Content-Based Book Recommendation System

CS4090 Project Final Report

Submitted by

Chekurthi Deepak	**B200802CS**
Meesala Suresh Gopi	**B200850CS**
Yendluri Jahnav	**B200847CS**

Under the Guidance of
Dr. Vidhya Kamakshi

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May 8, 2024

NATIONAL INSTITUTE OF TECHNOLOGY
CALICUT, KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2024

CERTIFICATE

Certified that this is a bonafide record of the project work titled

CONTENT-BASED BOOK RECOMMENDATION SYSTEM

done by

Chekurthi Deepak
Meesala Suresh Gopi
Yendluri Jahnav

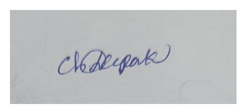
of eighth semester B. Tech in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut

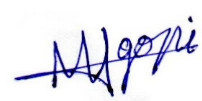
Project Guide
Dr. Vidhya Kamakshi
Assistant Professor

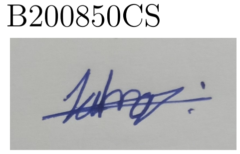
DECLARATION

By this declaration, I assert that the project entitled **Content-Based Book Recommendation System** is of my own work and that it contains no material published or written by any other person or material that has been awarded or submitted in any level of university or institute of higher education, except for those material which has been duly acknowledged and referenced.

Place :NIT Calicut
Date : 8-05-2024

Signature : 
Name: Chekurthi Deepak
Reg. No. : B200802CS

Signature : 
Name: Meesala Suresh Gopi
Reg. No. : B200850CS

Signature : 
Name: Yendluri Jahnav
Reg. No. : B200847CS

Abstract

Books are a source of knowledge that increases a person's wisdom. Since there are a huge number of books available for the user to access, we need a book recommendation model to filter the books based on the user's query. Many book recommendation systems have been introduced, depending on the filtering method. The major filtering methods used are collaborative, content-based, and hybrid filtering. Collaborative filtering mostly depends on the history of the user. This could be inappropriate for a person who cares about privacy. This collaborative filtering also has a major problem with new users' cold starts. To solve this privacy issue and the cold start problem [11], this paper proposes the Content-Based Book Recommendation System model, which doesn't use the user's history. To train the dataset, the pre-trained BERT model was used, which is used to recommend books. Using this approach, books are recommended. It recommends books based on book titles or book summaries given by users.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our guide and mentor Dr. Vidhya Kamakshi, who has guided us throughout the final year project. Without their active guidance, help, cooperation, and encouragement, we would not have made headway in the project. We would like to thank our parents and the faculty members for motivating us and being supportive throughout our work. We also take this opportunity to thank our friends who have cooperated with us throughout the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Faced by the flood of digital content of a huge amount, the avid readers and the occasional enthusiasts of books have the difficulty to find new and appropriate books to read.To boost the book discovery process, the Content-Based Book Recommendation System is introduced. This has the unique features of giving personalized recommendations based on the title, content, genre, and other essential attributes of books.

The system explores a book's content, identifying main themes and deep inner thoughts and feelings as well as the defining features of the story. This is achieved with the help of the most modern deep learning methods; they make personalized suggestions, highlighting the book summaries rather than using the individual user's history. Through examining the book content based on the themes, emotions, and inherent characteristics, the recommended books are derived from the heart of the literature in an attempt to escalate their relevance.

The development of the "Content-Based Book Recommendation System" is driven by the pressing need to navigate the overwhelming abundance of literary content in today's digital landscape. As literature diversifies, readers struggle to find books aligned with their tastes. Recommendation systems have been so extensively used these days that they have become the preferred choice for researchers. Also, recommendation architectures are critical equipment to enhance the user experience and promote sales and services for online websites and mobile applications. For example, Netflix says 80 percent of watched content is based on algorithmic recommendations.

Chapter 2

Literature Survey

2.1 Content-Based filtering

Several notable approaches have emerged in the realm of content-based filtering for book recommendation systems, each offering distinct methodologies and insights into enhancing user experience and engagement with literature.

Yiu-Kai et al[1]. Proposed a method targeted on studying e-book descriptions to assemble consumer profiles, thereby predicting scores for books unknown to the consumer. This approach revolves around the creation of consumer profiles represented as vectors encapsulating their pursuits. These profiles are meticulously crafted using aggregating descriptions of books previously rated by way of the consumer, with key phrases assigned weights based totally on period frequency-inverse record frequency (TF-IDF). Leveraging the vector space model (VSM), predictions of a consumer's score for an unseen ebook are made by calculating the similarity between the consumer profile and the book's description. This method offers a customized and effective method of recommending books tailored to users' choices, thereby enriching the e-book discovery experience.

Raymond et al[2]. introduced the LIBRA (Learning Intelligent Book Recommending Agent) gadget, which takes a completely unique method with the aid of studying individualized profiles from descriptions of examples, taking into consideration a nuanced knowledge of each consumer's options without necessitating contrast with other customers. LIBRA leverages computerized text categorization methods implemented to semi-based textual content extracted from the net, recommending gadgets based totally entirely on their intrinsic attributes rather than counting on the possibilities of other users. Through meticulous assessment throughout diverse genres which includes literature fiction, mystery, science, and technological know-how fiction, LIBRA showcased its efficacy in offering applicable and rather rated ebook tips, bolstered by using metrics inclusive of type accuracy, keep in mind, precision, F-degree, and rank correlation.

Chen et. al[6] proposed a transformer architecture. The BookGPT model proposed in this article will be examined and modeled by using the ChatGPT API, which is available with OpenAI. In contrast to the conventional recommendation system, the core recommendation functionality of BookGPT is based on LLM(Large Language Models) and its recommendation process leverages the understanding of and the natural language representation by the model, thus its output results are highly flexible. Therefore, through crafting appropriate prompt formats, this will help the model to understand tasks better and result in consistent and accurate output.

Liu et al[3]. proposed an innovative content-based recommendation algorithm harnessing convolutional neural networks (CNN) to predict latent factors from textual information extracted from multimedia resources. Employing the split Bregman iteration method[12] for model optimization, this approach transforms input text information into features of the learning resource, subsequently combining it with user preferences to predict ratings. By leveraging real-valued latent factor vectors and minimizing the mean squared error (MSE) of predictions. Figure 1 shows how these predictions are made. This method, particularly the CBCNN variant, demonstrated superior performance compared to alternative models, as evidenced by lower RMSE values.

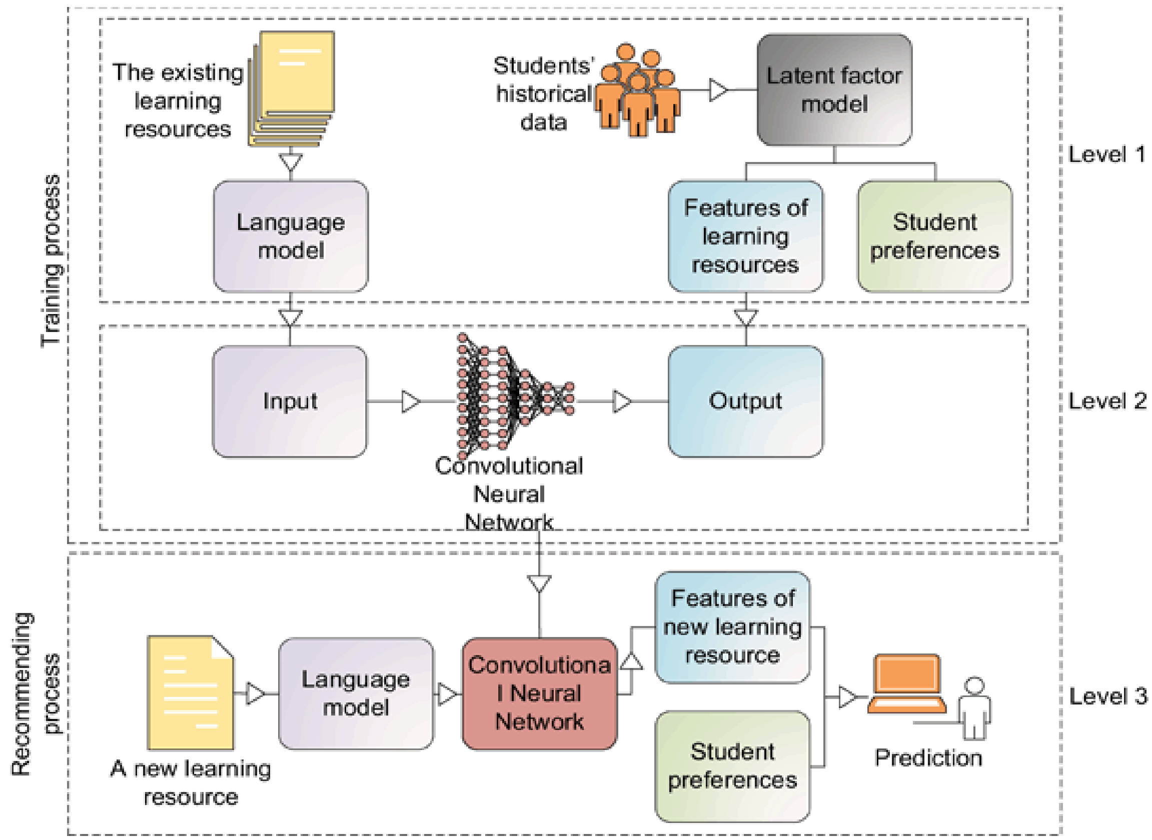


Figure 2.1: Process

2.2 Hybrid based filtering

Sarma et al[4] advanced a hybrid e-book recommendation machine integrating content-based and collaborative filtering. They cleaned and standardized datasets from Goodreads on Kaggle, then used ok-manner to cluster books based totally on user preferences and scores. They hired cosine similarity for measuring distances between books and evaluating similarities within clusters. The device's evaluation worried calculating sensitivity, specificity, and F1 ratings throughout ten datasets, displaying promising overall performance in figuring out effective and negative instances, in addition to providing relevant recommendations. ROC curve analysis supported the system's effectiveness, positioning maximum datasets close to the best diagonal line. Their system excels in refining hints by putting off less engaging books. They spotlight the hybrid nature of their layout, emphasizing the mixture of content-based and collaborative filtering, with clustering improving accuracy and relevance.

Table 2.1: Comparison of Recommendation Models

Description	Experiment Limitations	Ref
• Ranks the books that are recommended • Opinion Mining technique is used to improve the accuracy of the recommendation system	• Limited to computer science-related books • Only recommends 10 books for a particular query	[5]
• combined features from two widely used filtering techniques - content-based filter and collaborative filtering	• using sensitive attributes like the registered user, password, and email	[7]
• The matrix sparsity problem in filtering was solved by the author • Recommends books to newly admitted students with a percentage of high accuracy	• Clustering of books and length of books were not considered in the recommendation	[8]
• To increase accuracy using a user-based similarity matrix with a collaborative filtering algorithm	• Clustering can improve the accuracy and performance of the recommendation system	[9]
• Used Support vector machines to find the relationship between the title of the books or the bibliographic of the authors	• Overfitting problem due to less number of books in the dataset	[10]

Chapter 3

Problem Definition

Privacy concerns and cold start issues in existing book recommendation systems prompt the need for a Content-Based Recommendation System that doesn't rely on user history.

Chapter 4

Methodology

Objective: Utilize a pre-trained BERT model to recommend books based solely on attributes like titles and summaries.

4.1 Proposed Design

The sequential stages outlined in the flowchart define the Transformers-Enabled Book Recommendation Pipeline.

Data gathering

This dataset, sourced from bookdepository.com via the sp1thas/book-depository-dataset repository, comprises 100k books. It includes metadata such as title, description, authors, category,publisher, publish Date and price providing comprehensive details about each book without the real content.

Data preprocessing

Remove duplicates, and unique characters, and prevent phrases from ebook summaries. This step guarantees a smooth and standardized dataset for powerful model training. Tokenize the summaries. Employ

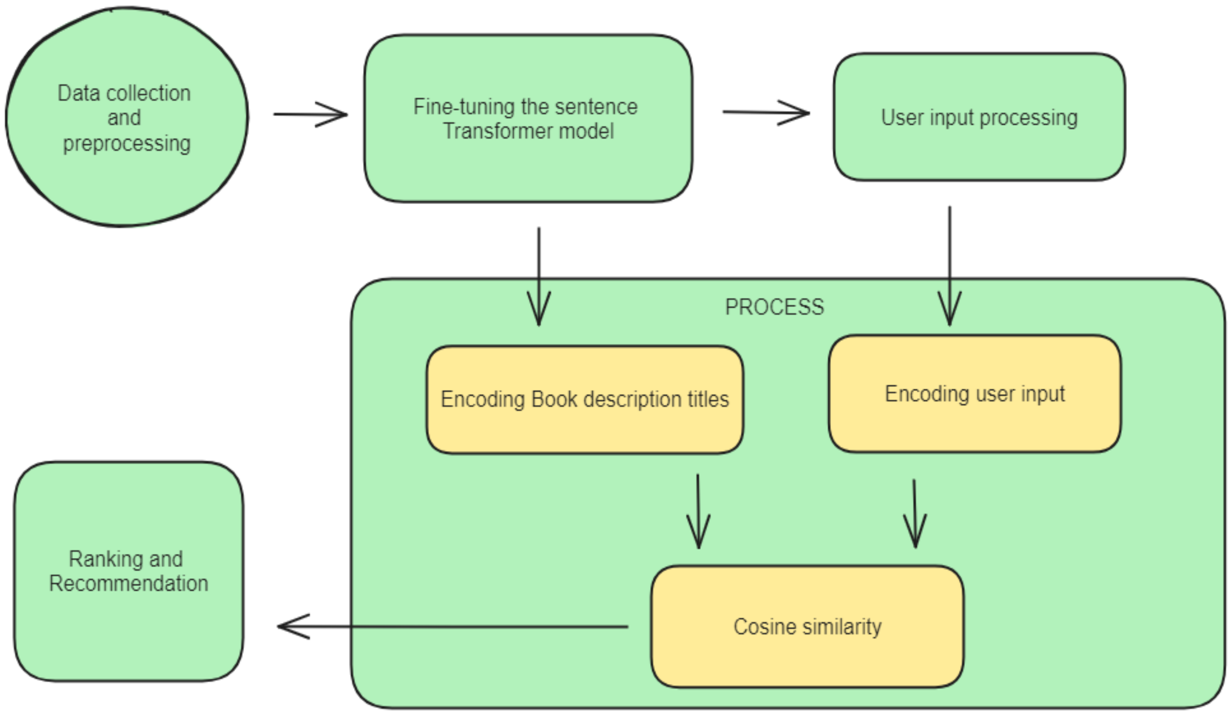


Figure 4.1: Transformers-Enabled Book Recommendation Pipeline

tokenization to convert textual content into numerical sequences, a prerequisite for Transformer models.

Build the model

1. Initialization of the Sentence Transformer Model

The script here in utilizes the Sentence Transformers [14] library to initialize (with the Sentence Transformer model) a model named MiniLM-L6-v2. Sentence transformers employ transformer architecture (e.g., BERT) [13], but this application is for sentence and text classification tasks. We initialize a class instance and assign it to the appropriate device. The availability of a GPU determines this; if not, CPU processing takes over.

2. Custom Triplet Loss Model

In order to perform this task, the TripletLossModel class is created as a result of PyTorch's nn.Module class extension.The transformer-named entity recognition model (Transformer-NER) uses a sentence transformer (SentenceTransformer) as an embedding model. The forward method takes only three inputs, which makes it possible to make you feel happy, sad, or any other separate thing. The sentence transformer converts the texts under study to obtain their embeddings as well.

3. Triplet Margin Loss

In this particular instance, the model uses nn.TripletMarginLoss as the loss function. The so-named triplet margin loss is proposed in order to adjust the model in such a way that negative and positive embeddings get as close as possible while anchor and negative embeddings become as far as possible.

4. Data Preparation

The script starts loading triplets (anchor, positive, and negative) from `triplet_data.csv` file. The data is formatted by a TripletDataset class, which represents a child of the Dataset class of PyTorch. Here, this Dataset is passed to a DataLoader in order to perform batching and shuffling of the data during training.

5. Training the model

please define a function- `train_triplet_model` to train the model. Each turn the module tries over the amount of triplets from the data loader ship', these are assigned to batches. Every run of the function is done by embedding anchor, positive, and negative instances using the Sentence Transformer model. Finally, the triple margin loss is obtained using the earlier embeddings. Hence the model applied the backpropagation technique to adapt its parameters with an optimizer (optim.Adam). The data frame containing the outputs of every step is recorded in order to periodically judge our progress.

The model is notable in that it extracts features using triplet data and is later trained with triplet margin loss. Through the tuning process, the pattern of weights in the model gets readjusted to ensure it can discriminate between the anchor and positive and negative examples, among others. This fine-tuning gives the model improved recommendation results because of its better performance. The model used is a sentence transformer as the embedding model, and triplet margin loss is necessitated for fine-tuning the model so as to improve its capability to predict text similarity with accurate book recommendations.

Training Algorithm Overview

1. Load the Sentence Transformer model from all-MiniLM-L6-v2 and triplet data (`triplet_data.csv`) which contains matching anchors, positive and negative examples.

2. Have a DataLoader that can manage the triplet data feed batch by batch and get it shuffled during training for better training speed.

3. Note that the nn.TripletMarginLoss function is used to specify the triplet loss and optim.Adam optimizer is employed for training the model.

4. The iteration loop is designed for each epoch to compute data batches with the triplet data, compute the embeddings, calculate loss, and update model parameters with the backpropagation technique.

5. Finally, save the trained model and utilize it when necessary, making sure that it is retrieved safely and conveniently.

Testing Algorithm Overview

1. Put the test data in `(increased_test_data_n=4.csv)` that is queries, and books that recommend into, for evaluation.

2. First of all train or compute embeddings of all the books in the data set using the Sentence Transformer model. Later, the user query embeddings can be compared with them.

3. Calculate cosine similarity by carrying out the pair-wise comparison of user query embeddings with book embeddings, to form the top N relevant book recommendations.

4. Do a weighted average of the overall accuracy throughout all user queries by comparing the recommended books with the actual recommended books, and thus the level of model performance can be checked.

Sentence Transformer Model The sentence transformer we employed is designed to carry out activities like sentence similarity comparison and semantic comprehension.The model is built on the same BERT architecture that has already proven to represent context more accurately and to be computationally efficient for handling text data. Using previously trained language models in the design allows it to fine-tune to specific tasks, even if the datasets for these tasks are limited. Sentence Transformer can compute semantic similarity between user queries and book presentations.Thus, its ranking system can be built on relevance. Meanwhile, the model's advanced linguistic comprehension can not only provide the appropriate context but also make the book recommendation more precise.

Fine-Tuning of Sentence Transformer Model

In this part of the process, the pre-trained book recommendation model is adapted to the custom task of sentence transformation. A transfer learning approach is adopted, which utilizes a model pre-skilled enough in text data to be able to adapt to the use of books and users's queries. The model will go through a fine-tuning process, which means that it will be tuned on a dataset of book descriptions and user queries, which will, in turn, increase its ability to capture the character of book content and the user's preferences. Through this vital stage, the model is fine-tuned, making it capable of serving the exact requirements of the book recommendation system better.

Encoding, Comparison, and Ranking

On reaching the fine-tuning stage, the Sentence Transformer model takes the user queries and the book descriptions and numericizes them into vectors. The underlying encoding process, in turn, enables the utilization of cosine similarity, commonly used for comparison and similarity measurements. The model gives a list of books sorted based on their relevance to the user's query, and the ones with the top ranks are the ones that contain the most semantically similar content to the query. The system will then present to the user a list of high-ranking options as a result. That's how the recommendations are made sure to be in line with the user's book tastes, thus bringing up rather precise and personalized suggestions.

4.2 Work Done

4.2.1 TF-IDF model

The TF-IDF model was used in experimentation to increase the accuracy of the system through the importance of words in book descriptions.The TF-IDF vectorization procedure transformed textual data into numerical representations, enabling the computation of similarity scores. Nevertheless, this way is not absolutely free of problems. The meaninglessness of each word without recognizing its contextual nuances may result in the system being unable to grasp contextual nuances. Stop-word sensitivity and the inability to handle synonyms would be additional challenges for the model.

4.2.2 LSTM model

The objective of a novel LSTM-based autoencoder method is to refine and widen the implementation of neural network techniques using the Long Short-Term Memory layer in personalization systems. TF-IDF, being one of the oldest, works on terms separately without taking into account the sequence. Nevertheless, while RNNs prove to us as a good tool to work with the texts and sequences, LSTMs display that they are better at capturing and representing the sequential dependencies and long-term effects. Here, the significance of that characteristic is that the second-word reading of the book previews the contents of the next chapters.

The attention-embedding process creates content-based recommendations by emphasizing the understanding of heavy semantic relations and contextual sociolinguistic nuances in the data text.

The recommendation project implemented LSTM-based autoencoders sequentially to enhance the system's ability to identify dependencies and semantics in book descriptions. However, deep learning has proven to be specifically beneficial for speech and image recognition, which are complex tasks; however, it is also not without its drawbacks, such as the increased computational requirements required for training neural networks with intricate neural network architectures like LSTMs. Increased computational intensity may also pose certain problems, especially in areas with resource gaps. In addition to this, the optimal performance of the models is directly dependent on hyperparameter tuning, which comes with increased complexity while developing the models. The effectiveness of the approach closely depends on the quality and representativeness of the dataset, thus highlighting the fact that careful selection is necessary not only in architecture but in hyperparameters as well.

4.2.3 Sentence Transformer model

The sentence transformers, instead of simpler LSTMs or vanilla BERT models, provide another layer of complexity and intricacy to the data consumption analysis process, overhauling the methods of textual data processing. Sentence transformers stand on top of transformer-based pre-trained models like BERT but are purposefully trained to generate high-quality sentence-level embeddings that capture semantic similarity and contextual information. This model is built with a sentence-level task optimization approach and is superior compared to the traditional transformers, which were designed for token-level tasks.

The Sentence Transformer model uses attention mechanisms, which allow it to handle a sequence as a whole while evaluating the context and meaning of each word in the sentence with respect to the entire sentence. The model adopts this holistic strategy by enabling it to capture long-distance dependencies as well as intricate meanings, which aid in creating contextualized embeddings that are very detailed. These embeddings embrace the deep context and its complex meaning, making them an ideal lot for content-based recommendations.

Incorporating sentence transformers into our recommendation model delivers tangible benefits, as the model is capable of leveraging a wealth of knowledge on the basis of various textual data sources that it has already been trained with. The model's extensive dataset fine-tuning enables it to recognize language patterns' complexes and semantic relationships, which drives more sophisticated embedding capability generation. Such semantic redundancy gives meaning and relevance to recommendations, which translates to more specific and useful suggestions for users.

In the world of recommendation systems, Sentence Transformers is a relatively new solution with versatile attributes and is able to overcome some of the issues of LSTM-based and simple BERT approaches. By emulating natural language, Sentence Transformers can mine the textual relationships down to the details, which consequently increases the prowess of the recommendation system to give the users tailored, contextualized suggestions about books.

Ranking these suggestions is done based on how closely they relate to the user's input. The model provides the user with the most accurate top 10 options in descending order of similarity, so the user is guaranteed to see the most relevant suggestions at the beginning. Although the training of machine learning can be quite intensive in terms of computation, the benefits in accuracy and better contextual depth are worthwhile, thus improving the user experience with personalized and tailored recommendations.

Chapter 5

Experimental Results

5.1 Result

The experimental results of our study show the performance of the sentence-transformer model presented in this section. The model is evaluated on a benchmark dataset, and various performance metrics are reported to assess their effectiveness.

5.2 Performace Metrics

To evaluate the performance of the models, we used the following metrics:
1.**Accuracy :** In our recommendation system, the accuracy metric quantifies the proportion of true recommendations that are successfully matched within the top-N recommendations provided by our model. This metric is computed as follows:

i. Intersection: The set of correct recommendations is determined by intersecting the set of true recommendations with the set of top-N recommended books from your model. Mathematically, this is expressed as:

$$\text{correct\_recommendations} = \text{true\_books} \cap \text{predicted\_books}$$

ii. Accuracy: Accuracy is the ratio of the number of correct recommendations to the total number of true recommendations. This can be calculated using the formula:

$$\text{accuracy} = \frac{\text{number of correct recommendations}}{\text{number of true recommendations}} = \frac{\text{len(correct\_recommendations)}}{\text{len(true\_books)}}$$

iii. Average Accuracy: Average accuracy is the mean accuracy across all test cases, providing an overall assessment of the model's performance. The formula for calculating average accuracy is:

$$\text{average\_accuracy} = \frac{\sum_{i=1}^{n} \text{accuracy}_i}{n}$$

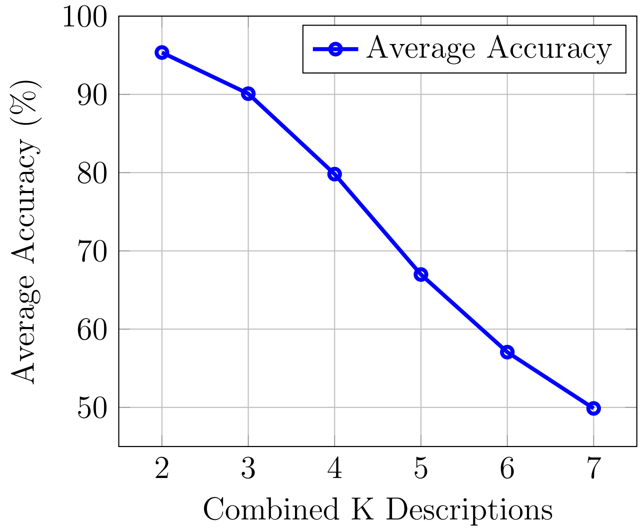
where n represents the number of test cases and accuracy_i is the accuracy for the i-th test case.

By using these formulas, the accuracy metric assesses how well the model's top-N recommendations align with the true recommendations from the test data, providing insight into the model's performance.

5.3 Test Results

Combined K Descriptions	Average Accuracy (%)
2	95.35
3	90.09
4	79.81
5	66.98
6	57.06
7	49.87

Table 5.1: K-values and their corresponding average accuracy (%)

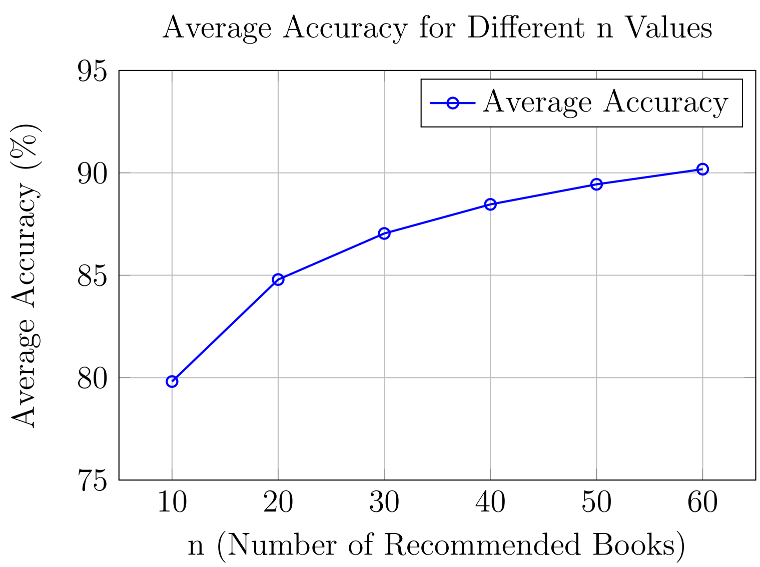


When the quantity of compound descriptions decreases, the data that is provided demonstrates the improvement that the book recommendation model has brought about. This improvement is associated with the query that is associated with (k).The model's average accuracy significantly improves when the k parameter is reduced from 7 to 2, with the highest rate of accuracy, which is 95.35%, being achieved when k is set to 2. Consequently, this may require the utilization of search user queries that are more specific and succinct by the consumers, which will ultimately result in more accurate recommendations.

The precision of the model at higher k values is significantly dependent on its capacity to interpret complex relationships between a large number of features at the same time. This, in turn, may lead to some misjudgments and noise information, which presents a challenge for the model to correctly understand the user's interests and provide content that corresponds to those interests. On the other hand, the model is superior when it comes to addressing the more complicated concerns that involve the two emotions, while this fact provides us with knowledge of the queries that are straightforward. results.

Number of Recommended Books(n)	Average Accuracy (%)
10	79.81
20	84.79
30	87.04
40	88.46
50	89.44
60	90.18

Table 5.2: n-values and their corresponding average accuracy (%)



When the mean accuracy of the book suggestion model increases with the rise in the number of top recommendations from 10 to 60, the data curves upward. This occurs between the ranges of 10 and 60. The presence of such a trend suggests that the model is becoming more accurate in its ability to cater to the tastes of users as the number of recommendations contained within the result set continues to grow.

When the value of n is increased, the number of books that are suggested will also increase. This will increase the likelihood that the recommendations that are made will have a high level of trustworthiness because they will be based on a greater number of books. To put it another way, the proximity of the model to the user's preferences will increase in proportion to the amount of space available for the best recommendations that are tailored to the user's desires.

Based on these findings, it appears that the performance of the model might be improved by including the recommendations of a highly integrated system as an essential component. The availability of a wide selection of books is therefore an essential component that will most likely result in users discovering several recommendations that are tailored to their particular areas of interest.

Chapter 6

Conclusion

The relevance of the content-based recommendation system algorithm in book suggestions is illustrated by the fact that it is capable of providing personalized recommendations by considering essential attributes of a book, such as description and title. These infrastructures provide the basis for personalized tips through the leverage of deep learning models like Sentence Transformers to capture complex patterns out of book data, and at the end, the quality of the recommendation improves. This is achieved by making user profiles with personalization a priority. The systems then provide tailored suggestions that correspond to the interests of a particular user.

Not only do content-based systems make the book selection process more interactive, but they also help the users explore a larger range of books beyond their reading history, thus adding more beauty to the book selection process. Since this type of system is advanced, it will be central to making reading more accessible and enjoyable for people with different literary tastes. AI and machine learning interlaced in book recommendations create a real perspective for people of different cultures and interests.

The project covers a content-based book recommendation system that suggests books according to the users' profiles. The model makes use of content-based filtering and sentence transformers, which are deep learning models, to analyze book descriptions to suggest more relevant books. The scheme suggests a complex blend of contextual awareness with reasonable computational complexity as principles of implementation. Instead of utilizing traditional systems such as TF-IDF and LSTM, the project is aimed at upgrading the book discovery system by socializing individual user needs and providing tailored recommendations.

References

[1] Y.-K. Ng, "Cbrec: a book recommendation system for children using the matrix factorization and content-based filtering approaches," *International Journal of Business Intelligence and Data Mining*, vol. 16, no. 2, pp. 129–149, 2020.

[2] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, 2000.

[3] X. Shen, H. Liu, *et al.*, "A content-based recommendation algorithm for learning resources," *Multimedia Systems*, March 2018.

[4] D. Sarma, T. Mittra, and M. S. Hossain, "Personalized book recommendation system using machine learning algorithm," *International Journal of Advanced Computer Science and Applications(IJACSA)*, 2021.

[5] S. S. Sohail, J. Siddiqui, and R. Ali, "Book recommendation system using opinion mining technique," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2013.

[6] Z. Li, Y. Chen, X. Zhang, *et al.*, "Bookgpt: A general framework for book recommendation empowered by large language model," *Electronics*, 2023.

[7] P. Mathew, B. Kuriakose, and V. Hegde, "Book recommendation system through content-based and collaborative filtering method,"

in *International Conference on Data Mining and Advanced Computing(SAPIENCE)*, 2016.

[8] J. Qi, S. Liu, Y. Song, and X. Liu, "Research on personalized book recommendation model for new readers," in *3rd International Conference on Information Systems Engineering (ICISE)*, 2018.

[9] M. Kommineni, P. Alekhya, T. M. Vyshnavi, V. Aparna, K. Swetha, and V. Mounika, "Machine learning-based efficient recommendation system for book selection using user-based collaborative filtering algorithm," in *Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020.

[10] K. Puritat and K. Intawong, "Development of an open source automated library system with book recommendation system for small libraries," in *Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2020.

[11] Y. Shao and Y.-h. Xie, "Research on cold-start problem of collaborative filtering algorithm," in *In Proceedings of the 3rd International Conference on Big Data Research (ICBDR '19)*, 2019.

[12] Y. Xu, T.-Z. Huang, J. Liu, and X.-G. Lv, "Split bregman iteration algorithm for image deblurring using fourth-order total bounded variation regularization model," *Journal of Applied Mathematics*, 2013.

[13] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.