methods—*TF-IDF*, *BoW*, *LSA*, *GloVe*, *FastText*, *BERT*, and *E5*—using cosine similarity to generate recommendations.

A system-centric evaluation approach was adopted to ensure consistent and objective comparison. The metrics—similarity, diversity, confidence, coverage, novelty, and popularity—were chosen to assess the quality, stability, and variability of recommendations in the absence of user interaction data. In addition to recommendation quality, resource usage metrics such as execution time and memory consumption were evaluated to understand the computational trade-offs of each model.

The results demonstrated that dense embedding models such as *GloVe*, *E5*, and *FastText* performed best in terms of similarity, coverage, and confidence, effectively identifying semantically relevant content. On the other hand, traditional models like *BoW* and *TF-IDF* produced higher diversity and unexpectedly achieved high scores in both popularity and novelty, likely due to their reliance on distinctive term overlaps. *BERT* and *E5* delivered strong overall performance but incurred higher computational costs.

Overall, the proposed evaluation framework enabled reproducible benchmarking of content-based recommenders, offering valuable insights into the trade-offs between semantic relevance, diversity, and efficiency. The methodology can be extended to other domains and datasets, supporting further research in content-aware recommendation systems.

5.1 Future Work

Several extensions could enhance the current framework and further explore the potential of recommendation systems in digital library contexts. Incorporating more recent state-of-the-art language models, especially transformer-based architectures, may improve semantic understanding and recommendation quality.

The evaluation could also be broadened by applying the framework to multilingual datasets, other media formats (e.g., abstracts, journal articles), or content from different domains such as education, healthcare, or entertainment. Improvements to text preprocessing—such as advanced normalization, domain-specific tokenization, or entity resolution—could further boost model performance.

Finally, while this study focused exclusively on content-based filtering due to the absence of user interaction data, future implementations may integrate collaborative or hybrid recommendation approaches. These would require user-level behavior data (e.g., clicks, ratings, session logs) to personalize and refine recommendations based on collective preferences. Exploring these dimensions would expand the framework's applicability to more interactive and adaptive recommendation environments.

References

- [1] Charu C. Aggarwal. 2016. *An Introduction to Recommender Systems*. 1–28 pages. doi:10.1007/978-3-319-29659-3 doi:10.1007/978-3-319-29659-3.
- [2] Malak Al-Hassan, Bilal Abu-Salih, Esra'a Alshdaifat, Ahmad Aloqaily, and Ali Rodan. 2024. An Improved Fusion-Based Semantic Similarity Measure for Effective Collaborative Filtering Recommendations. 17, 1 (2024). doi:10.1007/s44196-024-00429-4 doi:10.1007/s44196-024-00429-4.
- [3] Serge Stephane AMAN, Behou Gerard N'GUESSAN, Djama Djoman Alfred AGBO, and KONE Tiemoman. 2024. Search engine Performance optimization: methods and techniques. 12 (2024). doi:10.12688/f1000research.140393.3 doi:10.12688/f1000research.140393.3.
- [4] Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. 2014. *Dimensions and metrics for evaluating recommendation systems*. 245 – 273 pages. doi:10.1007/978-3-642-45135-5_10 doi:10.1007/978-3-642-45135-5_10.
- [5] Ali Taleb Mohammed Aymen and Saidi Imène. 2022. Scientific Paper Recommender Systems: A Review. 361 LNNS (2022), 896 – 906. doi:10.1007/978-3-030-92038-8_92 doi:10.1007/978-3-030-92038-8_92.
- [6] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. 2013. Entity recommendations in web search. 8219 LNCS, PART 2 (2013), 33 – 48. doi:10.1007/978-3-642-41338-4_3 doi:10.1007/978-3-642-41338-4_3.

- [7] Poonam B.Thorat, R. M. Goudar, and Sunita Barve. 2015. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications* 110, 4 (2015), 31–36. doi:10.5120/19308-0760 doi:10.5120/19308-0760.
- [8] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. 12, 4 (2002), 331 – 370. doi:10.1023/A:1021240730564 doi:10.1023/A:1021240730564.
- [9] D. De Nart and C. Tasso. 2014. A personalized concept-driven Recommender System for scientific libraries, Vol. 38. 84 – 91. doi:10.1016/j.procs.2014.10.015 doi:10.1016/j.procs.2014.10.015.
- [10] Asela Gunawardana, Guy Shani, and Sivan Yogev. 2022. *Evaluating Recommender Systems*. 547 – 601 pages. doi:10.1007/978-1-0716-2197-4_15 doi:10.1007/978-1-0716-2197-4_15.
- [11] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. 2017. Context-aware recommender system: A review of recent developmental process and future research direction. 7, 12 (2017). doi:10.3390/app7121211 doi:10.3390/app7121211.
- [12] Saidi Imène, Klouche Badia, and Mahammed Nadir. 2022. Knowledge Graph-Based Approaches for Related Entities Recommendation. 361 LNNS (2022), 488 – 496. doi:10.1007/978-3-030-92038-8_49 doi:10.1007/978-3-030-92038-8_49.
- [13] Cataldo Musto, Marco de Gemmis, Pasquale Lops, Fedelucio Narducci, and Giovanni Semeraro. 2022. *Semantics and Content-Based Recommendations*. 251 – 298 pages. doi:10.1007/978-1-0716-2197-4_7 doi:10.1007/978-1-0716-2197-4_7.
- [14] Mehrbakhsh Nilashi, Othman Ibrahim, and Karamollah Bagherifard. 2018. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. 92 (2018), 507 – 520. doi:10.1016/j.eswa.2017.09.058 doi:10.1016/j.eswa.2017.09.058.
- [15] Simon Philip, P.B. Shola, and Abari Ovy John. 2014. Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library. *International Journal of Advanced Computer Science and Applications* 5, 10 (2014). doi:10.14569/IJACSA.2014.051006 doi:10.14569/IJACSA.2014.051006.
- [16] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2010. *Content-based Recommender Systems: State of the Art and Trends*. 73–105. doi:10.1007/978-0-387-85820-3_3 doi:10.1007/978-0-387-85820-3_3.
- [17] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2010. *Introduction to Recommender Systems Handbook*. 1–35. doi:10.1007/978-0-387-85820-3_1 doi:10.1007/978-0-387-85820-3_1.
- [18] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2022. *Recommender Systems: Techniques, Applications, and Challenges*. 1 – 35 pages. doi:10.1007/978-1-0716-2197-4_1 doi:10.1007/978-1-0716-2197-4_1.
- [19] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How good your recommender system is? A survey on evaluations in recommendation. 10, 5 (2019), 813 – 831. doi:10.1007/s13042-017-0762-9 doi:10.1007/s13042-017-0762-9.
- [20] Yingying Sun, Jusheng Mi, and Chenxia Jin. 2024. Entropy-based concept drift detection in information systems. 290 (2024). doi:10.1016/j.knosys.2024.111596 doi:10.1016/j.knosys.2024.111596.
- [21] Ke Yan. 2024. Optimizing an English text reading recommendation model by integrating collaborative filtering algorithm and FastText classification method. 10, 9 (2024). doi:10.1016/j.heliyon.2024.e30413 doi:10.1016/j.heliyon.2024.e30413.
- [22] Eva Zangerle and Christine Bauer. 2023. Evaluating Recommender Systems: Survey and Framework. 55, 8 (2023). doi:10.1145/3556536 doi:10.1145/3556536.

Received 30 May 2025; revised 12 June 2025; accepted 15 June 2025