

Explainable Semantic Book Recommendation System Using Sentence-BERT Embeddings and Large Language Model-Based Explanations

Duggi Vineela

Master's Student, Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: duggivineela@gmail.com

Dr.V.KRISHNA REDDY

Principle & Professor, Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: vkrishnareddy@kluniversity.in

Abstract— With the rapid development of digital libraries and online book platforms, there is a growing need for recommender systems to be able to (i) identify suitable books and (ii) provide explanations regarding the recommendations. Traditional techniques based on keywords work well, but often do not capture the semantic links if two descriptions use different vocabularies. In this research, we propose an Explainable SBERT embeddings, cosine similarity and Gemini 2.5 Flash-based natural language explanations. The 15,147 books from the Google Books dataset are cleaned and converted to semantic vectors using SBERT. SBERT is used to convert 15,147 books from the Google Books dataset to semantic vectors after they have been cleaned. The most relevant titles, as judged by cosine similarity, are returned to the user for a given book, with the top five returned. Then, Gemini 2.5 Flash writes a brief overview of themes, genres, and narrative qualities. We experimentally analyse sample recommendations, evaluate similarity scores and compare with a TF-IDF baseline. The highest similarity score of SBERT in the case of “Sweet Revenge” is 0.6556 for “Public Secrets”. The average score of the top 5 recommendations is 0.62. The results indicate that semantic embeddings provide more context-aware suggestions and that LLM-generated explanations enhance transparency and interpretability.

Keywords— *Book Recommendation System, Sentence-BERT, Semantic Search, Explainable AI, Large Language Models, Gemini 2.5 Flash, Cosine Similarity, Natural Language Processing.*

How to cite this article: Vineela D, Reddy VK., Explainable Semantic Book Recommendation System Using Sentence-BERT Embeddings and Large Language Model-Based Explanations. Int J Drug Deliv Technol. 2026;16(86s): 1380-1386; DOI: 10.25258/ijddt.16.86s.118

I. INTRODUCTION

Digital libraries, online bookshops and e-learning platforms provide readers incredibly huge collections of books. The diversity has its good points, but it also causes an overload of information making it difficult to find manually. Recommender systems can alleviate this with their ranking of items that are likely to be of the same interest to the user or the context of a chosen item [10]. Content-based and collaborative methods are still prevalent but the term-heavy catalogues require representations that can convey meaning rather than mere term overlap.

Typical book recommenders make use of TF-IDF and cosine similarity. These techniques are easily implementable but also quite simple to compute and yet their similarity rating relies heavily on common terms. If the descriptions of books use different words [7] [11], then the books with similar themes or story structures may be missed. This issue is overcome by transformer based language models that produce contextualised representations of text [1]. Sentence-BERT tweaks BERT to generate sentence-level embeddings with a fixed length that are efficient for similarity search [2].

Explainability is a second challenge. Recommendations which are relevant in some cases might not be sufficiently interpretable if the system does not present an explicit rationale for such a recommendation. ERS try to make recommender systems more transparent, trustworthy, and intelligible for users by offering explanations of what models can be used to explain the output of a recommender system

[12], [13]. Large language models take this a step further by converting evidence of a semantic nature into natural language explanations.

We will be pairing the semantic retrieval with SBERT and explanations from Gemini 2.5 Flash. Our contributions were (1) semantic representation of Google Books descriptions with Sentence-BERT; (2) cosine-similarity ranking and a top-5 retrieval for these representations; (3) comparison to a baseline built using TF-IDF; (4) natural language explanations for each recommendation; and (5) validation on a dataset consisting of 15,147 books across multiple subject categories.

II. RELATED WORK

BERT [1] provided a deep bidirectional context representation of the language. The Sentence-BERT was introduced by Reimers and Gurevych [2] to obtain semantically meaningful sentence embeddings. Several BERT variations were explored for their embedding ability in the recommendation task by Javaji and Sarode [3] and Suhartono and Subalie [6] applied a double stack BERT architecture for content-based book recommendation. The results of these studies highlight the effectiveness of transformer representations for text-rich recommendation problems.

The traditional recommenders are still great sources for a baseline. The TF-IDF method and the cosine similarity are still popular in the area of text retrieval [7], [11]. The

collaborative and hybrid systems integrate the user-item interactions and content features [9,10]. Neural collaborative filtering [21] is a recent study on recommendation systems as is the BERT-based method BERT4Rec [23]. The latter is a graph-based collaborative filtering method [24] and the method LightGCN [25] are also recent studies on recommendation systems. The models can be used to improve personalisation and interaction modelling, but may require user histories which are not present in some pure content-based contexts.

Learning with explanation is an important extra objective. Zhang and Chen [26] explain in a survey explainable recommendation algorithms and demonstrate that explanations can improve transparency and consumer confidence. As models become more language-driven, it is not surprising that language models are increasingly used for reasoning in recommendations, as demonstrated by the XRec [12] frameworks, hybrid SHAP-natural-language [13] frameworks, knowledge-graph retrieval augmented [14] approaches and LLM recommender surveys [15]. We also explore other applications of LLMs in interactive and explainable recommendation, including Chat-REC [27] and LLM-based zero-shot rating [28].

The current research gap lies in either the book recommender only focusing on the semantic retrieval without the user-facing explanation, or the broad explainable recommendation without a compact book-specific semantic pipeline. The suggested method integrates SBERT retrieval, top-k ranking, TF-IDF comparison and LLM-generated explanations in one single workflow.

While a variety of research on semantic recommendation and explainable recommendation exists, the evidence of a clear, book-specific framework that integrates SBERT-based retrieval, an explicit TF_IDF-based baseline, top five ranking, and a layer of LLM explanation is limited in the source literature. The proposed study aims to address this gap by using the Google Books descriptions as the common information source for both baseline and semantic models, making it possible to compare them in a qualitative way directly and as a reader-facing explanation of the final suggestions, at the same time.

B) Research Gap

It was found that in the literature literature on explainability, the quality of the recommendation itself is not enough when the users want to know why an item is being recommended. In recent years, there have been several LLM-based and hybrid explainability techniques that demonstrate how natural language explanations can enhance a ranking model by converting model evidence into meaningful reasoning [12]-[15], [26]-[28]. According to the results, the proposed system is divided into two parts, retrieval and explanation: Sentence-BERT and cosine similarity algorithms are used to retrieve the ranking books, and after the retrieval, the semantic relation is explained with Gemini 2.5 Flash. The design is transparent in the recommendation procedure and the LLM can't just secretly alter the rating.

Furthermore, it has been proven that user-item interactions, ratings or sequential histories are important for significant personalisation in recommender systems [9] [21]-[25]. The signals are helpful but not always available for a new digital-library system. However, a content-based semantic model can operate directly on the book descriptions.

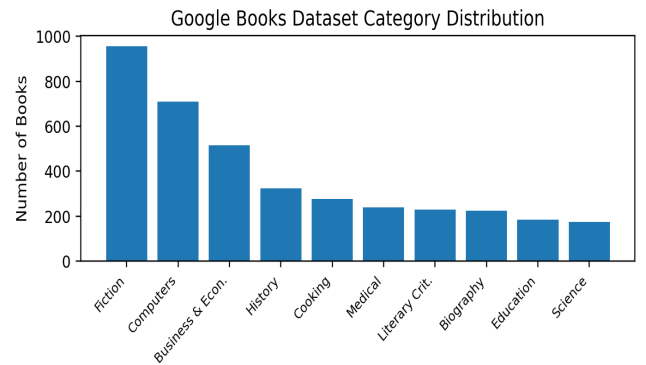
This trait aligns Sentence-BERT well with the current dataset; each recommendation can be directly extracted from the text, without the requirement of a long interaction history.

Our analysis shows a clear shift from lexical matching to representation in context and to explainable suggestion in the research we analysed. The basic TF-IDF and cosine similarity are also valid baselines as they are easily implementable and testable, but are not able to capture the similarities between books using different words for the same topics. This issue is overcome by models based on transformers (e.g., BERT and Sentence-BERT) which express whole descriptions in a semantic vector space, allowing to measure the similarity at the contextual level and not simply at the phrase level [1], [2], [6], [7].

III. MATERIALS AND METHODS

A) Dataset Collection

The trials are conducted with a sample set of 15,147 books from Google Books. These attributes include title, authors, categories, publisher information, ratings and text descriptions. The descriptions of the books were selected as the main representation because they describe the subject matter, themes and background of the book. The data set can be used for assessment in a variety of areas such as: Fiction, Computers, Business & Economics, History, Cooking, Medical, Literary Criticism, Biography & Autobiography, Education and Science.



“Fig. 1. Google Books Dataset Category Distribution”

“Table I. Dataset Category Distribution”

Category	Books
Fiction	956
Computers	710
Business & Economics	516
History	324
Cooking	277
Medical	239
Literary Criticism	229
Biography & Autobiography	223
Education	185
Science	173

B) Data Preprocessing

Preprocessing: Filter out the duplicate records, deal with incomplete book description, normalise text, and only keep

the records with complete book description. This will make sure that an empty or duplicate vector will not influence the similarity search. Sentence-BERT and TF-IDF work with the cleaned descriptions to be able to compare them with each other on the same textual material.

C) Sentence-BERT Embedding Generation

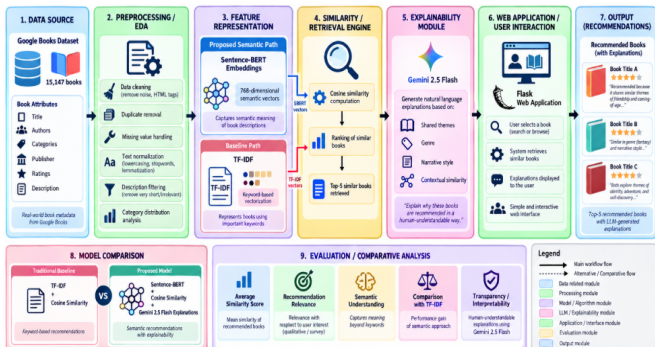
A pre-trained Sentence-BERT model [2] is used to embed each book description after cleaning it up. The model outputs a dense vector of fixed length that captures the context meaning. Unlike bag-of-words or tf-idf representations, dense embeddings can group together similar descriptions even if they use different words, which can be attributed to the similar concepts they describe.

D) Cosine Similarity and Top-K Retrieval

We measure the semantic similarity between the input book, A and the candidate book, B with cosine similarity: $\text{Similarity}(A,B) = (A \cdot B) / (\|A\| \|B\|)$. The candidate books are ranked in order and the book with the highest score is taken away and 5 books of the highest score are returned. Numbers that are closer to 1 indicate that the directions of the two semantic vectors are more similar.

E) LLM-Based Explanation Generation

The book chosen and each book on the list of recommendations is then sent to Gemini 2.5 Flash. The spontaneous attraction towards a quick summary based on common motifs, genres, story elements and historical context. The LLM does not compute similarity ranking, it runs after retrieval. This decouples the ranking from the explanation and the decision to recommend can be traced back to the SBERT and the cosine-similarity step .



“Fig. 2. Architecture of the Explainable Semantic Book Recommendation System”

F) End-to-End Workflow

Here is the whole pipeline: load the Google Books dataset, clean descriptions, create SBERT embeddings of descriptions, choose book, compute cosine similarities with other embeddings, rank books and return top 5, pass in the input description and the selected book description to Gemini 2.5 Flash, and display the top 5 books along with explanation. We employ a TF-IDF pipeline with the same descriptions as a baseline for qualitative comparison.

G) TF-IDF Baseline Construction

The same cleaned descriptions are transformed to sparse TF-IDF vectors for the baseline model. It assigns more credibility terms that are frequent in a particular description, and not frequent in the overall corpus. Cosine similarity is then calculated in the TF-IDF feature space and the five

highest scoring novels are obtained. Same input description guarantees that the difference we find is mostly due to the representation approach and not various preparation procedures. The baseline is deliberately kept simple as it is the widely-used keyword based retrieval method used in academic recommendation demonstrations.

H) Recommendation and Explanation Logic

When querying, the selected title is queried against its precomputed SBERT vector. The similarity with all the other book vectors is then computed and the list is sorted in decreasing order. The chosen element is eliminated from the output and the best 5 are returned. The selected title, the recommended title, and the descriptions of both are provided to the LLM for explanations generation. The model is asked to identify thematic and context elements in the prompt without adding any elements that are not present in the descriptions. It is an architecture that separates evidence retrieval from language generation, and enables a view of the textual rationale, as well as the numerical similarity score.

I) Software and Implementation Environment

The prototype has been implemented using Python 3.x with Google Colab . For data preparation, we rely on Pandas and Numpy, for TF-IDF and cosine-similarity utility functions we use scikit-learn, and on Sentence-Transformers ecosystem to create SBERT embeddings. The only time that you can use Gemini 2.5 Flash is at the explain phase. The embeddings of the book are generated offline, thus eliminating the need to rerun the transformer inference for each and every book recommendation request, and once the embeddings are generated, the online operation involves only vector similarity computation and top-k ranking. The organization permits a later switch to the vector database or approximate nearest-neighbour index as more records are added to the catalogue, which now has 15,147.

J) Evaluation Criteria

The source project evaluates the system based on values of semantic similarity, qualitative evaluation of the recommended titles, comparison with the results of TF-IDF and analysis of the explanations given. Reporting supervised classification accuracy is not appropriate since it is a retrieval task. Likewise, the percentage of correctness of the recommendation is not considered as similarity score. When the cosine is large, the two embeddings are near each other in the embedding space. However, a reliable way to check the true quality of the recommendations is by ranking them or conducting user studies. This difference is relevant when considering the reported mean similarity of 0.62.

IV. EXPERIMENTAL RESULTS

A) Experimental Setup

It is implemented in Google Colab with the help of Pandas and Numpy for data handling, Scikit-learn for TF-IDF and cosine-similarity, Sentence-BERT for semantic embeddings and Gemini 2.5 Flash for explanation generation. We judge whether the recommendations are relevant, similarity of recommendations, quality of sample explanations and qualitative differences between the outputs of TF-IDF and SBERT.

B) Recommendation Results

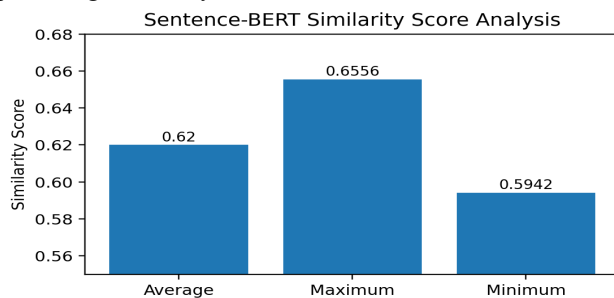
The similarity scores for the top five SBERT recommendations on input book ‘Sweet Revenge’ were

'Public Secrets' with a score of 0.6556, 'River's End' with a score of 0.6518, 'Once Upon a Castle' with a score of 0.6006, 'Genuine Lies' with a score of 0.5965, and 'Carolina Moon' with a score of 0.5942. This indicates that the model is able to identify books on the basis of their themes, which are suspense, romance, mystery, and character driven, rather than their typical description phrases.

Rank	Recommended Book	Similarity
1	Public Secrets	0.6556
2	River's End	0.6518
3	Once Upon a Castle	0.6006
4	Genuine Lies	0.5965
5	Carolina Moon	0.5942

C) Similarity Score Analysis

The mean similarity score for the top five results in the sample is 0.62. The highest is 0.6556 and the minimum is 0.5942. The small range indicates that only 5 books are recommended and their titles are similar. The values should be interpreted as "embedding space similarity" and not as percentage accuracy.



D) Explainable Recommendation Output

Explanations for "Sweet Revenge" and "River's End" suggest both have suspense, mystery, emotionally charged narrative, compelling characters, secrets and unresolved tension. The topics covered in the discussion of two stories "Public Secrets" and "Bed of Roses" are the complicated personal relationships, emotional anguish, trust and growth, and character driven stories. These explanations convert the semantic relationship between books into human understandable language, hence enhancing the rationale of the recommendation.

E) Comparison with TF-IDF

The TF-IDF baseline is mainly based on lexical overlap, while SBERT captures the entire meaning of the descriptions. The TF-IDF model finds the "Sweet Revenge" article as the most relevant result and the SBERT model finds the "Public Secrets" article as the most relevant result. TF-IDF gives the result of "Seize Today" and SBERT gives the result of "Public Secrets" for "River's End". The qualitative examples demonstrate that SBERT can retrieve related themes even though the language is different.

The semantic model is also linked with the explanation layer. The top-five findings determined by cosine similarity are grouped together in a short natural-language justification generated by Gemini 2.5 Flash. Thus, the suggested pipeline does not replace but rather improves the interpretability of the similarity score that generated the recommendation.

These differences are in agreement with the representation employed by each approach. TF-IDF weights phrases and thus emphasizes explicit text similarity. Sentence-BERT not only captures the meaning of the whole

description, but can also locate a book with relevant themes even if the description is phrased differently. It is not a quantitative measure of superiority on benchmark ranking measures (Precision@K, Recall@K or NDCG), but a qualitative proof of semantic relevance in the context of the source evaluation by examples.

The qualitative comparison of both retrieval methods shows that different top-ranked books can be retrieved by the baseline and semantic model, even with the same source descriptions. TF-IDF gives "Sweet Revenge" the output "Blithe Images", and Sentence-BERT gives "Sweet Revenge" the output "Public Secrets". The baseline suggests "The Law is a Lady" for "Genuine Lies", and Sentence-BERT suggests "Montana Sky". This also applies to "River's End", which return "Seize Today" from the TF-IDF and "Public Secrets" from the semantic model.

F) Discussion

The results confirm the main design choice of semantic sentence embeddings for book finding. The similarity is context aware, and remains basic in structure, while Gemini includes human-readable explanations without impacting the ranking itself. This can be helpful when there is no access to user history, as only textual descriptions are needed. In the current evaluation, however, the focus is on content only and no user ratings, click histories, Precision@K, Recall@K, NDCG, or human review of the quality of the explanations are used in a controlled manner. The personalised recommendation performance problem cannot be addressed without dealing with these restrictions.

G) Additional Recommendation Examples

The project offers additional examples of the stability of the semantic model in similar story situations. If you click on "Public Secrets", then you get "Bed of Roses", "The Love of My Life", "Where We Used to Roam", "Sweet Revenge", and "River's End". The Law is a Lady, Abandon Hope: Young Adult Fantasy Suspense, Sweet Revenge, Partners, and Suzanne's Diary for Nicholas are returned. These examples illustrate that it is not a strictly keyword-based system, but also one which can retrieve books that have a similar emotional, suspense, romance or character centered theme.

H) Interpretation of the Similarity Range

In the "Sweet Revenge" example, the max is 0.6556 and the min is 0.5942 and the difference is only about 0.0614. This means that the five returned books are in a relatively dense semantic neighbourhood surrounding the description of the query. (However, a score below 1 can be predicted because the novels are not duplicates and have different narratives and phrasing.) The score distribution can be used to rank in the corpus, but cannot be used with qualitative criteria without empirical benchmarking or user-based validation.

I) Explainability Analysis

The LLM layer is useful in that it converts the evidence of similarity into language that a reader can comprehend. The "Sweet Revenge" to "River's End" analysis stresses on suspense, research, emotionally charged narration, concealed truths, and strong characters. It is from "Public Secrets" to "Bed of Roses", relationships and trust, emotional strife and personal development and character building. Explanations such as this help because they give a thematic reason for a suggestion, instead of just a number

score. The quality of the explanation, however, depends on the factual content provided to the LLM, so production systems need to make sure they limit prompts and reduce unnecessary claims.

J) Practical Advantages

The proposed pipeline has various practical benefits. First, it tackles item cold-start by incorporating a newly launched book right from its textual description before user ratings are accessible. Second, the recommendation is not dependent on the identity of the user, and thus offers recommendation in the presence of contexts where it is not possible to use persistent user profiles. Third, the design is modular: the embedding model, similarity engine, and explanation model can be easily replaced. Lastly, the output is natural and can be displayed in a web application as each result naturally has a title, a similarity score, and an explanation. These attributes make the system very suitable for Digital libraries, educational catalogues and online bookshops with a large amount of textual metadata.

K) Limitations and Threats to Validity

This is an ongoing project of sorts that is proof of concept of the content. It has no readers' profiles, no explicit ratings, no purchase logs, and no sequential behaviour, so it cannot model reader preference except for the reader's current choice of book. The dataset can also be biased in representation of categories, descriptions or text generated by the publisher that has a negative effect on the quality of the embeddings. The analysis is largely based on the recommendations in the sample and similarity scores and therefore may not be used to measure performance in terms of Precision@K, Recall@K, MAP, NDCG, diversity, originality, or serendipity. In addition, LLM explanations are not evaluated for factual accuracy or utility by us. In future work, these restrictions call for a broader investigation.

L) Comparison with Broader Recommendation Models

If enough user-item interaction data is provided, neural collaborative filtering, such as SASRec, BERT4Rec, NGCF and LightGCN [21]–[25] can provide more personalisation. However, the proposed SBERT technique is meant for a different context: recommendation from descriptive text where there is a limited amount of interaction data. A viable large scale system could involve using the score of semantic content and the collaborative score simultaneously. Likely, the LLM explanation layer will condense both content similarity and user-preference evidence. This hybrid architecture would enable the current model to maintain their cold-start capability and would have the advantage of personalisation benefits of collaborative approaches.

M) User-Facing Recommendation Workflow

The design of the interface from the user's point of view is very simple. The reader selects or looks for a known book. The system loads the description and semantic embedding that was already stored for that title. The similarity values for the top 5 related novels and short natural language explanations are displayed. The user can see the score and the explanation at the same time, giving a quantitative signal and an intuitive explanation. The result card could also contain author, category, cover image and link to the full book record in a real Web interface. In the present approach, the approach is content based and thus no profile is required to provide user recommendations.

N) Scalability and Deployment Considerations

The current list of 15,147 books is not too large to perform the cosine-similarity calculation in an academic prototype. Even larger catalogues can be handled with the same architecture using a vector index in place of comparison of all catalog records to approximate nearest-neighbour retrieval. If new titles are added, it is possible to calculate the book embeddings offline and store them with metadata in a database. The online service simply needs to receive the query vector, retrieve the top candidates and pass the result set to the explanation model for the final shortened results. Explanations of commonly requested pairs of books are further reduced in LLM latency and cost since they can be cached. timeout handling, handling of API errors, validation of input and fallback explanation based on categories or shared keywords for external LLM service downtime should also be considered during the production deployment.

O) Explainability, Reliability, and Responsible Use

Natural language explanations only help the usability of the system if they are accurate with respect to the data which their advice is based upon. Therefore, the LLM needs to be restricted to the chosen and recommended descriptions and refrain from stating facts that are not provided in the metadata. A good solution should indicate the similarity score, highlight the themes that are common to both texts to show the areas of connection and should clearly state that the created text is an explanation and not a factual summary of the rest of the literature. This is especially important because generative models can produce eloquent, but unsubstantiated statements. The intelligibility, relevance, succinctness and faithfulness of explanations can be determined through human evaluation. The system should also not be a measure of literary value, but merely a close approximation to literary value of context of descriptions.

P) Future Evaluation Plan

Experimental testing with a larger set of query books could be carried out by gathering relevance judgements from readers or topic experts for a larger set of query books. Finally, the ranked lists can be tested with Precision@5, Recall@5, NDCG@5, Mean Reciprocal Rank, diversity, originality and serendipity. Alternatively, one can do a separate user study comparing three interfaces, namely, TF-IDF suggestions without explanations, SBERT recommendations without explanations, and SBERT recommendations with LLM explanations. This experiment would answer the question of whether the relevance of perceived relevance could be enhanced by the use of semantic retrieval and if the explanations increase trust, understanding and willingness to read recommended titles. It would also be beneficial to record latency and resource usage to quantify the true cost of embedding and inference of generations and LLMs.

Q) System Architecture Interpretation

The architecture might be regarded as two layers playing together. The offline layer is used to prepare the catalogue, which involves cleaning textual metadata and generating one SBERT vector per valid description. These are stored across with identifiers and metadata and thus will not require the complete dataset to be re-encoded for multiple recommendation requests. The online layer is activated on selecting a book. The selected vector is compared with the stored catalogue, the nearest semantic neighbours are sorted

and just the small top-k result set is sent to the explanation module. In this division, the repeated calculation is eliminated and it is easy to scale the system. It also enables reproducibility, as the retrieval choice can be reconstructed from the vectors and similarity scores stored, regardless of the changes to the later LLM explanation. It would be useful in a production scenario to store with the vector index the version of the model, embedding dimension, configuration of the preprocessing and the time stamp in order to track back the recommendations made at different times.

R) *Why Sentence-BERT is Suitable for Book Descriptions*

Book descriptions are more comprehensive and are semantically more dense than keywords. They are defined by paraphrase, not by repeated words and do not repeat the words of the plot, the subject, the place, the style of writing, the target audience, or the mood. Despite using different terms for similar concepts, two descriptions can be weakly related using the bag-of-words representation. Sentence-BERT solves this issue by representing entire sentences and paragraphs in a dense semantic space that represents similar sentences as nearby expressions. This is important for a book recommender, since the purpose of the recommendation is not to find duplicate information, but expressions of similar concepts. At the same time, SBERT is easier than building a fully supervised recommendation model since it may be utilised as pre-trained encoder. This can be useful for an academic prototype if there are no markers and no large scale interaction history of users.

S) *Role of the LLM Explanation Layer*

The LLM is not intended to replace the similarity engine, but is rather an optional feature following the similarity engine. This architecture doesn't rely on the generative model to infer the suggestion list, but lets the user or developer view the retrieval score. Only a small amount of evidence is provided for the explanation model: the selected title and description, the recommended title and description and optionally the similarity score. The reason is a short sentence that identifies common themes or aspects of context. This approach can be further strengthened by using structured prompts to ask that the explanation only refer to evidence that is present in the two descriptions. The application might also display a short 'why suggested' line with a longer explanation displayed behind an expanding control. The result is easy to understand and does not overload the reader with model generated text.

T) *Reproducibility and Future Benchmarking*

The entire experimental pipeline should be consistent with the pre-processing rules, SBERT model name, software versions, random seeds used (if applicable), and the list of query books assessed for future benchmarking. Multiple readers are able to provide multiple annotations for the same evaluation subset on its relevance to the query. It is possible to provide ranking metrics along with inter-rater agreement, for a fair evaluation of semantic retrieval quality compared to TF-IDF and other embedding models. We propose a distinction between EVALUATING an explanation and RECOVERING an explanation: Readers may rate the relevance, factual faithfulness, clarity, utility, and persuasiveness of an explanation on independent scales. This is important because when the underlying recommendation is weak, an explanation can appear convincing. Thus, there

should be a strict benchmark that assesses the quality of the rank list and the quality of the reason for the rank list.

V. CONCLUSION

In this study, we propose an Explainable Semantic Book Recommendation System based on Sentence-BERT embeddings, cosine similarity and Gemini 2.5 Flash explanations. The method uses 15,147 data from Google Books to map the descriptions into vectors and search the vector space for the 5 most semantically similar books and generate natural language explanations for the recommendations. The similarity between the samples and the evaluated samples is 0.62 on average for the top five most similar books, and it shows that books can be retrieved by semantic retrieval even if they do not have the same keywords. The LLM explanation layer enhances interpretability by converting the selected and recommended books into human-readable text.

The combination of collaborative filtering, reader profiles, ratings and reading history could be used for providing hybrid personalized recommendations in the future. It should be expanded to include Precision@K, Recall@K, NDCG, diversity, serendipity, and latency and user studies. There are also other potential enhancements to scalability and user engagement that can be realized through multilingual embeddings, vector databases, mobile/web deployments, and other alternative explanation models for LLMs.

The study proposed therefore provides a realistic link between semantic retrieval and explainable suggestion. Its strength lies not in its claim of the accuracy of its recommendations but rather in its design of separating the ranking stage from the similarity score and explanation stage conceptually. This enables the system to be expanded and tested in Higher Education digital-library applications and allows for easy personalised and large-scale evaluation.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.
- [3] S. R. Javaji and K. Sarode, "Multi-BERT for Embeddings for Recommendation System," arXiv:2308.13050, 2023.
- [4] T. Nimbeshah, B. K. Bao, and Z. Ali, "beeFormer: Bridging the Gap Between Semantic and Interaction Similarity in Recommender Systems," in Proc. ACM CIKM, 2024.
- [5] R. F. Kadhem, "Enhancing Arabic Content Recommendation Systems Using BERT-Based Semantic Representations and Hybrid Filtering Methods," in Proc. ICAIT, 2026.
- [6] D. Suhartono and A. Subalie, "Book Recommendation Using Double-Stack BERT: Utilizing BERT to Extract Sentence Relation Feature for a Content-Based Filtering System," Communications in Computer and Information Science, Springer, 2023.
- [7] "Book Recommendation System Using TF-IDF and Cosine Similarity," AIP Conference Proceedings, 2024.
- [8] W. Gao, "Research on Optimization of Library Book Recommendation System Based on the Collaborative Fusion of Transformer Architecture and Adaptive Extreme Learning Machine," J. Theor. Appl. Inf. Technol., vol. 17, no. 12, 2025.
- [9] O. Remadnia, F. Maazouzi, and D. Chefrou, "Hybrid Book Recommendation System Using Collaborative Filtering and Embedding Based Deep Learning," Informatica, vol. 49, no. 8, 2025.

- [10] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 3rd ed. Springer, 2022.
- [11] C. D. Manning, P. Raghavan, and B. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [12] Q. Ma, X. Ren, and C. Huang, "XRec: Large Language Models for Explainable Recommendation," *arXiv:2406.02377*, 2024.
- [13] M. N., R., and K. P. M. B., "A Hybrid Explainability Framework for Recommender Systems Using SHAP with Natural Language Interpretations," *Int. J. Innov. Res. Sci. Stud.*, vol. 8, no. 6, pp. 687–703, 2025.
- [14] S. Wang et al., "Knowledge Graph Retrieval-Augmented Generation for LLM-based Recommendation," in *Proc. ACL*, 2025.
- [15] J. Li et al., "Towards Next-Generation LLM-based Recommender Systems: A Survey and Beyond," *arXiv:2410.19744*, 2024.
- [16] "More Than What Was Chosen: LLM-based Explainable Recommendation Beyond Noisy User Preferences," *OpenReview*, 2026.
- [17] "Sentence Representations for Semantic Textual Similarity: A Systematic Review," 2026.
- [18] "Deep Learning for Journal Recommendation System of Research Papers," *Scientometrics*, vol. 128, no. 1, 2023.
- [19] "Diversity and Serendipity Preference-Aware Recommender System (DSPA-RS)," *Journal of Computational and Cognitive Engineering*, 2024.
- [20] "SessionRec: Next Session Prediction Paradigm for Generative Sequential Recommendation," *arXiv:2502.10157*, 2025.
- [21] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.
- [22] W.-C. Kang and J. McAuley, "Self-Attentive Sequential Recommendation," in *Proc. IEEE ICDM*, 2018, pp. 197–206, doi: 10.1109/ICDM.2018.00035.
- [23] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer," in *Proc. ACM CIKM*, 2019.
- [24] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural Graph Collaborative Filtering," in *Proc. 42nd Int. ACM SIGIR Conf.*, 2019, doi: 10.1145/3331184.3331267.
- [25] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proc. 43rd Int. ACM SIGIR Conf.*, 2020, doi: 10.1145/3397271.3401063.
- [26] Y. Zhang and X. Chen, "Explainable Recommendation: A Survey and New Perspectives," *Foundations and Trends in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020, doi: 10.1561/15000000066.
- [27] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System," *arXiv:2303.14524*, 2023.
- [28] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao, "Large Language Models are Zero-Shot Rankers for Recommender Systems," *arXiv:2305.08845*, 2023.
- [29] L. Wu et al., "A Survey on Large Language Models for Recommendation," *World Wide Web*, vol. 27, Art. no. 60, 2024, doi: 10.1007/s11280-024-01291-2.
- [30] Y. Zhang, "Tutorial on Explainable Recommendation and Search," in *Proc. ACM SIGIR/RecSys research community resources*, 2023.