Project Concrete Design

INFO-693-690: Human-AI Interaction

Presented by: Lauren Miller, Ananya Mhetras, Kriti Sarawgi

December 1, 2024

Instructor: Afsaneh Razi

Introduction

The motivation behind the Bookworm AI system is to provide recommendations to users for books to read. Many books exist and are published each year. Platforms like goodreads provide ratings and user-generated reviews, but can be overwhelming to the user when trying to seek a book recommendation. The bookworm AI system is meant to address this problem by providing advanced recommendations to simplify decision making and enhance the user experience of discovering books tailored to individual tastes. Factors driving the needs for this system include information overload (the overwhelming amount of books available), diverse preference, and time constraints to look for new books.

The objective of the Bookworm AI system is to provide personalized recommendations leveraged natural language processing techniques and cosine similarity to analyze book metadata. Collaborative filtering is used to incorporate insights from user interactions to ensure recommendations align with personal and community preferences. Other objectives of the system include encouraging book discovery, improving the reading experience, and ensuring scalability and flexibility as the system grows. This system will also recognize the importance of promoting literacy, promoting learning, supporting authors of different backgrounds, and fostering a community of readers (Park & Choi, 2024). The bookworm AI system is a personalized guide to better reading experiences by combining advanced AI techniques with user-centered design.

User Interface Design

The user interface (UI) of the Bookworm AI system is designed with a strong emphasis on usability, accessibility, and user engagement. Using Figma for prototyping, the website includes a homepage which features a search option along with access to the bookworm recommender system, a reading difficulty counter, a random genre generator and a database augmentation option.The website is designed with complete focus on the reader and their requirements from a book recommender system.

We took user feedback regarding the website, and before finalising the design, recorded what features they wanted from an AI system and also any additions they wanted to make the process feel more interactive. The suggestions were incorporated, so the system felt balanced and didn't lean heavily on just an AI system.

During our group discussion, we spoke about how there were hurdles while compiling and accessing a database for a recommender system. A collective solution was to also include a "add to the database" feature. This feature not only made sure that our design was user-centric but also benefited a new and upcoming website for a book recommender system.Since in the Human-AI class, user consent, privacy and transparency were consistent themes, we also felt including features like these made for a more honest design.

Apart from such features, we also wanted to make the design lively and interactive, while presenting the reader with a challenge and a feeling of having achieved something. For this, we included features such as a "random genre generator" and "reading difficulty counter" and "weekly editorial".For readers who want to make reading a more social experience, we also wove in a book club meeting feature. The result was a comprehensive recommender system which is also interactive and focuses on the user needs and pain points.

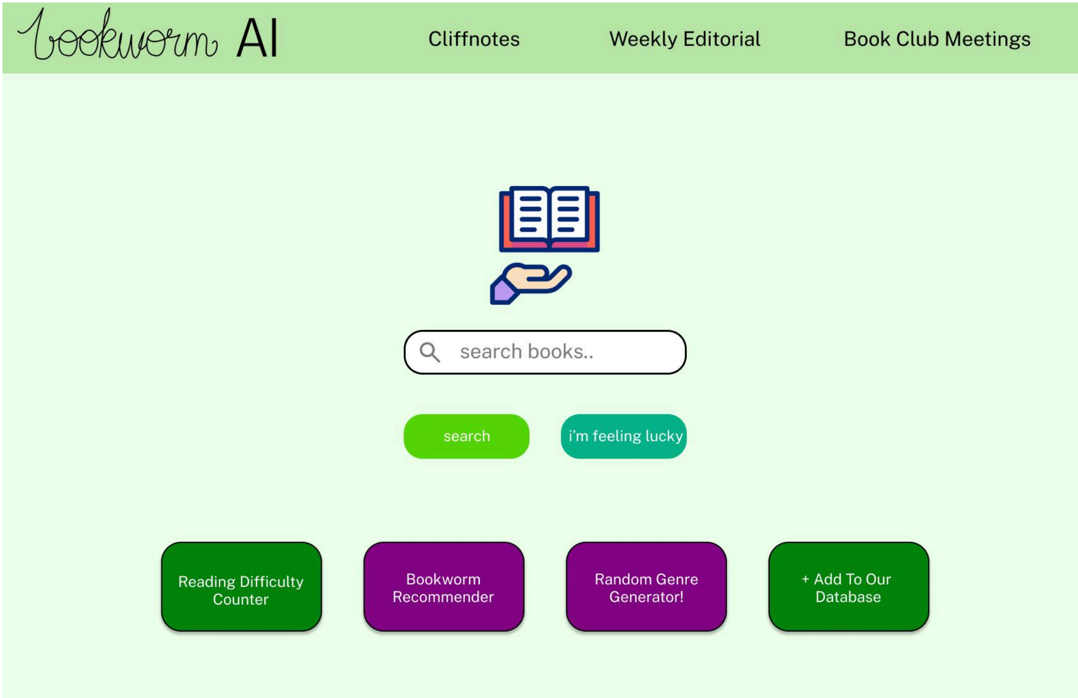

Fig. 1. Homepage of Bookworm AI Interface

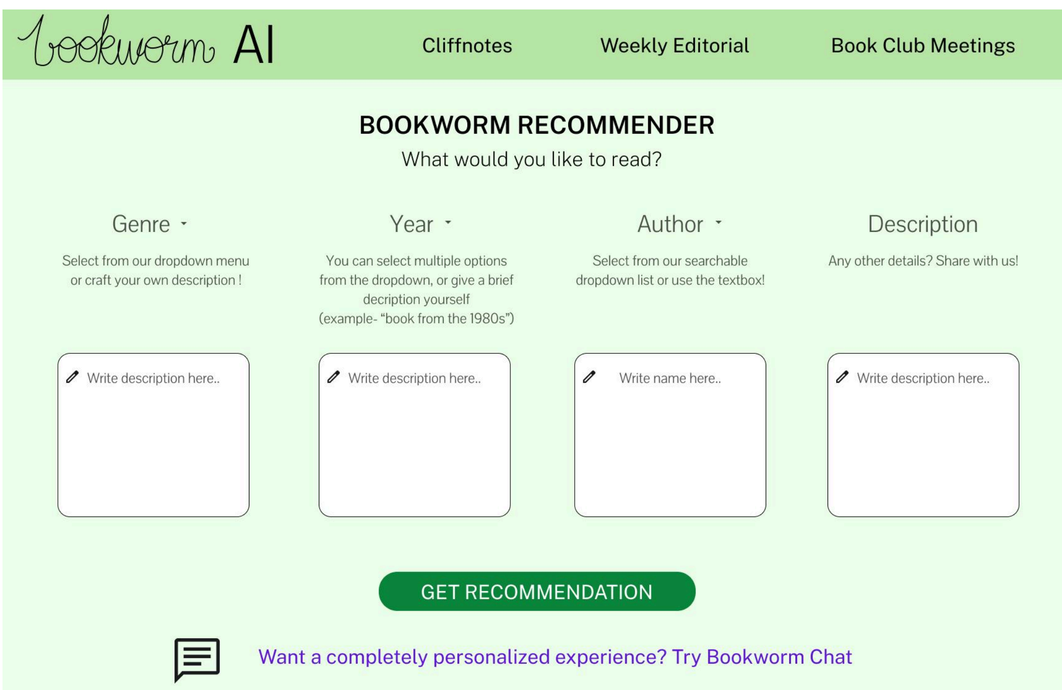

Fig. 2. Recommender system of Bookworm AI Interface

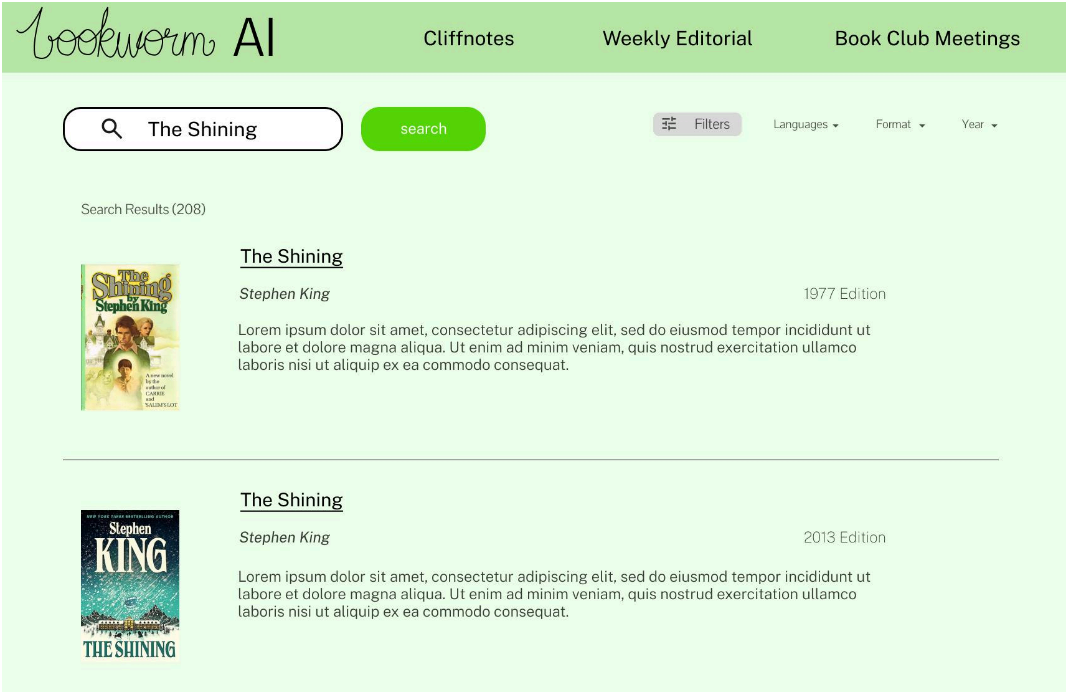

Fig. 3. Search Query of Bookworm AI Interface

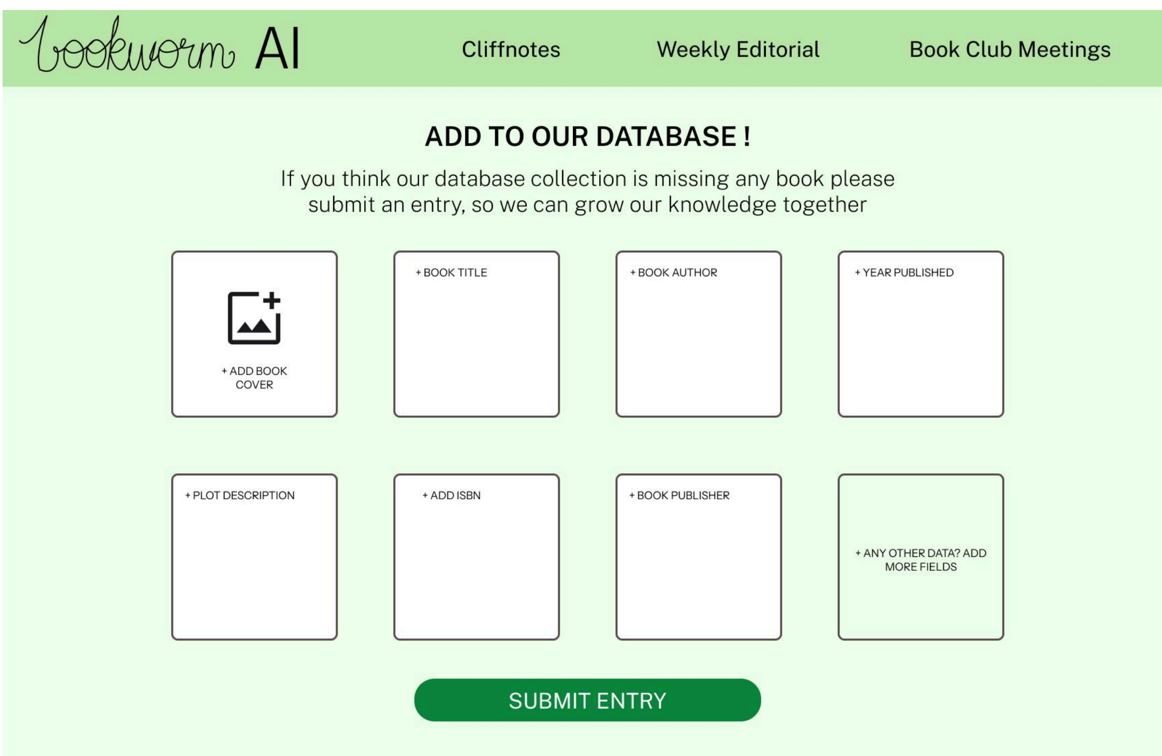

Fig. 4. Database Augmentation for Bookworm AI Interface

AI-Proof of Concept

The recommender system will be built to process large data sets and use both TF-IDF and cosine similarity to provide recommendations. The system analyzes and compares the content of titles using NLP techniques like TF-IDF. Cosine similarity is used to measure how similar the vectors are representing characteristics, guiding the recommendation process. The data is being used from Goodreads data that was scraped from the website by UCSD Book Graph. This data was collected in late 2017 from goodreads.com, where we only scraped users' public shelves. The User IDs and review IDs are anonymized in this data. The goodreads data is the best option for creating this recommender system because it provides book metadata along with user recommendations and ratings. These two things are imperative for making a comprehensive book recommendation system. (Wan, & McAuley, 2018) (Wan, et al., 2019)

The recommender system proof of concept notebook consists of three main parts of code. These parts include searching, generating books, and providing recommendations . The search function finds the top 10 most similar book titles to the user's query. This function preprocesses the query string (converts to lowercase and removes non-alphanumeric characters), converts the query to a TF-IDF vector, and computes cosine similarity between the query vectors. It then selects the 10 most similar books. The highest rating book is sorted first.

Next, the generating book list function iterates over user-provided book titles and calls the search function to find books that match the input title. The first matching book's book_id is taken and appended to a 'liked_books' list. If no book is found, a message is printed saying this.

Lastly, the recommend function is used to recommend books based on user interactions and liked books. The function maps the book_id to goodreads IDs. It then iterates through interactions data from goodreads. It identifies users who rated the liked_books with 4 or 5 stars (ratings >= 4). The users are collected and called 'overlap_users'. The function then iterates over the overlap users data and sees what books are highly rated by them, then retrieves a list of books based on the overlap_users highly rated books. This list is then merged with other book metadata, filtered to not include books the user has already read, and filtered to only show popular books.

This prototype code shows generating personalized book recommendations based on user's liked books and other users' ratings. This prototype system allows users to input any book title and the system will adapt to match similar ones. The system combines TF-IDF with user rating for hybrid recommendations. The output provides clickable links and cover images for a user-friendly experience.

Overall, this prototype system shows a basic proof of concept of this book recommendation system. Further enhancements include incorporating diverse recommendations by mixing highly popular books with niche or lesser-known titles and implementing machine learning models to track user behavior over time or based on real-time interaction during a single session. Further improvements to the system include gamification of the system such as adding rewards or badges for engagement. Adding a chat-based interaction system to this recommendation system would also enhance usability. Also, advanced content integration such as audiobook and ebook integration (offering previews or ability to purchase through other sites) would enhance the user's access to content. This could be used to promote engaging with the recommendations from the system.

Evaluation

The evaluation of the Bookworm AI system will employ both quantitative and qualitative metrics to ensure effectiveness, usability, and user satisfaction. Quantitative metrics include precision and recall to measure recommendation accuracy, click-through rate (CTR), session duration, and repeat usage rates to assess user engagement. Metrics for diversity and novelty are used to ensure recommendations include varied genres, authors, and lesser-known titles. System performance is monitored by tracking response times and scalability under large datasets and concurrent user interactions. Qualitative measures include user satisfaction surveys, task-based usability testing, and feedback on ethical aspects like data privacy and system transparency. Benchmarking against industry standards, such as comparing performance with Goodreads or Amazon, and ensuring compliance with accessibility and data protection standards (e.g., WCAG 2.1 and GDPR), further validate system quality. Continuous improvement is achieved through A/B testing to refine algorithms and interface designs, iterative updates based on user behavior, and incorporating community input for new features or enhancements. These evaluation strategies collectively ensure that Bookworm AI remains a user-centric and impactful tool for personalized book recommendations.

Conclusion

To create an ethically sound recommendation system, we must balance accountability, transparency, fairness, and privacy. This system will use anonymized data to create recommendations, and continually learn from its users to improve recommendations. The conceptual design of this system is based on trial and error in designing a comprehensive system. By taking all these design ideas and ethics into practice, we will be able to work towards building a recommendation system that provides great recommendations for books.

References

Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender systems: An overview. *AI Magazine, 32*(3), 13-18. https://doi.org/10.1609/aimag.v32i3.2361

Park, KS., Choi, H. (2024) How to Harness the Power of GPT for Scientific Research: A Comprehensive Review of Methodologies, Applications, and Ethical Considerations. *Nucl Med Mol Imaging* 58, 323–331 . https://doi.org/10.1007/s13139-024-00876-z

Rodríguez-Hernández, M. del C., & Ilarri, S. (2021). AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions. *Knowledge-Based Systems, 215*, 106740. https://doi.org/10.1016/j.knosys.2021.106740.

Shangying Hua, Shuangci Jin, Shengyi Jiang; The Limitations and Ethical Considerations of ChatGPT. *Data Intelligence* 2024; 6 (1): 201–239. doi: https://doi.org/10.1162/dint_a_00243

Wan, M., & McAuley, J. (2018). Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM.

Wan, M., Misra, R., Nakashole, N., & McAuley, J. (2019). Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*.

Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems, 7*(2), 439-457.
https://doi.org/10.1007/s40747-020-00212-w:contentReference[oaicite:0]{index=0}.