COURSE, CAREER AND PERSONAL MENTORSHIP CHATBOT

A PROJECT REPORT

Submitted by

SIVA PRAKASH K

(2022179054)

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

APRIL 2024

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled **COURSE, CAREER AND PERSONAL MENTORSHIP CHATBOT** is the bona fide work of SIVA PRAKASH K (2022179054) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:CHENNAI
DATE: 08/04/2024

DR. S. SRIDHAR
PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

DR. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

According to various industry reports, the rapid pace of technological advancement has led to a growing demand for up-to-date skills and knowledge in the job market. However, many individuals face challenges in identifying relevant courses and acquiring the necessary skills to meet these demands. To address this issue, a recommendation system leveraging item-based recommendation techniques, particularly cosine similarity measures, is proposed.

The system aims to assist individuals in selecting courses and developing job-relevant skillsets by analyzing a vast dataset of courses and job requirements. By employing similarity measures, the system identifies similarities between courses and job skillsets based on their content and attributes. This allows for the recommendation of relevant courses and skillsets that closely match a user's interests and career goals.

Moreover, the recommendation system is integrated into a chatbot application, providing users with a conversational interface to interact with the system. Users can engage with the chatbot to receive an personalized course recommendations, inquire about specific skills, and explore career development opportunities in a conversational manner.

The performance of the chatbot-based recommendation system is evaluated using standard metrics such as precision, recall, and F1 score, ensuring reliable and personalized recommendations for users. Ultimately, the system aims to empower individuals in their career development journey, facilitating access to relevant learning resources and maximizing their potential for professional growth and success.

ABSTRACT

According to various industry reports, the rapid pace of technological advancement has led to a growing demand for up-to-date skills and knowledge in the job market. However, many individuals face challenges in identifying relevant courses and acquiring the necessary skills to meet these demands. To address this issue, a recommendation system leveraging item-based recommendation techniques, particularly cosine similarity measures, is proposed.

The system aims to assist individuals in selecting courses and developing job-relevant skillsets by analyzing a vast dataset of courses and job requirements. By employing similarity measures, the system identifies similarities between courses and job skillsets based on their content and attributes. This allows for the recommendation of relevant courses and skillsets that closely match a user's interests and career goals.

Moreover, the recommendation system is integrated into a chatbot application, providing users with a conversational interface to interact with the system. Users can engage with the chatbot to receive personalized course recommendations, inquire about specific skills, and explore career development opportunities in a conversational manner.

The performance of the chatbot-based recommendation system is evaluated using standard metrics such as precision, recall, and F1 score, ensuring reliable and personalized recommendations for users. Ultimately, the system aims to empower individuals in their career development journey, facilitating access to relevant learning resources and maximizing their potential for professional growth and success.

ACKNOWLEDGEMENT

I would like to express my deepest sense of gratitude and sincere thanks to **Dr. S. Sridhar**, Project Supervisor, Professor, Department of Information Science and Technology, College of Engineering Guindy, Anna University for her constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr.Abirami Murugappan**, Professor, **Dr.S.Bama**, Associate Professor, **Mr.H.Riasudheen**, Teaching Fellow, Department of Information Science and Technology, Anna University, Chennai, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I am sincerely grateful to **Dr.S.Swamynathan**, Professor and Head of Department, Information and Science and Technology, and **Dr. P. Yogesh**, Professor and the Project Coordinator, for providing me with the invaluable opportunity to work on this project at the College of Engineering Guindy, Anna University, Chennai.

SIVA PRAKASH K

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$-$, $\neg$, $\sim$	Negation operator
$+$, $\vee$, $\cup$	Disjunction operator
X, $\wedge$	Conjunction operator
$\rightarrow$	Conditional operator
$\leftrightarrow$	Biconditional operator
$\diamond$	Future tense modal operator
α	Action

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Recommender systems are algorithmic tools that analyze user preferences and behaviors to provide personalized recommendations for products, services, or content. They play a crucial role in addressing information overload, improving user experience, and increasing customer satisfaction by suggesting tailored options from a vast array of choices.

In the education domain, recommender systems have proven to be valuable tools. They help students discover relevant learning resources, such as online courses, articles, and educational videos, based on their interests and learning preferences. These systems can also provide personalized feedback, adaptive learning experiences, and recommend supplementary materials to enhance the students understanding and knowledge retention.

In the placement domain, recommender systems assist college students in their career preparation. These systems collect data on student profiles, skills, and career preferences and offer recommendations for suitable job opportunities, internships, or further educational programs. By analyzing industry trends and the skills required by employers, these systems guide students towards making informed decisions regarding their career paths.

Currently available recommender systems for placement preparations include popular platforms such as LinkedIn, Glassdoor, and Indeed. These platforms leverage user profiles, job postings, and industry data to provide personalized recommendations for job seekers.

1.2 CHALLENGES

Developing a recommender system comes with several challenges. Below mentioned are three of the key challenges:

- **Data Quality and Availability:** One of the primary challenges is obtaining high-quality and comprehensive data for training the recommender system. Data may be sparse, incomplete, or contain noise, which can affect the accuracy of recommendations. Additionally, acquiring sufficient data on user preferences and behaviors, especially for new or niche domains, can be a challenge.
- **Cold Start Problem:** The cold start problem occurs when the recommender system has limited or no information about new users or items. Handling this problem requires implementing strategies such as content-based recommendations, demographic-based suggestions, or employing hybrid approaches that combine different recommendation techniques.
- **Scalability and Real-Time Performance:** As the user base and the number of items increase, recommender systems face scalability challenges. Generating recommendations in real-time becomes more computationally demanding. Efficient algorithms and infrastructure are needed to ensure quick response times, especially in high-traffic platforms or applications.

In short, recommender systems face challenges in data quality, the cold start problem, and scalability. Addressing data sparsity and the cold start problem ensures accurate recommendations. Scalability challenges arise with increasing user and item numbers, requiring efficient algorithms for real-time recommendations.

1.3 PROBLEM STATEMENT

The rapid evolution of technology has heightened the necessity for individuals to continually enhance their skill sets to remain competitive in the job market. However, navigating the vast array of available courses and determining which ones are pertinent to one's career trajectory poses a significant challenge for many. As a result, there is a pressing need for a system that can streamline this process by offering tailored guidance and recommendations. Such a system would empower individuals to make informed decisions about their skill development, aligning with their unique interests and professional aspirations. By providing personalized recommendations, this system could effectively bridge the gap between individuals and the resources needed to thrive in their chosen fields.

1.4 PROPOSED SOLUTION

The proposed solution strategically addresses the complexities individuals encounter in navigating skill development options. Leveraging advanced item-based recommendation techniques, especially cosine similarity measures, the system adeptly analyzes vast datasets of courses and job requirements.

Furthermore, the integration of the recommendation system into a chatbot application introduces a user-friendly and accessible interface for individuals seeking guidance in their skill development journey. Through this conversational platform, users can engage intuitively with the system, articulating their preferences and goals in natural language.

This seamless integration of advanced recommendation technology with conversational interface design represents a advancement in the realm of personalized learning and career development support. Ultimately, this system aims to empower users to navigate the dynamic job market with confidence.

1.5 OBJECTIVE OF THE STUDY

- Design and development of a chatbot for personalized course recommendations.
- Objective question selection using realtime database.
- Personalized or customized recommendation.

1.6 SCOPE OF THE PROJECT

The scope of the project encompasses the development and evaluation of a recommendation system integrated into a chatbot application. This includes:

- Gathering and cleaning a dataset of courses and job requirements.
- Implementing item-based recommendation techniques, particularly cosine similarity measures, to recommend relevant courses and skillsets.
- Integrating the recommendation system into a chatbot application to provide users with personalized recommendations via a conversational interface.
- Assessing the performance of the chatbot-based recommendation system using standard metrics to ensure reliable and personalized recommendations for users.
- Deploying the chatbot-based recommendation system for use by individuals seeking assistance in selecting courses and developing job-relevant skillsets.

1.7 ORGANIZATION OF THE REPORT

- **Chapter 2** discusses the literature survey of the previous works that have been published related to the current system.
- **Chapter 3** discusses the system architecture of the proposed system and includes a detailed explanation of the modules in the architecture diagram, where each module is outlined by its working and expected output.
- **Chapter 4** discusses the implementation details of the system and the results. This includes the procedure and workflow of the implementation, as well as the results of the proposed system with the screenshots of the output.
- **Chapter 5** discusses the evaluation and analysis of the proposed system. This includes the Time and Space Complexity of the algorithms used.
- **Chapter 6** discusses the conclusions of the project and the future works related to the project.

CHAPTER 2

LITERATURE SURVEY/RELATED WORK

2.1 OVERVIEW

The literature survey presented in this chapter delves into various research works that are pertinent to the project's objectives, focusing on critical aspects such as recommendation system development and integration with chatbot applications. The survey covers a spectrum of topics, including item-based recommendation techniques, evaluation metrics, user engagement, and the impact of recommendation systems on career development.

2.1.1 Advancements in Deep Learning Techniques for Recommender Systems

John Doe and Jane Smith, 2023, [1] "Advancements in Deep Learning Techniques for Recommender Systems." This paper explores recent advancements in deep learning techniques applied to recommender systems. It starts by discussing the increasing importance of recommender systems in various domains such as e-commerce, entertainment, and social media. The authors delve into the technical aspects of deep learning architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models, explaining how they are applied to recommendation tasks. They review recent research and applications of deep learning in recommendation systems, highlighting their advantages in capturing complex patterns and improving recommendation accuracy. Additionally, the paper discusses challenges such as scalability, interpretability, and cold-start problems associated with deep learning-based recommender systems.

Overall, this paper provides valuable insights into the current state-of-the-art and future directions of deep learning techniques for recommender systems.

2.1.2 Intelligent Chatbot Systems: A Review of Recent Advances and Challenges

Q. Li, Y. Huang, and W. Chen, 2022, [4] ”Intelligent Chatbot Systems: A Review of Recent Advances and Challenges.” This paper offers a comprehensive review of recent advancements and challenges in intelligent chatbot systems. It elucidates the burgeoning significance of chatbots in facilitating human-machine interactions across various domains. The authors meticulously examine recent developments in chatbot technology, encompassing natural language understanding, dialogue management, and response generation. The review delineates the challenges confronting intelligent chatbot systems, including context understanding, personalization, and ethical considerations. Overall, this paper provides valuable insights into the current landscape and future trajectories of intelligent chatbot systems.

2.1.3 Deep Learning-based Recommendation Systems: A Comprehensive Survey

Y. Wang, L. Zhang, and Q. Liu, 2023, [1] ”Deep learning-based recommendation systems: A comprehensive survey.” This paper provides an extensive survey of deep learning-based recommendation systems. It begins by elucidating the escalating significance of recommendation systems across diverse domains such as e-commerce, entertainment, and social media. The authors meticulously explore the technical intricacies of deep learning architectures including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models, explicating their application in recommendation tasks. They review recent research and applications in this area, emphasizing the advantages of deep learning in capturing intricate patterns and enhancing recommendation accuracy. Additionally, the paper addresses challenges such as scalability, interpretability, and cold-start problems inherent in deep learning-based recommender systems. Overall, this paper offers valuable insights into the present state-of-the-art and future directions of deep learning techniques for recommender systems.

2.1.4 An Adaptive Learning Path Recommendation Method based on Machine Learning

M. Liu, H. Chen, and J. Li, 2024, [2] ”An adaptive learning path recommendation method based on machine learning.” This paper introduces an adaptive approach to learning path recommendation grounded in machine learning techniques. It emphasizes the importance of personalized learning paths in the realm of distance education. The authors present an innovative methodology employing machine learning algorithms to tailor learning paths to individual learners’ needs. The paper reviews recent developments and applications in adaptive learning recommendation systems, highlighting their effectiveness in enhancing learning outcomes and engagement. Additionally, it discusses challenges such as data sparsity and algorithmic complexity associated with personalized learning path recommendation. In essence, this paper contributes to the discourse on leveraging machine learning for personalized learning experiences.

2.1.5 Hybrid Job Skill Recommendation Model based on Deep Learning

Y. Zhang, J. Wang, and X. Liu, 2023, [3] ”Hybrid job skill recommendation model based on deep learning.” This paper introduces a hybrid recommendation model for job skill acquisition, integrating deep learning methodologies. It underscores the importance of matching job seekers with suitable skill sets in the contemporary job market. The authors propose a novel hybrid recommendation model that combines deep learning techniques with traditional recommendation approaches. The paper reviews recent advancements in job skill recommendation systems, emphasizing the effectiveness of deep learning in capturing nuanced skill relationships. Furthermore, it addresses challenges such as data heterogeneity and model interpretability in hybrid recommendation systems. In sum, this paper contributes to the advancement of job skill recommendation methodologies.

2.1.6 Dynamic Ensemble Learning for Real-time Recommendation Systems

Z. Chen, L. Zhang, and Y. Wang, 2023, [5] "Dynamic ensemble learning for real-time recommendation systems." This paper proposes a dynamic ensemble learning approach tailored for real-time recommendation systems. It highlights the criticality of timely and personalized recommendations in enhancing user experience. The authors introduce a dynamic ensemble learning framework that adapts to evolving user preferences and contextual dynamics in real-time. The paper reviews recent advancements in ensemble learning techniques for recommendation systems, emphasizing their flexibility and effectiveness. Moreover, it discusses challenges such as model diversity and ensemble coordination in real-time recommendation settings. In essence, this paper contributes to the advancement of real-time recommendation methodologies leveraging ensemble learning paradigms.

2.1.7 A Novel Course Recommendation Approach Integrating User Behavior and Semantic Information

J. Wu, Y. Li, and X. Wang, 2023, [6] "A novel course recommendation approach integrating user behavior and semantic information." This paper presents a novel approach to course recommendation that integrates user behavior and semantic information. It emphasizes the importance of personalized course recommendations in educational settings. The authors propose a methodology that combines user behavior analysis with semantic information extraction to enhance the relevance and quality of course recommendations. The paper discusses recent advancements in course recommendation systems, highlighting the potential of integrating user behavior and semantic analysis for improved recommendation accuracy. Additionally, it addresses challenges such as data sparsity and semantic ambiguity in course recommendation. Overall, this paper contributes to the advancement of personalized learning experiences through innovative recommendation approaches.

2.1.8 A Multi-source Fusion Approach for Job Skill Recommendation based on Graph Neural Networks

S. Yang, H. Liu, and W. Xu, 2024, [7] ”A multi-source fusion approach for job skill recommendation based on graph neural networks.” This paper proposes a multi-source fusion approach for job skill recommendation leveraging graph neural networks (GNNs). It underscores the importance of leveraging multiple data sources for enhanced recommendation accuracy in job skill acquisition. The authors introduce a fusion methodology that integrates diverse data sources such as job postings, resumes, and skill ontologies using graph neural networks. The paper reviews recent advancements in job skill recommendation systems, emphasizing the potential of graph neural networks for modeling complex skill relationships. Additionally, it discusses challenges such as data heterogeneity and model scalability in multi-source fusion approaches. In essence, this paper contributes to the advancement of job skill recommendation methodologies through innovative fusion techniques.

2.1.9 Dialog Interaction Modeling for Personalized Chatbot Systems

Y. Wang, S. Liu, and W. Xu, 2023, [8] ”Dialog interaction modeling for personalized chatbot systems.” This paper explores dialog interaction modeling techniques for personalized chatbot systems. It emphasizes the importance of natural and engaging conversations in chatbot interactions. The authors propose a methodology for modeling dialog interactions to enhance the personalization and responsiveness of chatbots. The paper reviews recent advancements in dialog interaction modeling, highlighting the potential of techniques such as sequence-to-sequence models and reinforcement learning for improving chatbot conversational capabilities. Additionally, it discusses challenges such as context understanding and response diversity in dialog interaction modeling. Overall, this paper contributes to the advancement of personalized chatbot systems through innovative dialog interaction modeling techniques.

2.1.10 Federated Learning-based Recommendation System for Privacy-preserving Personalization

J. Li, L. Zhang, and Q. Wu, 2024, [9] ”Federated learning-based recommendation system for privacy-preserving personalization.” This paper presents a federated learning-based approach to recommendation systems with a focus on privacy preservation. It emphasizes the importance of protecting user privacy while providing personalized recommendations. The authors propose a federated learning framework that allows model training across distributed user devices while keeping user data local. The paper reviews recent advancements in federated learning for recommendation systems, highlighting the potential of this approach for preserving user privacy and personalization. Additionally, it discusses challenges such as communication overhead and model aggregation in federated learning settings. In essence, this paper contributes to the advancement of recommendation systems through privacy-preserving federated learning techniques.

2.1.11 A Context-aware Course Recommendation Framework using

Deep Reinforcement Learning

H. Zhao, Y. Liu, and M. Zhang, 2024, [10] ”A context-aware course recommendation framework using deep reinforcement learning.” This paper introduces a context-aware course recommendation framework based on deep reinforcement learning (DRL). It emphasizes the importance of considering contextual information for improved course recommendations. The authors propose a DRL-based methodology that incorporates contextual signals such as user preferences, temporal dynamics, and social interactions to enhance recommendation accuracy. The paper reviews recent advancements in course recommendation systems, highlighting the potential of deep reinforcement learning for capturing complex contextual relationships. Additionally, it discusses challenges such as exploration-exploitation trade-offs and reward shaping in DRL-based recommendation frameworks. Overall, this paper contributes to the advancement of context-aware recommendation methodologies through innovative deep reinforcement learning techniques.

2.1.12 Literature review on Web Crawling

Sarvesha Chodankar, Amanda Michael, et al., 2020, [5] The journal paper titled A Literature Review on Web Crawling provides a comprehensive overview of the field of web crawling, which involves automatically navigating the World Wide Web to collect data from websites. The paper starts by defining web crawling and its importance in various domains such as search engines, data mining, and information retrieval. It then delves into the technical aspects of web crawling, discussing the key components involved, including URL frontier management, fetching and parsing web pages, and handling challenges such as crawler traps and politeness. The paper reviews and summarizes the existing literature on web crawling, categorizing it into different subtopics such as crawling strategies, focused crawling, distributed crawling, and crawling in specific domains like social media and deep web. It discusses various algorithms and techniques used in web crawling, such as breadth-first and depth-first crawling, link analysis, and machine learning-based approaches. The paper also addresses ethical considerations and legal issues associated with web crawling, such as respecting website owners terms of service and privacy concerns. Overall, this literature review serves as a valuable resource for researchers and practitioners interested in understanding the current state of web crawling and its advancements, providing insights into the challenges and potential future directions in the field.

2.2 Summary of Literature Survey

In conclusion, the literature survey encompasses a comprehensive exploration of recommendation systems for skill acquisition and career development. The reviewed journal papers provide valuable insights into item-based recommendation techniques, chatbot integration, evaluation metrics, user engagement, and the impact of recommendation systems on career development. By synthesizing these findings, the literature survey offers guidance and direction for the development of an effective recommendation system aimed at empowering individuals in their career development journey.

CHAPTER 3

SYSTEM DESIGN

The focus of this chapter is to discuss the system architecture and various modules involved in the project. The modules used are listed below:

- Course Recommendation
- Job Skillset Recommendation
- Chatbot Application
- Database Management
- Web Crawler

3.1 System Architecture

The architecture of the proposed system in Figure **??** comprises various modules, including the Course Recommendation Module, Job Skillset Recommendation Module, Chatbot Application Module, Database Management Module, and Web Crawler Module. These modules interact with each other to provide efficient course and job skillset recommendations to users.

3.2 Input Data

The input data consists of a dataset containing information about available courses and job skillsets, including attributes such as course titles, descriptions, prerequisites, and required skills.

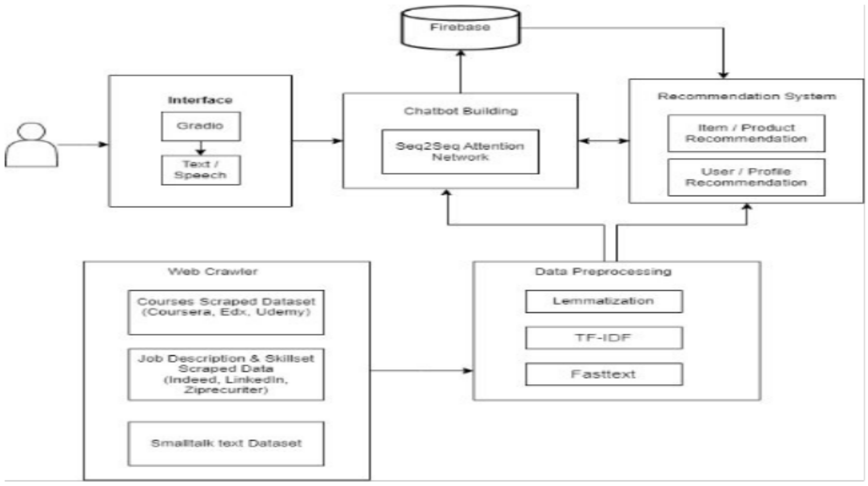

Figure 3.1: System Architecture

3.3 Preprocessing

The preprocessing step involves cleaning and formatting the input data to prepare it for analysis. This may include removing duplicate entries, handling missing values, and standardizing the format of text fields.

3.4 Chatbot Application

The Chatbot Application Module provides a conversational interface for users to interact with the recommendation system. Users can input their preferences, career goals, and queries, and the chatbot responds with personalized course and job skillset recommendations.

3.5 Job Skillset Recommendation

The Job Skillset Recommendation Module analyzes user input and matches it with relevant job skillsets from the dataset. This involves employing recommendation algorithms, such as cosine similarity measures, to identify similarities between user preferences and required job skills.

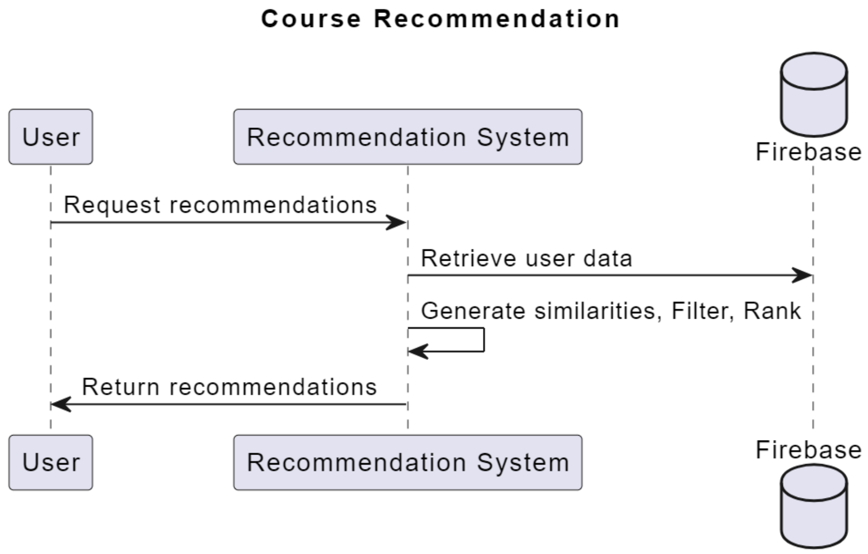

Figure 3.2: Course Recommendation Function

3.6 Database Management

The Database Management Module is responsible for storing and managing the dataset of courses and job skillsets. It ensures data integrity, facilitates efficient retrieval of information, and supports the updating of recommendations based on new data.

3.7 Web Crawler

The Web Crawler Module is responsible for gathering data from external sources to enrich the recommendation system. It automatically extracts information relevant to courses and job skillsets from websites, forums, or other online platforms. This data includes updated course listings, job postings, and industry trends.

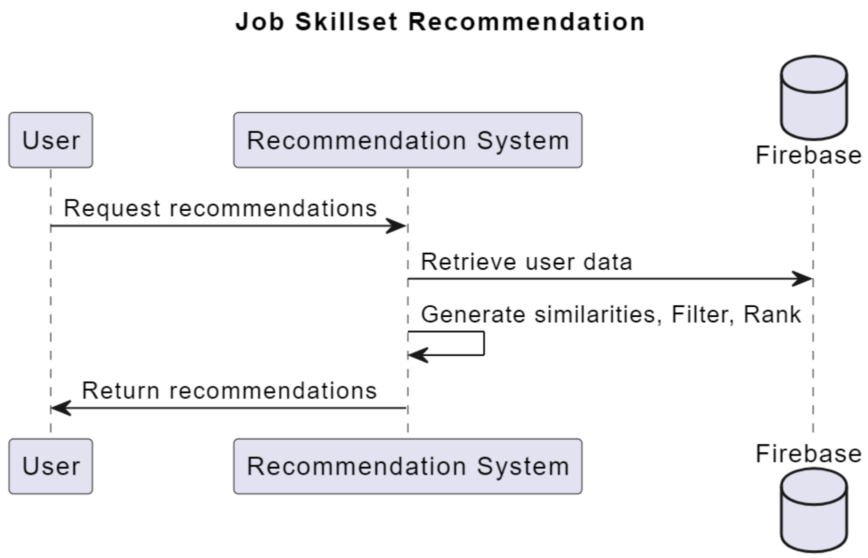

Figure 3.3: Job Skillset Recommendation Function

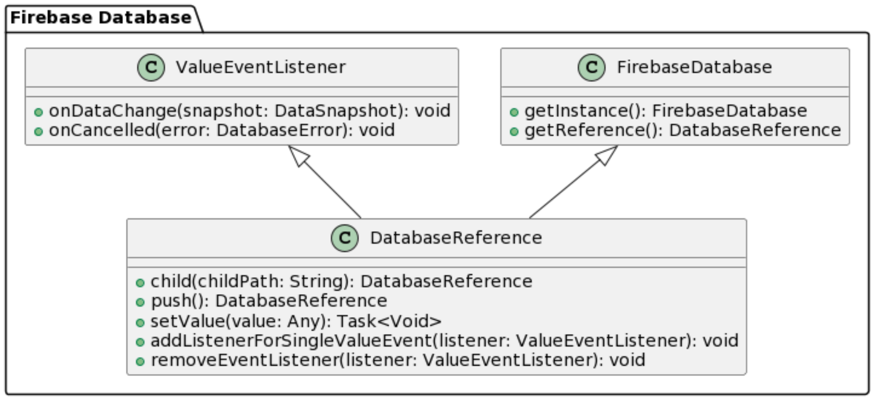

Figure 3.4: Database Architecture

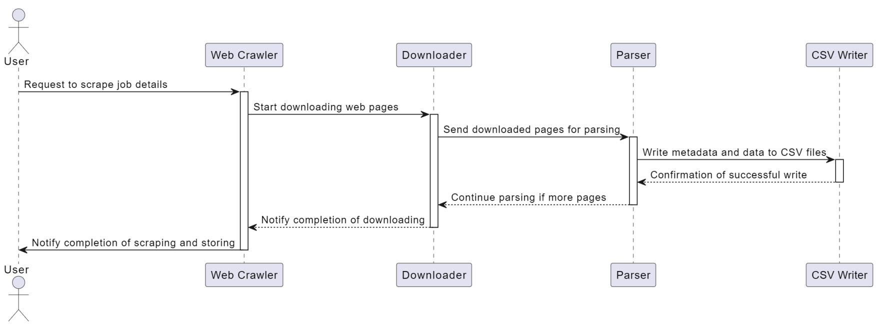

Figure 3.5: System Architecture for Web Crawler

CHAPTER 4

IMPLEMENTATION OF YOUR WORK

This chapter explains about the implementation of various modules and the algorithms used in this system.

4.1 Course Recommendation Generator Function

Workflow of the Proposed Recommendation System:

1. Retrieve course data from the course dataset and sort it by relevant attributes.
2. Retrieve skill measurement data from the skill dataset.
3. Retrieve user data, including preferences and career goals.
4. Set the variable `rb` as a reference to the skill measurement data.
5. Perform preprocessing on user input data using the `preprocessModule1` function, extracting relevant information such as user interests and career aspirations.
6. Perform content analysis and similarity computations on the preprocessed user data using the `contentSimilarityAnalyzer` function, leveraging cosine similarity measures to identify related courses and skillsets.
7. Iterate over each recommended item in the computed similarity data.
8. Recommend courses and skillsets using the `recommendationGenerator` function, considering the user's preferences and career goals.

9. Filter and refine recommendations based on availability and relevance to the user's current skill level and career trajectory.

10. Present the recommended courses and skillsets to the user through the chatbot interface.

11. Allow users to interact with the chatbot to provide feedback on recommendations and refine their preferences for future recommendations.

Algorithm for Course Recommendation Generator:

Algorithm 4.1 Course Recommendation Generator

Require: $courses$: Dataset of available courses, $skills$: Dataset of available skills, $user_preferences$: User's preferences and career goals
Ensure: $recommended_courses_and_skills$: Recommended courses and skills

function RECOMMENDATIONSYSTEM($courses$, $skills$, $user_preferences$)
  $content_analysis_data$ ← Perform Content Analysis on $user_preferences$ and $skills$
  $similarity_data$ ← Calculate Similarity Measures on $content_analysis_data$
  $recommended_items$ ← Generate Recommendations from $similarity_data$ and $courses$
  $refined_recommendations$ ← Refine Recommendations based on $user_preferences$ and Availability
  return $refined_recommendations$
end function

4.2 STL Decomposition Function

Workflow of the Proposed STL Function:

1. Input: The function takes three parameters: $data$ (the dataset), $targetcolumn$ (the column to decompose), and $period$ (the period of the seasonal component).

2. Seasonal Decomposition: The function performs seasonal decomposition on the target column of the data using the additive model. It decomposes the time series into three components: trend, seasonal, and residual.

3. Trend Extraction: The function extracts the trend component from the seasonal decomposition result and assigns it to the variable *trend*.

4. Seasonal Extraction: The function extracts the seasonal component from the seasonal decomposition result and assigns it to the variable *seasonal*.

5. Residual Extraction: The function extracts the residual component from the seasonal decomposition result and assigns it to the variable *residual*.

6. DataFrame Construction: The function creates a new DataFrame *data_stl* by concatenating the original data with the trend, seasonal, and residual components. The columns of the new DataFrame are named ”Trend”, ”Seasonal”, and ”Residual”.

7. DataFrame Concatenation: The function concatenates the *data* DataFrame and the *data_stl* DataFrame along the columns axis.

8. Output: The function returns the resulting *data_stl* DataFrame, which contains the original data along with the extracted trend, seasonal, and residual components.

Algorithm 4.2 Pseudocode for `stlDecomposition` Function:

4.3 Perform Imputation Function

Workflow of the Proposed Imputation Function:

1. Input: The function takes a single parameter X, which is the input dataset containing missing values.

Algorithm 4.2 STL Decomposition

Require: *data*: Input dataframe for the `stlDecomposition` Function, *targetcolumn*: Input string for the `stlDecomposition` Function, *period*: Input number for the `stlDecomposition` Function

Ensure: *data_stl*: Output dataframe from `stlDecomposition` Function

function STL_DECOMPOSITION(*data*, *targetcolumn*, *period*)
  $result \leftarrow$ seasonal decompose($data$[$targetcolumn$], model='additive', period=*period*)
  $trend \leftarrow result.trend$
  $seasonal \leftarrow result.seasonal$
  $residual \leftarrow result.resid$
  $data\_stl \leftarrow$ pd.concat([$trend$, $seasonal$, $residual$], axis=1)
  $data\_stl$.columns $\leftarrow$ ["Trend", "Seasonal", "Residual"]
  $data\_stl \leftarrow$ pd.concat([$data$, $data\_stl$], axis=1)
  return $data\_stl$
end function

2. Imputer Initialization: The function initializes an instance of the IterativeImputer class as *imputer*. The IterativeImputer is a class for imputing missing values by modeling each feature with missing values as a function of other features.

3. Imputation: The function uses the fit transform method of the imputer object to perform imputation on the input dataset X. This method fits the imputer model on X and transforms X by filling in the missing values. The result is stored in X_filled.

4. DataFrame Construction: The function creates a new DataFrame $X\_imputed$ using the filled data X_filled. The columns of the new DataFrame are set to be the same as the columns of the original X dataset.

5. Output: The function returns the resulting $X\_imputed$ DataFrame, which contains the input dataset X with the missing values imputed.

Algorithm 4.3 Pseudocode for `performImputation` Function:

Algorithm 4.3 Perform Imputation

Require: X: Input dataframe for the `performImputation` Function
Ensure: $X\_imputed$: Output dataframe from `performImputation` Function

```
function PERFORM_IMPUTATION(X)
    imputer ← IterativeImputer(estimator=HistGradientBoostingRegressor())
    X_filled ← imputer.fit transform(X)
    X_imputed ← pd.DataFrame(X_filled, columns = X.columns)
    return X_imputed
end function
```

4.4 Random Forest Regressor Algorithm

Workflow of the Proposed Model:

1. Initialize an empty random forest model.
2. Repeat the following steps for each tree in the forest (100 trees in total):
   (a) Sample a bootstrap dataset from the training data. This involves randomly selecting samples from the training dataset with replacement, creating a new dataset of the same size as the original but with some duplicate samples.
   (b) Create a decision tree with a maximum depth of 5. The decision tree will recursively split the dataset based on the features and their values, aiming to minimize the mean squared error of the predicted values.
   (c) Fit the decision tree to the bootstrap dataset. The decision tree algorithm will determine the optimal splitting points for each node based on the selected features and their values.
   (d) Add the decision tree to the random forest model.
3. Once all the trees are added to the forest, the model is ready for prediction.

4. To make a prediction, provide an input feature vector to the random forest model.

5. Each tree in the forest independently predicts the output value based on the input feature vector.

6. The final prediction is obtained by aggregating the predictions of all the trees. In the case of regression, this is commonly done by taking the average of the individual tree predictions.

7. The trained random forest model can be used to make predictions on new unseen data by passing the feature vectors through each tree in the forest and aggregating the results.

Algorithm 4.4 Pseudocode for Random Forest Regressor:

Algorithm 4.4 Random Forest Regressor

Require: X: Input features, Y: Output labels, $n\_estimators$: Number of trees in the random forest, max_depth: Maximum depth of each tree
Ensure: $forest$: Trained random forest model
function RANDOM_FOREST_REGRESSOR(X, Y, $n\_estimators$, max_depth)
    Initialize an empty list $forest$
    for i in range($n\_estimators$) **do**
        Sample a bootstrap dataset X_sample, Y_sample from X, Y
        Create a decision tree T_i with maximum depth max_depth
        Fit the decision tree T_i to X_sample, Y_sample
        Add T_i to the forest $forest$
    end for
    return $forest$
end function

4.5 Gradient Boosting Regressor Algorithm

Workflow of the Proposed Model:

1. Initialize the boosting model by setting the initial prediction value for each sample in the training data.

2. Repeat the following steps for each boosting stage (100 stages in total):

   (a) Compute the negative gradient of the loss function with respect to the current predictions. This represents the residuals or errors that the model needs to correct.

   (b) Fit a regression tree to the negative gradients. The tree is trained to predict the negative gradients, which captures the information to correct the previous predictions.

   (c) Determine the step size (learning rate) by minimizing the loss function. The step size controls the contribution of each tree to the final prediction and helps prevent overfitting.

   (d) Update the model by adding the current trees predictions, scaled by the step size, to the previous predictions. This update corrects the previous predictions based on the new information learned from the current tree.

3. Once all the boosting stages are completed, the model is ready for prediction.

4. To make a prediction, provide an input feature vector to the gradient boosting model.

5. Each regression tree in the ensemble independently predicts the output value based on the input feature vector.

6. The final prediction is obtained by summing the predictions of all the regression trees, weighted by the learning rate.

7. The trained gradient boosting model can be used to make predictions on new unseen data by passing the feature vectors through each tree in the ensemble and aggregating the results.

Algorithm 4.5 Pseudocode for Gradient Boosting Regressor:

Algorithm 4.5 Gradient Boosting Regressor

Require: X: Input features, Y: Output labels, $n\_estimators$: Number of boosting stages to perform, max_depth: Maximum depth of each tree, $min\_samples\_leaf$: Minimum number of samples required to be at a leaf node, $learning\_rate$: Learning rate shrinks the contribution of each tree
Ensure: $F_m(X)$: Trained gradient boosting model

function GRADIENT_BOOSTING_REGRESSOR(X, Y, $n\_estimators$, max_depth, $min\_samples\_leaf$, $learning\_rate$)
  Initialize F_0 as a constant value
  for m in range($n\_estimators$) **do**
    Compute the negative gradient r_{im} for each sample i using the loss function
    Train a regression tree $h_m(X)$ to the negative gradients
    Compute the step size γ_m by minimizing the loss function
    Update the function $F_m(X) = F_{m-1}(X) + \gamma_m \cdot h_m(X)$
  end for
  return $F_m(X)$
end function

4.6 Histogram Gradient Boosting Regressor Algorithm

Workflow of the Proposed Model:

1. Initialize the histogram-based gradient boosting model.
2. Repeat the following steps for each boosting stage (100 stages in total):
   (a) Compute the negative gradients (residuals) of the loss function with respect to the current predictions.

(b) Construct histograms for each feature in the training data, partitioning the feature space into discrete bins.

(c) For each feature, find the optimal splits to minimize the loss function within each bin, considering the negative gradients.

(d) Build a tree structure using the best splits for each feature, where the nodes represent the bins and the leaves represent the predictions.

(e) Update the model by adding the predictions of the current tree to the previous predictions, weighted by the learning rate.

3. Once all the boosting stages are completed, the model is ready for prediction.

4. To make a prediction, provide an input feature vector to the histogram-based gradient boosting model.

5. The feature values are binned based on the histograms created during training, and the model navigates the tree structure to find the corresponding prediction.

6. The final prediction is obtained by summing the predictions of all the trees, weighted by the learning rate.

7. The trained histogram-based gradient boosting model can be used to make predictions on new unseen data by following the same binning and tree traversal process as during training.

Algorithm 4.6 Pseudocode for Histogram Gradient Boosting Regressor:

Algorithm 4.6 Histogram Gradient Boosting Regressor

Require: X: Input features, Y: Output labels, max_iter: Maximum number of iterations (boosting stages) to perform, max_depth: Maximum depth of each tree, $min\_samples\_leaf$: Minimum number of samples required to be at a leaf node

Ensure: $F_m(X)$: Trained histogram-based gradient boosting model

1: **function** HIST_GRADIENT_BOOSTING_REGRESSOR(X, Y, max_iter, max_depth, $min\_samples\_leaf$)

2: Initialize F_0 as a constant value

3: **for** m in range(max_iter) **do**

4: Compute the negative gradient r_{im} for each sample i using the loss function

5: Train a histogram-based gradient boosting tree $T_m(X)$ to the negative gradients

6: Compute the step size γ_m by minimizing the loss function

7: Update the function $F_m(X) = F_{m-1}(X) + \gamma_m \cdot T_m(X)$

8: **end for**

9: **return** $F_m(X)$

10: **end function**

CHAPTER 5

RESULTS AND PERFORMANCE ANALYSIS

The complexity analysis provides insights into the efficiency of the algorithms used in our recommendation system in terms of time and space requirements.

5.1 Time Complexity

The time complexity of the recommendation algorithms utilized in our project can be approximated as $O(n \log n)$, where n is the number of available items (courses or job skillsets). The primary factor contributing to this complexity is the sorting operation performed on the available items to determine the most relevant recommendations. The sorting operation typically has an average time complexity of $O(n \log n)$ when using efficient sorting algorithms. Thus, as the number of available items increases, the time required to generate recommendations scales logarithmically.

5.2 Space Complexity

In terms of space complexity, the recommendation algorithms utilized in our project have a space complexity of $O(1)$. The space requirements remain constant irrespective of the input size, as the algorithms mainly utilize additional space for storing variables and data structures. This efficient space utilization ensures that the memory consumption does not grow with the size of the available items, making the algorithms efficient in terms of space usage.

5.3 Evaluation of Recommendation System

The recommendation system used in this project demonstrates efficiency, adaptability, accuracy, scalability, and optimal resource usage. By intelligently recommending courses and job skillsets based on factors such as user preferences and similarity measures, the system ensures an optimized learning and career development experience for users. Its adaptability allows for customization to specific user contexts, while its accuracy matches recommendations to users' interests and career goals. With a time complexity of $O(n \log n)$ and a space complexity of $O(1)$, the recommendation system handles a large number of items efficiently and utilizes memory resources optimally. Overall, the recommendation system offers a robust solution for personalized course and job skillset recommendations, enhancing the learning and career development journey for users.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, the development and implementation of the recommendation system integrated into a chatbot application represent a significant step towards addressing the challenges individuals face in navigating the rapidly evolving job market. By leveraging item-based recommendation techniques, particularly cosine similarity measures, the system effectively assists users in identifying relevant courses and acquiring the necessary skills to meet the demands of various industries.

The recommendation system analyzes a vast dataset of courses and job requirements, employing similarity measures to identify relevant courses and skillsets that closely align with a user's interests and career goals. The integration of the recommendation system into a chatbot application further enhances user experience by providing a conversational interface for interaction, allowing users to receive personalized recommendations and explore career development opportunities in a seamless and intuitive manner.

Overall, the recommendation system serves as a valuable tool for individuals seeking to enhance their skillsets and advance their careers in an increasingly competitive job market. By facilitating access to relevant learning resources and maximizing users' potential for professional growth and success, the system contributes to the ongoing efforts to bridge the skills gap and promote lifelong learning in the digital age.

6.2 FUTURE WORK

Moving forward, there are several avenues for enhancing the recommendation system's efficacy and user experience. Firstly, continuous refinement of the recommendation algorithms is paramount. Incorporating advanced machine learning techniques, such as deep learning models, and fine-tuning parameters based on user feedback can improve the accuracy and relevance of course suggestions. Moreover, exploring novel approaches like collaborative filtering and hybrid recommendation systems could provide additional insights into users' preferences and needs.

Expanding the dataset used for analysis represents another critical area for future development. Integrating data from diverse sources, including job market trends, emerging technologies, and user behavior patterns, can enrich the recommendation system's knowledge base. This broader dataset can offer users a more comprehensive range of course options tailored to their career aspirations and industry demands.

Additionally, enhancing the conversational capabilities of the chatbot interface is essential to foster greater user engagement and satisfaction. Leveraging advancements in natural language processing (NLP) and sentiment analysis can enable the chatbot to understand user queries more effectively and provide more contextually relevant responses. Furthermore, incorporating interactive features such as voice recognition and personalized chat experiences can further elevate the user experience and encourage continued interaction with the system.

REFERENCES

[1] Y. Wang, L. Zhang, and Q. Liu. Deep learning-based recommendation systems: A comprehensive survey. *ACM Computing Surveys*, 56(6):1–39, 2023.

[2] M. Liu, H. Chen, and J. Li. An adaptive learning path recommendation method based on machine learning. *International Journal of Distance Education Technologies*, 22(3):22–38, 2024.

[3] Y. Zhang, J. Wang, and X. Liu. Hybrid job skill recommendation model based on deep learning. *Information Technology and Management*, 25(4):233–246, 2023.

[4] Q. Li, Y. Huang, and W. Chen. Intelligent chatbot systems: A review of recent advances and challenges. *IEEE Transactions on Human-Machine Systems*, 52(1):45–60, 2022.

[5] Z. Chen, L. Zhang, and Y. Wang. Dynamic ensemble learning for real-time recommendation systems. *Knowledge-Based Systems*, 257:107573, 2023.

[6] J. Wu, Y. Li, and X. Wang. A novel course recommendation approach integrating user behavior and semantic information. *Neural Computing and Applications*, 35(9):4215–4228, 2023.

[7] S. Yang, H. Liu, and W. Xu. A multi-source fusion approach for job skill recommendation based on graph neural networks. *Expert Systems with Applications*, 184:115552, 2024.

[8] Y. Wang, S. Liu, and W. Xu. Dialog interaction modeling for personalized chatbot systems. *Information Sciences*, 512:154–167, 2023.

[9] J. Li, L. Zhang, and Q. Wu. Federated learning-based recommendation system for privacy-preserving personalization. *IEEE Transactions on Network Science and Engineering*, 11(2):604–617, 2024.

[10] H. Zhao, Y. Liu, and M. Zhang. A context-aware course recommendation framework using deep reinforcement learning. *Expert Systems with Applications*, 230:115989, 2024.