Book Recommender System

Trevor Adriaanse, Tony Dupre,Yashwanth Reddy, Thao Tran

Problem Statement

Our goal is to develop a book recommender system. With an overwhelming array of new books to choose from, readers struggle to discover new titles that align with their interests and preferences. Traditional methods of book recommendations, such as manual browsing through physical libraries or relying on word-of-mouth, are time-consuming and may not effectively cater to individual tastes. There is a clear need for a book recommender system that leverages machine learning techniques to provide tailored recommendations, ensuring that readers can easily discover books that match their specific tastes.

Data Collection and Preprocessing

We've identified a number of promising data sources for our book recommender system. Goodreads, Kaggle, Common Sense Media, and Amazon each offer large collections of books, together with user preference data and other sources of information that may be useful in developing a predictive algorithm. If we move forward on one of these datasets, both Amazon and Goodreads feature APIs, while Kaggle supports direct downloading. In all cases, we have to take care to preprocess data and examine what features (e.g., book title, year of publication, genre, and more) are present for building a recommender system. Ethical considerations include personal identifiable information if we use customer reviews from Amazon. We may also consider filtering out potentially fake reviews using existing fake review APIs. One technical challenge we expect to encounter is creating an efficient system that runs over a large-scale collection of books. Perhaps a vector database as a subcomponent in our overall pipeline will be a way to contend with this issue.

Feature Engineering

Feature engineering plays a crucial role in extracting meaningful information and constructing relevant features from books data. As discussed above, the features we derive from the raw data are dependent on which dataset we decide to use. However, for sake of example, fields such as title, author, and genre reliably appear in each dataset. Three classes of features that will likely confer invaluable signal for our recommender engine include:

- metadata features, such as title, author, and publication date
- user features, such as the ability to prompt the engine ("I'm interested in mystery book from the 1990s") or past reviews, ratings, or readings lists
- global features, such as popularity and trends
- Synthetic data generated using GPT-4 about the topics and summaries related to book
- Mapping book contents as embeddings (If data is available)

Model Selection and Evaluation

Models will need to address two distinct steps

- Processing review data for all book titles in our dataset
- Comparing vectorized features for matches and recommendations

For the purpose of processing review text data, we can start with TF- IDF and other baseline approaches, and then compare those to pre-trained models fine-tuned on our review dataset. For baseline evaluation of the model, content filtering could be used to benchmark and the final model will be built on collaborative filtering with user and books content/reviews as embedded vectors.

Training and Optimization
To develop and fit our neural models, we will use PyTorch. We may perform some basic hyperparameter search, where appropriate.

Deployment and Monitoring
We will begin by organizing our work in a custom Python library. Our goal is to think carefully about data representation and processing through our pipeline, informed by what we learn in class, with an eye towards using vector stores for efficient modeling, a robust monitoring framework, and an end-user interface with user feedback (thumbs up/down) to facilitate a pleasant experience for the user and allow the recommender to adapt to user preferences.

The model endpoint needs to have low latency for inference (<1ms) and appropriate compute/storage to be looked at.

Ethical Considerations
Some of the larger ethical concerns to be aware of as we design our system would be Data Privacy/Protection and Information Bias. As the system would collect and include user data, we would need to implement notification and Terms of Service and data collection consent acceptance for all users that aim to be as transparent as possible about our system data security and anonymity efforts, along with user data access.

For the model itself, we need to take care not to allow existing biases in the data to be amplified and presented through recommendations. A monitoring component would be necessary to address this, that would include defining bias metrics and evaluating training data, and possibly using regularization to penalize identified sentiment or negative intent such as review-bombing or off-topic content. It would then also be equally important to then disclose this criteria as transparently as possible.

Reflection and Future Improvements

Since manual user feedback may not be scalable, another evaluation approach to measure a user's preliminary interest is whether they add the recommended title to their book queue or click to purchase the title. Additional data enrichment sources like book reading level may be helpful especially for recommendations for younger readers. Other future improvements may consider discovering common links between genres/user data that were previously unknown or adjusting weights for new books mentioned at a particular rate (whether good or bad).