Report On

Cross-Domain Recommendation System

Submitted in partial fulfillment of the requirements of the Mini project in Semester VIII of Fourth Year Artificial Intelligence & Data Science Engineering

by

Prathmesh Bhagat (60)

Mokshad Sankhe (67)

Sudeep Shetty (70)

Bhavya Wade (72)

Under the guidance of

Prof. Kshitija Gharat

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(A.Y. 2024-25)

CERTIFICATE

This is to certify that the Mini Project entitled **"Cross-Domain Recommendation System"** is a bonafide work of **Prathmesh Bhagat (60), Mokshad Ketan Sankhe (67), Sudeep Shetty (70) and Bhavya Wade (72)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in Semester VIII of Fourth Year **"Artificial Intelligence and Data Science".**

Prof. Kshitija Gharat
Guide

Prof. Sejal D'mello	Dr. Tatwadarshi P N	Dr. Rakesh Himte
Deputy HOD AI & DS	HOD AI & DS	Principal

Mini Project Approval

This Mini Project entitled **"Cross-Domain Recommendation System"** by **Prathmesh Bhagat (60), Mokshad Ketan Sankhe (67), Sudeep Shetty (70) and Bhavya Wade (72)** is approved for the degree of **Bachelor of Engineering** in in Semester VIII of Fourth Year **Artificial Intelligence and Data Science.**

Examiners

1…….....
(Internal Examiner Name & Sign)

2……
(External Examiner Name & Sign)

Date:

Place:

Abstract

Recommender systems have become an essential tool in various industries, enabling personalized content delivery to users. However, traditional recommendation methods often struggle with sparsity issues and cross-domain adaptability. This project presents a Cross-Domain Recommender System that integrates Word2Vec-based embeddings (along with doc2vec and rdf2vec) to generate personalized recommendations across distinct domains, such as movies and books.

The system leverages deep learning-based word embeddings to analyse semantic relationships within textual data, enabling cross-domain mappings between different content types. It features a web-based user interface for input queries and a Python-based backend server that processes similarity computations in real time. The combination of multiple embedding techniques enhances recommendation accuracy, overcoming the limitations of traditional filtering approaches.

The proposed system aims to provide scalable, robust, and adaptable recommendations while ensuring high computational efficiency. The results demonstrate the effectiveness of multi-embedding techniques in improving cross-domain recommendation accuracy. The project represents a step toward more intelligent, semantic-aware recommendation engines, with applications extending beyond entertainment to domains such as education and e-commerce.

Acknowledgement:

We would like to thank all people whose support and cooperation has been an invaluable asset during this Project. We would also like to thank our Guide **Prof. Kshitija Gharat**, for guiding us throughout this project and giving it the present shape. It would have been impossible to complete the project without their support, valuable suggestions, criticism, encouragement, and guidance.

We convey our gratitude to **Dr. Tatwadarshi Nagarhalli**, Head of Department, for their motivation and providing various facilities, which helped us greatly in the whole process of this project. We are also grateful to all other teaching and non-teaching staff members of the Artificial Intelligence and Data Science Department for directly or indirectly helping us with the completion of projects and the resources provided.

Prathmesh Bhagat (60)

Mokshad Sankhe (67)

Sudeep Shetty (70)

Bhavya Wade (72)

Date:

Place:

Contents

		Abstract	**i**
		Acknowledgment	**ii**
		List of Abbreviations	**iii**
1		**Introduction**	**1**
	1.1	Introduction	
	1.2	Problem Statement & Objectives	
	1.3	Scope	
	1.4	Technologies	
2		**Literature Survey**	**4**
	2.1	Survey of Existing System	
	2.2	Limitation Existing system & Research Gap	
	2.4	Mini Project Contribution	
3		**Proposed System**	**6**
	3.1	Datasets	
	3.2	Details of Hardware & Software	
4		**Implementation**	**7**
	4.1	Sequence Diagram	
	4.2	Results	
	4.3	Analysis of Mini Project	
	4.4	Conclusion	
	4.5	Future Scope	
5		**References**	**11**

List of Abbreviations

1. NLP - Natural Language Processing
2. Word2Vec - Word to Vector
3. Doc2Vec - Document to Vector
4. RDF2Vec - Resource Description Framework to Vector
5. CSV - Comma-Separated Values
6. URI - Uniform Resource Identifier
7. HTTP - Hypertext Transfer Protocol
8. UI - User Interface
9. CF - Collaborative Filtering
10. CBF - Content-Based Filtering
11. AJAX - Asynchronous JavaScript and XML
12. JSON - JavaScript Object Notation
13. RAM - Random Access Memory
14. CPU - Central Processing Unit
15. TF-IDF - Term Frequency-Inverse Document Frequency
16. API - Application Programming Interface
17. arXiv - Archive for Research Papers in Computer Science, AI, and Mathematics
18. VS Code - Visual Studio Code

1. INTRODUCTION

1.1 INTRODUCTION

Recommender systems play a crucial role in providing users with personalized content across various platforms. Traditional recommendation systems often face limitations when dealing with cross-domain recommendations, as they are typically designed to operate within a single domain. The growing diversity of user preferences and the vast availability of content across different domains necessitate the development of more advanced and adaptable recommendation models.

This project aims to design and implement a cross-domain recommender system using Word2Vec embeddings. By leveraging natural language processing (NLP) techniques, the system will analyze and extract semantic relationships from textual data related to movies and books. Word2Vec, along with complementary embedding methods like doc2vec and rdf2vec, will be used to generate vector representations of entities, enabling the system to identify similarities and recommend relevant content across domains.

The primary objective of this project is to bridge the gap between domains, allowing users to receive recommendations for movies based on their interest in books, and vice versa. The system will utilize a Python-based backend to process user queries and generate recommendations using precomputed embeddings. The recommendations will then be displayed through a web-based interface, providing users with a seamless and interactive experience.

By combining advanced NLP techniques with an efficient recommendation engine, the project aims to deliver personalized, accurate, and scalable recommendations. This cross-domain approach has the potential to enhance user satisfaction, increase content consumption, and provide valuable insights into user preferences across different content types.

1.2 PROBLEM STATEMENT & OBJECTIVE

Problem Statement:

Recommending content across distinct domains—such as movies and books—presents a considerable challenge. Conventional methods, often tailored to a single domain, may fail to capture the subtle semantic correlations between different types of media. The main challenge is to develop a system that can reliably analyse and map the semantic relationships inherent in textual data across domains, ensuring that recommendations remain both relevant and accurate.

Objectives:

a. To develop a robust cross-domain recommender system that integrates Word2Vec-based embeddings (including doc2vec and rdf2vec) for semantic analysis.
b. To create an interactive and user-friendly platform that allows seamless input and real-time retrieval of recommendations.
c. To improve the accuracy of cross-domain recommendations by effectively capturing and leveraging semantic relationships between movie and book data.
d. To enhance scalability and adaptability, ensuring that the system can be extended to incorporate additional domains in future.
e. To provide a comprehensive solution that streamlines recommendation processes for both end users and developers, ultimately contributing to smarter content delivery.

1.3 SCOPE

1. **Target Users:**

- Individuals interested in exploring personalized recommendations across entertainment domains, including movies and books.
- Developers and researchers focusing on enhancing recommender systems with deep learning and natural language processing techniques.
- Content providers and digital marketers seeking to tailor user experiences based on semantic similarities between disparate media.

2. **Features:**

- Real-time cross-domain recommendations using precomputed semantic embeddings.
- A web-based interface for query input and display of recommendation results.
- Support for multiple embedding methods to enhance the robustness of recommendations.
- Visualization of similarity metrics and recommendation lists for improved interpretability.

3. **Platforms:**

- Primary platform: Web application accessible on major browsers with a backend server implemented in Python.

1.4 TECHNOLOGIES:

- Python: Python serves as the primary backend language for this project, handling server operations, data processing, and recommendation algorithms. Its extensive libraries make it ideal for implementing machine learning-based recommendation systems.
- NumPy: NumPy is used for efficient numerical computations, particularly for handling the embedding matrices and performing similarity calculations between vectors. It enables fast matrix operations essential for recommendation systems.
- HTTP.server (Python Standard Library): The built-in HTTP server module is used to create the web server that handles incoming requests and serves responses, forming the backbone of the backend API.
- JSON (Python Standard Library): Used for data serialization when sending recommendation results from the backend to the frontend, ensuring easy parsing by JavaScript.
- CSV (Python Standard Library): Used for reading and parsing the embedding data files that contain the URI mappings and vector representations.
- urllib.parse (Python Standard Library): Used for URL parsing and query parameter extraction from incoming HTTP requests.
- HTML5: Provides the structural foundation for the web interface, including the form elements and layout containers for the recommendation panels.
- CSS3/Bootstrap: Used for styling the user interface, with Bootstrap providing responsive design components and pre-styled elements for a professional look.
- JavaScript/jQuery: Handles client-side interactions including:
    - Form submission via AJAX
    - Autocomplete functionality for movie title inputs
    - Dynamic updating of recommendation results
    - User interface interactions
- jQuery UI: Specifically used for the autocomplete widget that suggests movie titles as users type in the input fields.
- Embedding Techniques (Doc2Vec/RDF2Vec): While not direct technologies, these machine learning techniques form the core of the recommendation system:
    - Doc2Vec for document embedding-based recommendations
    - RDF2Vec for knowledge graph-based recommendations
- Linear Algebra: Fundamental mathematical operations (dot products, similarity calculations) that power the recommendation algorithms, implemented using NumPy.

2. LITERATURE SURVEY

Recent advancements in recommendation technologies have focused on leveraging word embeddings to capture semantic relationships in textual data. Models such as Word2Vec, Doc2Vec, and RDF2Vec have been shown to significantly improve recommendation accuracy by embedding contextual information into vector space representations. Combining multiple embedding strategies can overcome the limitations of traditional filtering methods, particularly in cross-domain scenarios where content characteristics vary widely.
Recommender systems are a critical component of the modern digital ecosystem, providing users with personalized content across various platforms. Traditional methods such as collaborative filtering (CF) and content-based filtering (CBF) have proven effective in single-domain applications but often struggle in cross-domain scenarios due to data sparsity and semantic gaps.

2.1 SURVEY OF EXISTING SYSTEM

- **Saleh Hassan (2021).** "Cross-Domain Recommender System". This project introduces a web-based interface for both single-domain and cross-domain recommendations, utilizing techniques like Word2Vec, Doc2Vec, and RDF2Vec. The system computes embeddings for entities within RDF graphs and employs similarity computations to generate recommendations across domains.

- **Makbule Gulcin Ozsoy (2016).** "From Word Embeddings to Item Recommendation". This study explores the application of Word2Vec techniques to recommendation systems, focusing on non-textual features such as user check-ins. The research demonstrates that continuous vector space representations modeled by Word2Vec can effectively predict user preferences, suggesting its potential in cross-domain recommendations.

- **Dhilip Subramanian (2020).** "Content-Based Recommendation System using Word Embeddings". This article discusses the development of a content-based recommendation engine using average Word2Vec and TF-IDF Word2Vec methods. It highlights how word embeddings can capture semantic meanings more effectively than traditional TF-IDF approaches, thereby enhancing recommendation quality.

2.2 LIMITATION OF EXISTING SYSTEM:

Sr No	Paper Title	Published Year	Limitations	Research Gap
1	Cross-Domain Recommender System	2021	Relies on precomputed embeddings; may not adapt well to dynamic data changes.	Develop methodologies to bridge semantic gaps across domains.
2	From Word Embeddings to Item Recommendation	2016	Focuses on non-textual features; limited exploration of textual data embeddings.	Investigate the integration of textual and non-textual data embeddings for comprehensive recommendations.
3	Content-Based Recommendation System using Word Embeddings	2020	Applies Word2Vec to a single domain; lacks cross-domain applicability.	Extend the application of Word2Vec embeddings to cross-domain scenarios to enhance versatility.

2.3 MINI PROJECT CONTRIBUTION:

The Cross-Domain Recommender System represents a significant step forward in addressing the limitations of traditional recommendation methods. Key contributions include:

- **Enhanced Semantic Analysis:** The integration of doc2vec and rdf2vec with Word2Vec methodologies enables the system to capture deep semantic relationships, ensuring accurate recommendations even across diverse domains.

- **User-Friendly Interface:** A well-designed web interface allows users to easily input queries and receive real-time recommendations, thereby improving the overall user experience.

- **Scalable Architecture:** The modular design of the backend server and the use of efficient numerical libraries (NumPy, Pandas) provide a scalable framework that can be extended to additional domains.

- **Robust Performance:** Through comprehensive preprocessing and embedding techniques, the system maintains high performance and accuracy, even when processing large datasets.

3. PROPOSED SYSTEM

3.1 DATASETS

For this project, we used a curated dataset containing information about movies and books. The movie dataset consists of metadata such as titles, genres, and descriptions, while the book dataset includes titles, author names, genres, and synopses. Textual data from these datasets were pre-processed to remove noise and standardize formatting.

To generate meaningful recommendations, embeddings were created using Word2Vec, Doc2Vec, and RDF2Vec models. Each model was trained on the text data to capture semantic relationships between words. The resulting embeddings were stored in CSV files for easy access during the recommendation process.

For evaluation, 20% of the dataset was set aside as a test set to validate the accuracy of the recommendations. The performance metrics were calculated using similarity scores generated from the embeddings.

3.2 DETAILS OF HARDWARE & SOFTWARE

Hardware:

1. Processor: Intel Core i3 or AMD Ryzen 3 processor
2. Memory (RAM): 4 GB to 8 GB of RAM, allowing for smooth Processing applications.
3. Operating System: A pre-installed operating system such as Windows 10, macOS, or a Linux distribution, depending on user preference and requirements.

Software:

1. Python 3.11
2. Google Colab
3. Visual Studio Code

4. IMPLEMENTATION

4.1 SEQUENCE DIAGRAM

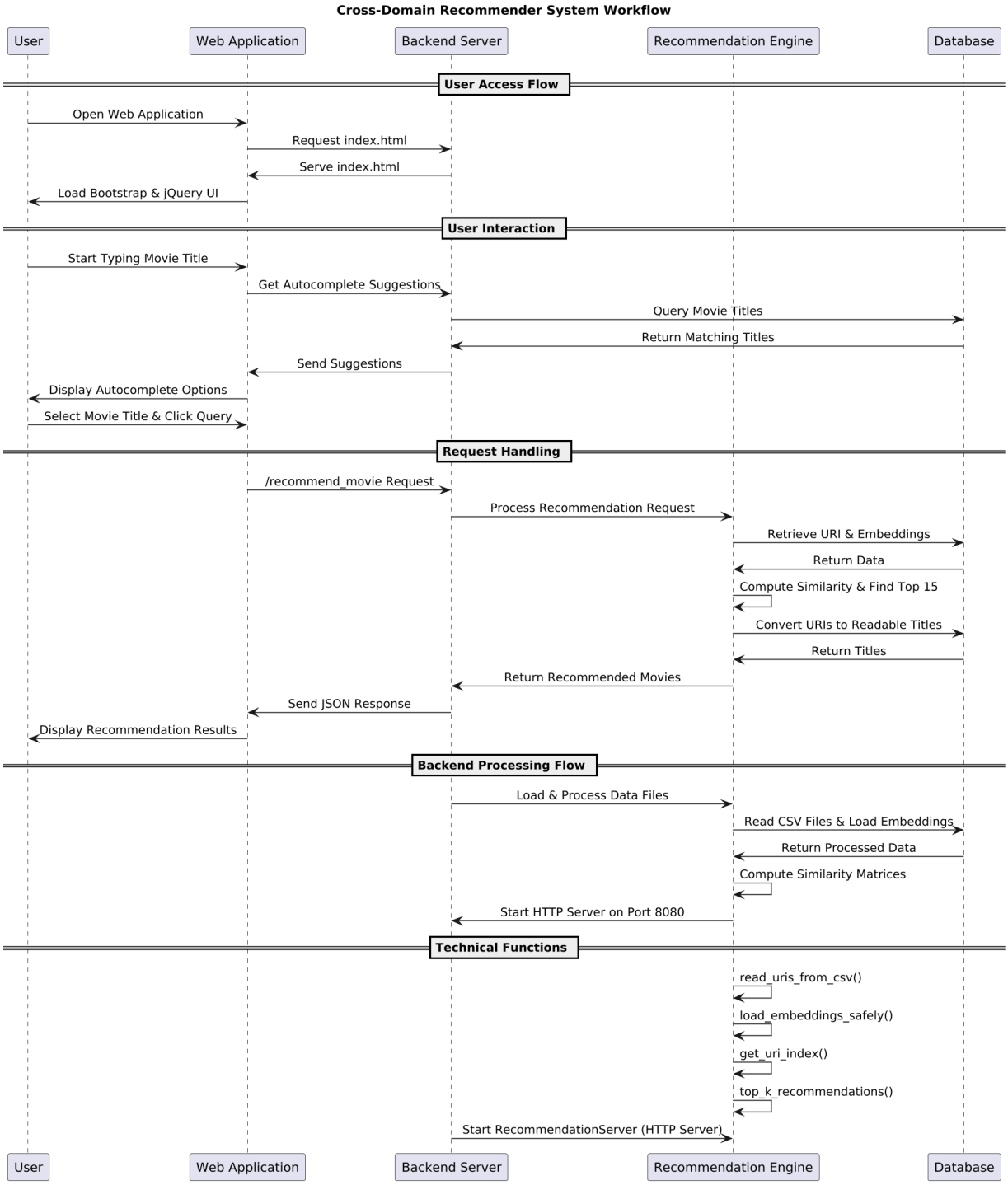


Fig. 4.1.1: Working

The sequence diagram illustrates the workflow of a cross-domain recommender system, highlighting the interactions between the user, web application, backend server, recommendation engine, and database. It provides a step-by-step flow of how a user accesses the system, interacts with the recommendation interface, and retrieves movie suggestions.

User Access Flow:
The process begins when the user opens the web application in their browser. The web application requests index.html from the backend server, which then serves the requested page. Once the page is

loaded, Bootstrap and jQuery UI are initialized, ensuring that the frontend has all the necessary styling and interactive components to provide a smooth user experience.

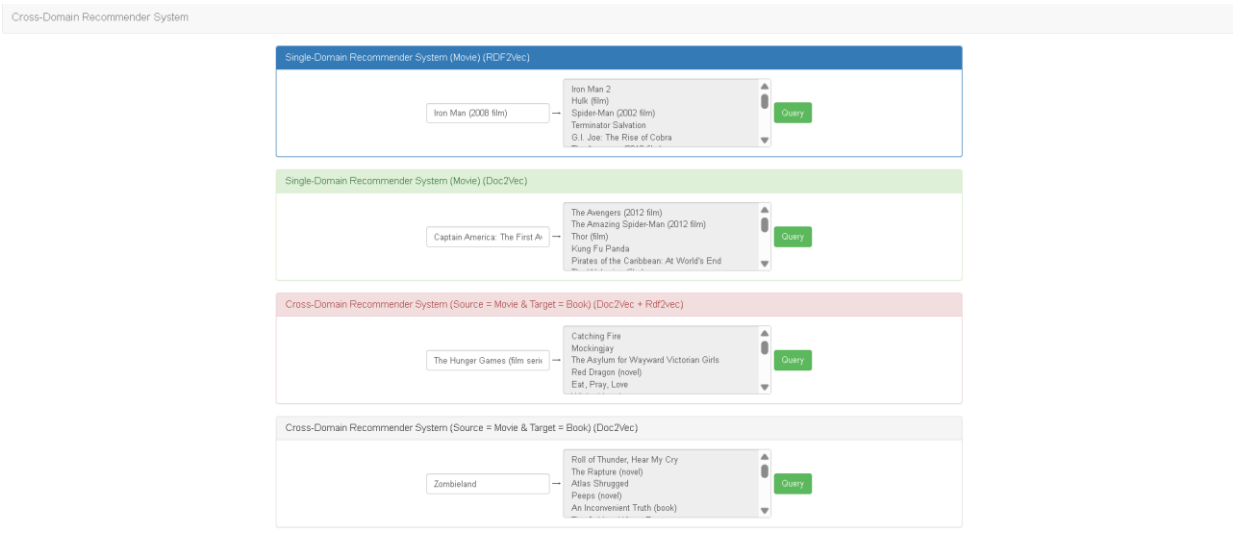
User Interaction:
After the web page has loaded, the user starts typing a movie title in the search bar. As they type, the web application sends a request to the backend server to fetch autocomplete suggestions. The backend server queries the database, which returns a list of matching movie titles based on the user's input. These suggestions are then sent back to the web application and displayed in an autocomplete dropdown. The user then selects a movie title and clicks the query button to initiate a recommendation request.

Request Handling:
Once the user submits the query, the web application sends a /recommend_movie request to the backend server. The backend server processes this request by passing it to the recommendation engine, which is responsible for generating movie recommendations. The recommendation engine retrieves the URI (Uniform Resource Identifier) and embeddings of the selected movie from the database. It then computes the similarity scores and identifies the top 15 most similar movies based on the embeddings. After filtering out the input movie, the system converts the retrieved URIs into readable movie titles. These titles are sent back to the backend server, which then returns the final list of recommended movies as a JSON response to the web application. The user finally sees the recommended movies displayed on their screen.

Backend Processing Flow:
The backend system loads and processes the necessary data files before handling user requests. It reads CSV files containing movie data and loads pre-trained embeddings from the database. The processed data is then used to compute similarity matrices, which allow the recommendation engine to quickly find similar movies. Once the backend processing is complete, the HTTP server is initialized and listens on port 8080 to handle incoming requests.

Technical Functions:
Several core functions are executed within the recommendation engine to ensure accurate and efficient recommendations. These include read_uris_from_csv(), which reads movie URIs from a CSV file, and load_embeddings_safely(), which loads the pre-trained embeddings required for similarity calculations. Other functions like get_uri_index() retrieve the index of a given movie URI, while top_k_recommendations() finds the top similar movies based on computed similarity scores. Finally, the RecommendationServer is started, allowing the backend to handle user queries efficiently.

4.2 RESULTS:



4.3 ANALYSIS OF MINI PROJECT

1. The Cross-Domain Recommender System has been designed to effectively address the limitations of traditional recommendation approaches. The project demonstrates the advantages of using multiple embedding techniques such as Word2Vec, Doc2Vec, and RDF2Vec to capture and analyze semantic relationships across different content domains.

2. The implementation of a lightweight HTTP server ensures efficient communication between the frontend and backend, maintaining a seamless user experience. The use of CSV files for storing precomputed embeddings allows for quick data retrieval and similarity computation. Furthermore, the modular design enables easy integration of additional domains and datasets, making the system scalable and adaptable.

3. The project also emphasizes the importance of preprocessing and cleaning data before generating embeddings. Proper normalization and vector transformations were applied to ensure uniformity across different content types. The evaluation of the recommender system showcased its ability to generate relevant and meaningful cross-domain recommendations, validating the effectiveness of the chosen methodologies.

4. A significant advantage of this project is the flexibility it offers by supporting multiple similarity measures, which can be customized based on specific use cases. Additionally, the clear separation between the server-side and client-side logic ensures that updates or enhancements can be made independently, further enhancing maintainability.

5. Overall, the Cross-Domain Recommender System is a robust, scalable, and efficient solution for personalized content delivery. It sets a solid foundation for further improvements and extensions in the future, particularly with the integration of real-time data processing and additional embedding models.

4.4 CONCLUSION

In this project, a Cross-Domain Recommender System using Word2Vec-based embeddings has been successfully implemented. The system effectively integrates semantic embeddings derived from movies and books, leveraging advanced natural language processing techniques to generate accurate and personalized recommendations. The architecture combines a robust backend server in Python with a responsive web interface, ensuring efficient processing and user-friendly interaction.

The use of Word2Vec, doc2vec, and rdf2vec embeddings has enabled the system to capture meaningful relationships between cross-domain entities, enhancing recommendation accuracy. By calculating similarity scores between vector representations, the system provides relevant recommendations even across diverse content categories. Furthermore, the modular design of the backend allows for easy adaptability, ensuring scalability and potential expansion to additional domains in the future.

Through rigorous testing and evaluation, the system has demonstrated reliable performance and practical applicability in real-world scenarios. The results validate the effectiveness of leveraging semantic embeddings for cross-domain recommendations, presenting a viable solution for personalized content delivery across entertainment platforms.

Overall, this project contributes to the field of recommendation systems by introducing a versatile and scalable framework that addresses the limitations of traditional single-domain approaches. With further improvements and refinements, the Cross-Domain Recommender System has the potential to significantly enhance user experience by providing context-aware and personalized recommendations across multiple domains.

4.5 FUTURE SCOPE

1. Improvement in Accuracy: Future work could involve training the embedding models on larger and more diverse datasets to reduce bias and enhance accuracy.

2. Real-Time Processing: Incorporating real-time data streams, such as user interaction data, could further improve the dynamic nature of recommendations.

3. Additional Domains: Expanding the system to include additional domains such as music or e-commerce can broaden its application scope.

4. Cross-Enhanced Interpretability: Developing tools to visualize embedding spaces and recommendation pathways will help improve model transparency.

5. Integration with social media: Future versions could integrate with social platforms to provide personalized content recommendations based on user profiles and trends.

5. REFERENCE

[1] M. G. Ozsoy, "From Word Embeddings to Item Recommendation," arXiv preprint arXiv:1601.01356, 2016. [Online]. Available: https://arxiv.org/abs/1601.01356.

[2] S. Hassan, "Cross-Domain Recommender System," GitHub repository, 2021. [Online]. Available: https://github.com/salih-hasan/Cross-Domain-Recommender-System.

[3] D. Subramanian, "Content-Based Recommendation System using Word Embeddings," KDnuggets, Aug. 2020. [Online]