TOWARDS AN EXPLAINABLE COURSE RECOMMENDATION SYSTEM (XCRS)

by

Muhammed Yasin Horasanlı

B.S., Computer Engineering, İstanbul Technical University, 2020

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2024

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Hüseyin Birkan Yılmaz, and my co-supervisor, Fatma Başak Aydemir, for their invaluable guidance, support, and expertise throughout this journey. The support and feedback I received have played a crucial role in shaping this thesis.

I am deeply thankful to my family for their love and to my wife, Halime Şeyda Horasanlı, for her patience and encouragement. Her unwavering support and belief in me have strengthened and inspired me.

I would also like to give special thanks to my friends, Alim Özdemir, Resul Tugay, and Durmuş Yılmaz for their support and generous giving of their time, energy, and thoughtful advice throughout this process. I also thank my colleagues and everyone who supported me along the way.

ABSTRACT

TOWARDS AN EXPLAINABLE COURSE RECOMMENDATION SYSTEM (XCRS)

The rapid expansion of online learning resources in Information Technology (IT) necessitates personalized guidance and transparent course recommendations. Learners face an overwhelming number of courses and career pathways, making it challenging to find tailored content aligned with their learning objectives. Existing recommendation systems often lack transparency and explanations for their suggestions. To address this gap, this study introduces the Explainable Course Recommendation System (XCRS)—the first to recommend both career roles and associated courses in an explainable manner. XCRS leverages large language model (LLM) embeddings from Google, OpenAI, MistralAI, VoyageAI, and Cohere to generate personalized suggestions based on users' knowledge, experiences, and learning interests. A unique dataset of online courses was collected and shared publicly to enrich the recommendation resource base. XCRS utilizes cosine similarity for precise vector matching and Retrieval-Augmented Generation (RAG) to enhance explanation quality, improving both transparency and relevance. Evaluated through both user and expert judgment studies, XCRS demonstrated high performance in transparency and efficiency, rated at 89.6% and 88.8%, respectively, as well as overall user satisfaction. Experts found recommendations well-aligned with user needs, validating the system's capacity to support informed decision-making in educational and career pathways. All resources needed to reproduce our results, such as code, datasets, and experimental setup, are openly provided.

ÖZET

AÇIKLANABİLİR BİR DERS ÖNERİ SİSTEMİNE DOĞRU

Bilgi Teknolojileri alanındaki çevrimiçi öğrenme kaynaklarının hızlı genişlemesi, kişiselleştirilmiş rehberlik ve şeffaf kurs önerilerini zorunlu kılmaktadır. Bireyler, öğrenme hedefleriyle uyumlu içerik bulmayı zorlaştıran, çok sayıda kurs ve kariyer yolu ile karşı karşıya kalmaktadır. Mevcut öneri sistemleri genellikle önerileri için şeffaflık ve açıklamalar sağlamaktan yoksundur. Bu eksikliği gidermek amacıyla, bu çalışma Açıklanabilir Kurs Öneri Sistemi'ni (XCRS) tanıtmaktadır. XCRS, hem kariyer rolleri hem de bunlara bağlı kursları açıklanabilir bir şekilde öneren ilk sistemdir. XCRS, kullanıcının bilgisi, deneyimleri ve öğrenme ilgi alanlarına dayalı olarak kişiselleştirilmiş öneriler üretmek için Google, OpenAI, MistralAI, VoyageAI ve Cohere gibi büyük dil modellerini (LLM) kullanır. Öneri kaynaklarını zenginleştirmek amacıyla çevrimiçi bir kurs veri seti toplanmış ve herkese açık olarak paylaşılmıştır. XCRS, vektör eşleştirmesi için kosinüs benzerliği yöntemini ve açıklama kalitesini artırmak için Almayla Artırılmış Üretim (RAG) kullanır, böylece hem şeffaflık hem de alaka düzeyi iyileşir. Kullanıcı ve uzman değerlendirme çalışmaları ile test edilen XCRS, şeffaflıkta ve verimlilikte sırasıyla %89.6 ve %88.8 oranında yüksek performans göstermiş, ayrıca genel kullanıcı memnuniyetini de sağlamıştır. Uzmanlar, önerilerin kullanıcı ihtiyaçlarıyla iyi uyumlu olduğunu belirlemiş ve sistemin eğitim ve kariyer yollarında bilinçli karar verme desteğini doğrulamıştır. Sonuçlarımızı ve deneysel kurulumumuzu yeniden üretmek için gerekli olan tüm kaynaklar, kod, veri setleri ve deneysel yapı dahil olmak üzere, açık bir şekilde sunulmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. BACKGROUND . . . 5
2.1. Recommendation Systems . . . 5
2.2. Explainability in AI and Recommendation Systems . . . 7
2.3. LLMs and Embeddings . . . 9
3. RELATED WORK . . . 11
3.1. LLM-Based Recommendation Systems . . . 11
3.2. Explainable Recommendation Systems . . . 12
3.3. Recommendation Systems in Education and Career Contexts . . . 15
4. METHOD . . . 21
4.1. Course Data Collection . . . 22
4.1.1. Udemy Affiliate API Integration and Web Scrapping . . . 23
4.1.2. Extracted Features . . . 23
4.1.3. Data Cleansing and Elimination . . . 24
4.2. Roadmap Data Collection . . . 26
4.2.1. Roadmap Node Classification and Numbering . . . 26
4.2.2. Data Organization in DataFrame . . . 29
4.3. Model Selection and Embedding Generation . . . 29
4.4. User Input Processing . . . 30
4.5. Vector Similarity Search . . . 32
4.5.1. Similarity Matrices . . . 36
4.5.2. Similarity Score Thresholds . . . 38

4.6. Role Recommendation . . . 39
4.6.1. Role Coverage Score . . . 40
4.6.2. Role Explanation Generation . . . 44
4.7. Course Recommendation . . . 46
4.7.1. Course Explanation Generation . . . 48
5. XCRS WEB APPLICATION . . . 50
5.1. Technology Stack . . . 50
5.2. User Interface . . . 51
5.3. Workflow Overview . . . 53
6. EVALUATION . . . 55
6.1. User Study . . . 57
6.1.1. Participants . . . 57
6.1.2. Procedure . . . 57
6.1.3. Survey Structure . . . 58
6.1.4. Participants' Demographics . . . 59
6.1.5. Survey Results . . . 61
6.1.6. Statistical Analysis of Survey Data . . . 69
6.2. Expert Judgment Study . . . 73
6.2.1. Study Design and Method . . . 74
6.2.2. Results . . . 76
6.2.3. Qualitative Feedback and Suggestions . . . 83
7. DISCUSSION . . . 87
7.1. Comparative Analysis with the Related Work . . . 91
7.2. Scope and Limitations . . . 93
7.3. Threats to Validity . . . 94
8. CONCLUSION . . . 97
8.1. Future Work and System Improvements . . . 99
REFERENCES . . . 101
APPENDIX A: USER STUDY SURVEY QUESTIONS . . . 120
APPENDIX B: EXPERT JUDGMENT STUDY SURVEY QUESTIONS . . . 122

LIST OF FIGURES

Figure 2.1. (a) Collaborative filtering and (b) content-based filtering recommendation methods. 6

Figure 4.1. Summary of the steps in the implementation process of the XCRS. 21

Figure 4.2. Structure of the extracted course data for "HTML5 and CSS3 Fundamentals". 24

Figure 4.3. Distribution of courses by subcategory under Development, IT & Software, and Office Productivity categories. 25

Figure 4.4. Roadmaps of AI & Data Scientist and Android Developer roles. . 27

Figure 4.5. Structure of the roadmap data. 28

Figure 4.6. Distribution of concept counts across different roles. 28

Figure 4.7. Embedding generation for courses, concepts, and user knowledge units using five LLMs. 30

Figure 4.8. Course - Concept - User entity relation diagram. 36

Figure 4.9. Illustration of the process for creating the similarity matrices. . . . 37

Figure 4.10. Kernel Density Estimate (KDE) plots showing the distribution of similarity scores in the Course × Concept Similarity Matrices across different LLMs. 38

Figure 4.11. Overview of the role recommendation flow. 40

Figure 4.12. Custom activation function which maps values from range [-25,100] to [0,100]. 43

Figure 4.13. Detailed process flow for the role recommendation. 44

Figure 4.14. Flow of explanation generation for the role recommendation. . . . 46

Figure 4.15. Detailed process flow for the course recommendation. 47

Figure 5.1. (a) Home page and (b) course cheat sheet of the XCRS web app. . 52

Figure 5.2. Example result page for career roles of the XCRS web app displaying Model-4 recommendations: (a) the first career role, (b) the second career role, and (c) the third career role. 53

Figure 5.3. System architecture diagram of the XCRS, illustrating key components and data flow for generating personalized recommendations and explanations. 54

Figure 6.1. Feedback-gathering process used in the evaluation studies. 56

Figure 6.2. Results of demographic questions: (a) education level, (b) employment status, and (c) job experience. 60

Figure 6.3. Distribution of participants' satisfaction across different large language models used in the study. 62

Figure 6.4. Distribution of participant satisfaction regarding the relevance of role and course recommendations provided by the system. 63

Figure 6.5. Participants' likelihood of considering a course recommended by the system. 64

Figure 6.6. Participants' ratings on the ease of understanding the explanations provided for the system's recommendations. 65

Figure 6.7. Participants' ratings on the ease of navigating and using the system to obtain necessary recommendations. 66

Figure 6.8. Participants' responses on whether the system helped them explore courses outside of their usual preferences. 67

Figure 6.9. Participants' likelihood of using the system to assist with their career planning. 68

Figure 6.10. Participants' likelihood of recommending the system to their peers. 69

Figure 6.11. Personas used in the expert judgment study: (a) computer science student, (b) graphic design graduate, and (c) computer engineering graduate. 74

Figure 6.12. Distribution of model selections by experts, showing the total number of selections for each model across all personas. 78

Figure 6.13. Experts' ratings on the relevance of role and course recommendations. 80

Figure 6.14. Experts' ratings on the accuracy of role and course recommendations. 81

Figure 6.15. Experts' ratings on the quality of explanations for the role and course recommendations. 82

Figure 6.16. Experts' ratings on the system's overall effectiveness. 83

LIST OF TABLES

Table 3.1. Overview of studies on course and career recommendation systems with a focus on explainability. 17

Table 6.1. Average ratings and percentage values for the usefulness of recommendations provided by each model. The highest scores are highlighted, and the second-highest scores are underlined. 61

Table 6.2. Kruskal-Wallis H test results for the education level dimension. . . 71

Table 6.3. Kruskal-Wallis H test results for the employment status dimension. Values highlighted in yellow indicate statistically significant differences. 71

Table 6.4. Kruskal-Wallis H test results for the job experience dimension. Values highlighted in yellow indicate statistically significant differences. 72

Table 6.5. Model preferences of the three experts for each persona, displaying the selected models for Persona A, B, and C. 77

Table 6.6. Expert ratings on evaluation dimensions. 79

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CSS	Cascading Style Sheets
DevOps	Development Operations
GPT	Generative Pre-trained Transformer
HTML	Hypertext Mark-up Language
IT	Information Technology
KDE	Kernel Density Estimate
LLM	Large Language Model
NLP	Natural Language Processing
NoSQL	Non-SQL
QA	Quality Assurance
RAG	Retrieval-Augmented Generation
REST	Representational State Transfer
SQL	Structured Query Language
SVN	Subversion
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
UX	User Experience
XAI	Explainable Artificial Intelligence
XCRS	Explainable Course Recommendation System

1. INTRODUCTION

Recommendation systems are systems designed to suggest relevant items to users based on prior data and preferences, predicting what a user might favor next [1]. Widely utilized across industries such as e-commerce, streaming services, online dating, and healthcare, these systems enhance user experiences by offering personalized suggestions. For instance, Tinder [2], a popular dating app, uses personal preferences and geolocation data to recommend potential matches. Beyond dating and shopping, recommendation systems have found applications in various domains, including education and career development, where they help users navigate complex choices regarding learning paths and career goals.

The education sector, particularly in the area of distance learning, has seen an exponential rise in demand for personalized recommendation systems. The growth of Massive Open Online Courses (MOOCs) has made it easier than ever for learners to access courses from leading universities and training programs worldwide. In 2021, the number of MOOCs reached over 19,400, serving around 220 million learners worldwide [3]. The popularity of online learning platforms like Udemy [4], edX [5], and Coursera [6] reflects the growing desire for accessible, flexible education. However, while these platforms feature recommendation engines to assist users in finding courses ("Recommended for You," "Top Picks"), they often lack transparency in their decision-making, failing to explain the reasoning behind the recommendations.

Given the surge in demand for Information Technology (IT) related skills and the increasing number of online courses available for learning these skills, there is a pressing need for a recommendation system that can not only suggest relevant courses but also explain its reasoning. The IT sector, with its rapidly evolving technologies, diverse career pathways, and numerous subfields, requires professionals to constantly adapt to new tools and concepts. For learners navigating this complex landscape, receiving targeted and explainable recommendations can play a crucial role in shaping their

educational and career trajectories. The challenges extend to both computer science students and self-taught developers. Traditional university students may find it difficult to choose a specialization or plan their career path due to the variety of emerging roles in the IT sector [7]. Similarly, self-taught learners who wish to shift their careers into software engineering often struggle to identify the right courses or career roles that match their goals and interests. With the plethora of courses available, the need for guidance that goes beyond 'black-box' recommendations becomes even more apparent.

While many existing recommendation systems have effectively integrated content-based filtering, collaborative filtering, and hybrid models to assist users in finding relevant items, they fall short of providing explainable and tailored guidance. Traditional systems may recommend courses or career paths based on user behaviors or preferences of similar users, but they often do not offer insights into why those recommendations are suitable. As a result, users may lack the understanding necessary to make informed decisions about their educational and career journeys. In the specific context of the IT sector, the complexity of subfields, evolving technologies, and diverse skill requirements necessitate a recommendation system that can provide clarity and direction with reasoning. Current recommendation engines, such as those used on MOOCs or other educational platforms, may offer lists of 'recommended' courses but typically fail to align these recommendations with individual user backgrounds, interests, and career goals. Furthermore, the absence of explanations for these recommendations can lead to user disengagement or misalignment between their career aspirations and the suggested courses.

This thesis proposes the Explainable Course Recommendation System (XCRS), aimed at providing explainable, tailored recommendations for both courses and career roles within the IT domain. The primary objective of XCRS is to analyze the user's background and preferences to understand their goals, deduce the most suitable courses and career roles, and present these recommendations in an explainable and personalized manner. By leveraging large language model (LLM) embeddings, XCRS seeks to enhance both the personalization and transparency of recommendations. The system

aims to achieve the following objectives:

- Generate career role and course recommendations aligned with users' backgrounds, skills, and interests.
- Provide clear explanations for each recommendation, aiding users in making informed decisions about their learning paths.
- Enhance the recommendation process by incorporating multiple LLMs to capture diverse perspectives and offer varied options.

To achieve these objectives, I developed the XCRS as a system that integrates LLM embeddings to generate top-k recommendations for users based on their inputted knowledge units (i.e., courses, concepts, subjects, topics, etc.). The system takes into account factors such as the user's background knowledge and areas of interest. It then provides tailored recommendations for both career roles and corresponding learning paths, accompanied by explanations that detail the reasoning behind each suggestion. The system is evaluated using a two-pronged approach:

(i) A user study involving participants from computer science and IT-related backgrounds to assess the system's performance based on effectiveness, persuasiveness, transparency, efficiency, serendipity, and satisfaction.

(ii) An expert judgment study was conducted to collect feedback from domain experts in education, AI, and career counseling, aiming to validate the accuracy, relevance, and overall effectiveness of the recommendations, as well as the quality of the explanations.

The XCRS system is designed with a user-friendly interface that enables easy navigation and provides clear explanations to enhance user trust and understanding. To support accurate and relevant recommendations, I scrapped a comprehensive dataset from a MOOC platform Udemy [4], focusing on courses within the IT sector. This ensures that the recommendations are grounded in up-to-date and practical content. Moreover, by integrating multiple LLMs, XCRS allows users to explore recommen-

dations from different perspectives, offering a unique opportunity to compare results across varied recommendation approaches. These foundational elements set the stage for the multifaceted contributions of this thesis:

(i) The development of XCRS, an explainable course and career role recommendation system tailored to the IT sector, leveraging LLM embeddings for transparent and relevant suggestions.
(ii) The evaluation of multiple LLMs to understand their effectiveness in providing diverse and high-quality recommendations.
(iii) The creation of a dataset scraped from Udemy, enabling the system to work with a rich and realistic set of courses.
(iv) A comprehensive evaluation through a user study and expert judgment study, offering insights into user experience and expert perspectives on the system's performance.

The thesis is structured as follows. Chapter 2 provides the necessary background on recommendation systems, explainability, and LLM embeddings. Chapter 3 reviews the related work in explainable recommendation systems within educational and career contexts. Chapter 4 outlines the approach behind XCRS, including data collection, processing, and model selection. Chapter 5 describes the XCRS web application and its user interface. Chapter 6 details the evaluation through user and expert studies. Chapter 7 discusses the findings, limitations, and threats to validity. Finally, Chapter 8 concludes the thesis with key findings, implications, and future work.

2. BACKGROUND

The emergence of large language models (LLMs) and their applications across various domains have revolutionized the way information is processed and utilized. In the context of recommendation systems, the integration of LLM embeddings has introduced novel ways to improve the relevance and personalization of recommendations. This chapter provides an overview of the foundational concepts, technologies, and theories that underpin the development of an explainable career role and course recommendation system using LLM embeddings. It discusses the key components, including recommendation systems, LLM embeddings, and explainability in AI, and sets the stage for understanding how these elements converge in the proposed system.

2.1. Recommendation Systems

Recommendation systems have been a popular research topic in the fields of computer science and information technology for many years. Recommender systems discover patterns in datasets to find out what different users want and need and to recommend similar items [8]. The term 'item' is used to refer to the items that the recommendation system suggests to the users [9].

There are various methods for constructing recommendation systems, such as collaborative filtering, content-based filtering, and techniques that blend both collaborative and content-based approaches [9]. Figure 2.1 shows the basic frameworks of collaborative filtering and content-based filtering. In recommender systems using collaborative filtering [10–12], recommendations rely only on the historical interactions between users and things that have been recorded with the name of user-item interactions matrix. Similarities between users are calculated to create groups called neighborhood [1]. Users receive recommendations based on the ratings from their neighborhoods. Collaborative filtering can be investigated in two categories: memory-based and model-based [13]. Alternatively, content-based filtering [14–16] depends on

the domain since it places greater focus on the evaluation of the characteristics of the elements to provide predictions [1]. The main principle of the content-based methods is to determine items that the user favors and their characteristics and then to recommend unveiled items based on characteristics [9].

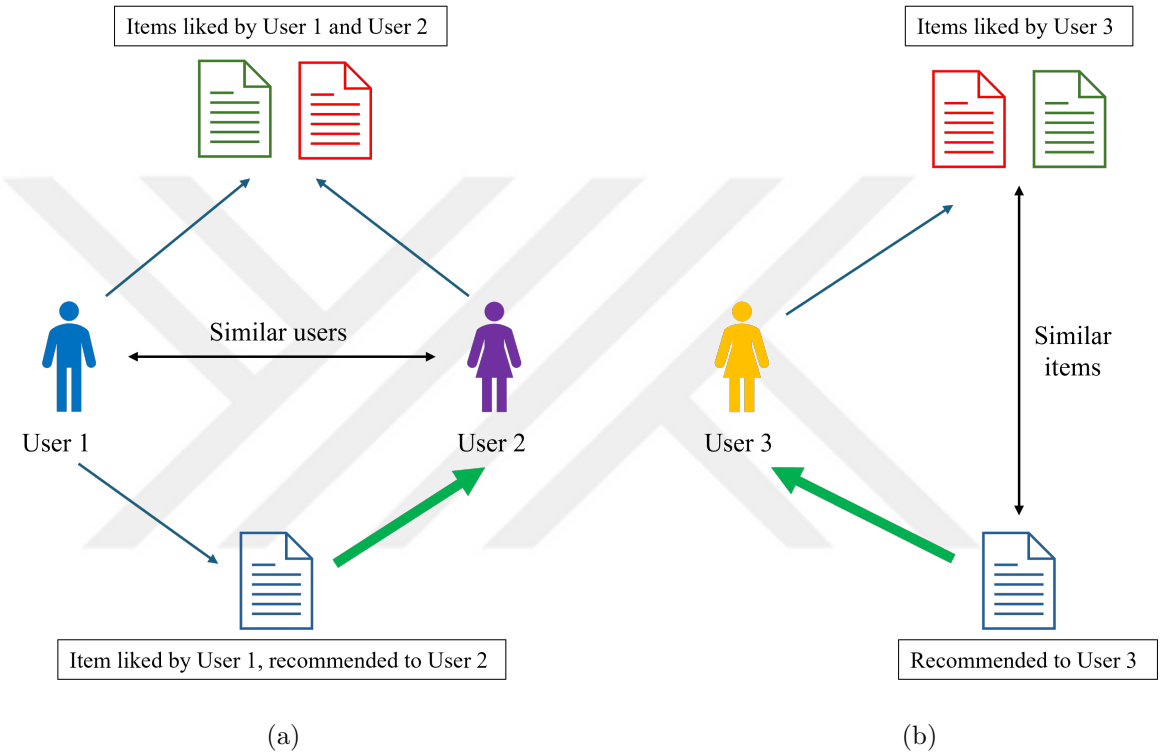


Figure 2.1. (a) Collaborative filtering and (b) content-based filtering recommendation methods.

In addition to collaborative and content-based filtering, recommendation systems encompass knowledge-based and deep learning-based approaches. Knowledge-based recommender systems make use of domain knowledge or logical inferences to match user preferences with items, offering recommendations that align closely with users' specific needs or goals [17]. Alternatively, deep learning-based models leverage their capacity to process large-scale data and identify key features through multiple hidden layers. These models are particularly effective in handling high-dimensional and sparse data, as is often the case in educational platforms like MOOCs. Deep learning not

only enables real-time data processing but also supports feature extraction and performance assessment, thereby enhancing the quality of teaching and learning [18]. Such capabilities make deep learning a powerful tool for understanding learner behavior and personalizing educational experiences within MOOC platforms. The growing adoption of MOOCs, which offer a wide variety of courses across numerous disciplines, has significantly increased the demand for personalized course recommendation systems to navigate extensive course catalogs and tailor educational pathways to individual learners. However, the opaque nature of these models raises concerns about trust and interpretability, underscoring the need for explainability in AI-driven recommendation systems.

2.2. Explainability in AI and Recommendation Systems

Systems may seem enigmatic to the users, such as black-boxes; however, they have had a significant impact on many facets of their users' lives through their outputs [19]. These complex software systems have led to a need for transparent and reliable software. Explainability is an emerging non-functional requirement that mitigates systems' lack of transparency [20]. Emerging AI models need to be explainable to disclose the output information by supplying explanations. Based on this need, Explainable AI (XAI) has become popular.

Recommendation systems that offer explanations are called explainable recommendation systems [21–23]. Explainable recommender systems try to explain to users why they would like a specific product, which helps them accept or be satisfied with suggestions and builds trust in recommender systems [24]. There are recommender models for different application areas that try to satisfy the explainability objective [24–27]. Purposes may differ for each area; however, [28] proposes seven potential goals for explanations in recommendation systems, which are transparency, scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction. According to [28], the aims of explanations in recommender systems can be summarized as follows:

- Transparency: Explains how the system works.
- Scrutability: Allows users to inform the system when it is wrong.
- Trust: Aims to increase users' confidence in the system.
- Effectiveness: Helps users make informed and good decisions.
- Persuasiveness: Convince users to try or buy recommended items.
- Efficiency: Aids users in making decisions faster.
- Satisfaction: Enhances the ease of use or enjoyment of the system.

Explainable recommendation systems go beyond traditional recommender models by not only providing suggestions but also offering comprehensible reasons behind those recommendations. The drive toward explainability in AI stems from the need for transparency and user trust, especially in domains where recommendations significantly impact decision-making, such as career planning and education. Potential goals for explanations in recommendation systems, as identified by Tintarev and Masthoff [28], serve as transitive subjects bridging user understanding and system outputs. By addressing these goals, the XCRS helps to cultivate an interactive and adaptive learning environment, empowering users with more precise insights and control in their educational journeys.

Explainability in AI and recommendation systems can be classified into two main categories: model-intrinsic and model-agnostic approaches [29]. Model-intrinsic approaches create interpretable models with transparent decision-making processes, enabling explanations to arise directly from the model's internal workings [30]. In contrast, model-agnostic approaches, also known as post-hoc explanations [31], consider the underlying model as an unobservable system and generate explanations after decisions are made [29]. In this thesis, the recommendation system follows a model-agnostic approach, where explanations are generated after the recommendations are made. This approach allows the system to offer insights into why specific courses or career roles are suggested to a user by interpreting the contributions of various user inputs. The explainability methods employed use AI-driven explanations, thereby making the system more understandable and trustworthy for end-users.

2.3. LLMs and Embeddings

LLMs have revolutionized natural language processing (NLP), offering unprecedented capabilities in understanding and generating human language. These models, exemplified by architectures like GPT [32] and BERT [33], are particularly valuable in the context of recommendation systems.

At the foundation of LLMs are embeddings, which serve as dense vector representations for words, sentences, or documents. These embeddings encode semantic relationships within a continuous vector space, positioning semantically related concepts in close proximity [34]. This property makes LLM embeddings especially powerful for recommendation systems.

In recommender systems in career and education contexts, LLM embeddings play a pivotal role by translating diverse textual inputs—such as course descriptions, job roles, or user queries—into meaningful vector representations. These embeddings then serve as the foundation for generating recommendations. By comparing the embeddings of user input with those of available career roles or courses, the system can provide personalized suggestions based on semantic similarity rather than just statistical patterns. LLM embeddings introduce a more nuanced approach to recommendations by enabling models to understand relationships between concepts at a deeper level. For example, an LLM-based career recommendation system can understand that "machine learning" and "artificial intelligence" are closely related fields. This allows it to recommend roles or courses that align with the user's interests even if exact matches are not present in the dataset. This capability is particularly valuable in the rapidly evolving job market, where new roles and skills emerge frequently.

The use of LLM embeddings is particularly beneficial for explainable recommendation systems. By providing a clear, interpretable connection between user preferences and suggested items, these models can offer insights into why certain careers or courses are being recommended. This explainability is crucial in career guidance and educa-

tional contexts, where users need to understand the rationale behind suggestions to make informed decisions about their future.

In conclusion, LLMs and their embeddings are crucial in enhancing the personalization, accuracy, and explainability of career role and course recommendation systems. By leveraging the semantic understanding embedded in these models, such systems can offer more relevant, context-aware suggestions. This contributes to the overall goal of making AI-driven career and educational recommendations both more effective and explainable, ultimately helping users make better-informed decisions about their professional and academic paths.

3. RELATED WORK

This chapter builds upon the concepts introduced in Chapter 2, providing an overview of relevant research and literature that informs the development of the XCRS. This chapter examines the intersection of large language models, explainable recommendation systems, and the application of recommendation systems in career and education contexts. Furthermore, it identifies gaps in the current body of work that this research aims to address, particularly in integrating explainability into LLM-based recommendation systems for educational and career guidance.

3.1. LLM-Based Recommendation Systems

The use of LLMs in recommendation systems has seen a significant increase, mirroring their widespread adoption in other fields. LLMs hold significant potential as recommender systems in industrial applications, fostering innovative research and experimentation in both academic and industrial settings [35]. At least three comprehensive surveys have already been conducted on the application of LLMs in recommendation systems [36–38]. These surveys highlight how LLMs are revolutionizing recommendation tasks by enhancing systems' ability to generalize, personalize, and handle complex tasks.

The publication by Wu et al. [36] serves as a comprehensive survey of how LLMs are applied to recommendation tasks. It provides an overview of the methods, applications, and model paradigms—both discriminative and generative—for integrating LLMs into recommendation systems. They divide the existing work into three categories for the adaptation of LLMs in recommendations: (1) LLM Embeddings + Recommendation Systems, where LLMs extract features for traditional models; (2) LLM Tokens + Recommendation Systems, where LLMs generate token representations for user-item interactions; (3) LLMs as Recommendation Systems, where LLMs autonomously handle the entire recommendation process.

The work by Zhao et al. [37] provides a broad analysis of how LLMs are integrated into different stages of the recommendation process. It categorizes LLM-based recommendation systems into three main approaches: (1) LLMs act as a recommender, where LLMs directly perform recommendation functions (e.g., top-k recommendations, explanation generation); (2) Bridge LLMs and Recommendation Systems, where LLMs provide data augmentation to improve traditional recommendation models; (3) LLM-based autonomous agents, where LLMs emulate user behaviors and break down complex recommendation tasks into smaller, manageable sub-tasks.

Chen et al. [38] provide an evolutionary perspective on the development of recommender systems, tracing their progression from traditional list-based recommendations to conversational agents fully powered by LLMs. This work classifies LLM-based recommendation systems into three categories: (1) LLMs for list-wise recommendations: Enhancing ranking and feature engineering in traditional recommendation systems; (2) LLMs for conversational recommendations: Leveraging LLMs to provide interactive, conversational recommendations; (3) LLM-powered recommendation agents: Fully autonomous agents capable of managing complex, multi-step recommendation processes.

Considering these surveys [36–38], the use of LLMs in recommendation systems is both widespread and analyzed under various paradigms. In this study, I utilized the combination of LLM embeddings and recommendation systems with the LLMs functioning as recommenders, aiding in generating top-k recommendations and explanations.

3.2. Explainable Recommendation Systems

The capacity of recommender systems to clearly explain their recommendations is a vital factor in enhancing both their usefulness and user experience. A well-crafted explanation should not aim to convince the user to accept a recommendation but instead empower them to assess the actual quality of the suggested item more accurately [22].

Wang et al. [39] propose a multi-task learning model for explainable recommendation by integrating user preference modeling with opinionated content modeling. The system employs joint tensor factorization to generate not only personalized recommendations but also feature-level explanations based on user-generated reviews. The model focuses on improving user trust and satisfaction by explaining how certain product features align with user preferences. Extensive experiments on Amazon and Yelp datasets demonstrate that this method outperforms traditional algorithms in both recommendation accuracy and the quality of explanations.

Chen et al. [40] introduce the Neural Attentional Regression model with Review-level Explanations (NARRE) that utilizes attention mechanisms to assess the usefulness of reviews and provide review-level explanations alongside accurate rating predictions. The model outperforms traditional approaches by selecting more relevant reviews through attention-based pooling, enhancing both recommendation quality and explainability. Experiments on Amazon and Yelp datasets demonstrate NARRE's superiority in rating prediction compared to deep learning and matrix factorization-based models. Additionally, crowd-sourced evaluations show that NARRE-selected reviews align with user-rated usefulness, making it highly effective for real-world e-commerce applications.

Wang et al. [41] introduce the Tree-enhanced Embedding Model (TEM), which combines the interpretability of tree-based models with the generalization ability of embedding-based models to create explainable recommendations. TEM leverages gradient boosting decision trees (GBDT) to extract cross features and uses an attention mechanism to assign personalized weights to these features in combination with user and item embeddings. Moreover, TEM's transparent and scrutable framework allows for user-driven adjustments to recommendations.

Wang et al. [42] present a reinforcement learning framework for explainable recommendations, which can be applied to any recommendation model (model-agnostic) while ensuring the generated explanations are personalized and high-quality. The

framework consists of two agents: one generating explanations and another evaluating their consistency and ability to predict user preferences. Extensive evaluations, including user studies, demonstrate that the framework generates more useful and sentimentally aligned explanations compared to baseline models such as NARRE [40]. This model offers improved explainability, flexibility, and control over the quality of explanations.

Xian et al. [27] propose the Policy-Guided Path Reasoning (PGPR) framework, which combines reinforcement learning and knowledge graphs to create explainable recommendation systems. The framework uses multi-hop paths in a knowledge graph to provide reasoning-based explanations for recommendations, improving transparency and decision-making. Experiments on Amazon datasets show that PGPR outperforms traditional methods in accuracy and explainability, generating interpretable paths between users and recommended items. This approach enhances both the recommendation process and the ability to explain recommendations in a causal and interpretable way.

Li et al. [43] introduce PETER, a transformer-based model designed for explainable recommendation systems that generates personalized natural language explanations. By linking user and item IDs with words through a context prediction task, PETER achieves a high level of personalization and enhances both recommendation accuracy and explanation quality. Extensive experiments show that PETER outperforms traditional models like ACMLM [44] and NETE [45] in generating diverse, feature-relevant explanations, while maintaining competitive performance in rating prediction. This approach provides a unified framework for both recommendation and explanation generation, making it a powerful tool for explainable AI in e-commerce and other domains.

3.3. Recommendation Systems in Education and Career Contexts

The application of recommendation systems in the domains of career guidance and education has garnered significant attention in recent years, driven by the need for personalized and scalable solutions to help individuals navigate increasingly complex educational and career landscapes. In educational contexts, recommendation systems are commonly used to assist students in selecting courses, majors, or extracurricular activities. Similarly, in career contexts, they can guide users in discovering suitable career roles, job opportunities, or professional development programs based on their skills, interests, and goals. Much of the earlier existing work in this area has focused on implementing content-based [46–49], collaborative filtering [50–53], and hybrid models [54, 55] to generate recommendations. However, there are also a considerable number of knowledge-based and ontology-based studies [56, 57]. Alternatively, using deep learning-based methods [58–61] has become quite common among recent studies.

Course recommendation systems have been widely researched topic, with attention given to various aspects, including goal-based [58], grade-aware [55], job market-aware [62], work plan recommender [63], learning path recommender [64], and teacher-oriented recommenders [65–67]. Some studies investigate peer learner recommender systems [68–71], thread/discussion recommender systems [72–74], question recommendation systems [75–77] in education context.

While some recommendation systems aim to recommend for university course selection and curriculum recommendation [78, 79], some aim to recommend courses in educational platforms such as Massive Open Online Courses (MOOCs) and e-learning [80–82].

In the context of career-oriented recommender systems, research has predominantly focused on job recommendation models [83–85], career path recommendation systems [86], skill recommendation systems [87], and course recommendation systems tailored to career development [88–90].

To further understand the current landscape of explainable recommendation systems in career and education contexts, several studies have been examined, focusing on their approaches to course and career role recommendations as well as the techniques employed to achieve explainability. Table 3.1 provides an overview and comparison of key characteristics across these studies, highlighting their methodologies, domains of application, and evaluation metrics. The subsequent summaries delve into each work in greater detail, elucidating how these systems contribute to the field of explainable career and course recommendation.

In the domain of software engineering education, Vo et al. [7] present a hybrid course recommendation system (CRS) tailored to the specific needs of SE students and faculty. The CRS integrates data mining and natural language processing (NLP) techniques to provide personalized recommendations for university courses, career paths, jobs, and industry-demanded skills. The system solves the cold-start problem by combining a context-aware module, which uses named entity recognition to extract technical skills from course and job descriptions, with a collaborative filtering model based on Restricted Boltzmann Machines. Evaluations conducted through a user survey of 94 participants revealed positive feedback, with 75% finding the recommendations useful for their academic and career planning.

Striebel et al. [88] present a career-based explainable course recommendation framework that leverages a knowledge graph (KG) to link jobs, courses, and skills. The system uses a random walk-based inference over the KG to generate course recommendations that align with a user's career goal, offering transparent explanations based on the skills taught in each course. Evaluations using datasets from the microelectronics industry demonstrate that the KG-based approach improves both the quality of recommendations and their explainability when compared to traditional ranking and similarity-based baselines.

Table 3.1. Overview of studies on course and career recommendation systems with a focus on explainability.

Study	Recommends	Domain	Baselines	Explanations	Technical Methods	Evaluation Methods	Statistical Test and Metrics	Code
Vo et al. (2022) [7]	University courses, careers, jobs, skills	Software Eng. Education	-	-	Data Mining, Named Entity Recognition, Restricted Boltzman Machine	User Study	Shapiro-Wilk test, Kruskal-Willis test, Friedman test, Effectiveness	-
Striebel, J., R. Myers and X. Liu (2023) [88]	Course, skill	Microelectronics Education	BM25 document ranking func. Cosine similarity	Template-based, important skills for the user-specified career goal.	Knowledge Graphs (KG)	Preliminary Evaluation, (Expert Judgment Study) Baseline Comparison	Precision, Normalized Discounted Cumulative Gain (NDCG)	Publicly available
Jiang et al. (2023) [91]	Course units in MOOCs (topics and skills)	Chinese, Mathematics, English, History, Geography, Physics, Chemistry, Biology, Physical Education, Music, Art, IT, Politics, and Others.	Item-based Collaborative Filtering (I-CF) [92] User-based Collaborative Filtering (U-CF) [10] Bayesian personalized ranking (BPR) [93] Probabilistic matrix factorization (PMF) [94] Bayesian PMF (BPMF) [95] Ekar [96]	Knowledge graph path visualization	Reinforcement Learning, Knowledge Graphs	Baseline Comparison	Hit-rate (HR), Precision, Recall, NDCG	-
Frej et al. (2024) [62]	Courses based on job market skills	IT and IT management	Exhaustive Recom. Greedy Recom. Deep Q Network (DQN) [97] Proximal Policy Optimization (PPO) [98]	Skill alignment with job market demand, sequential skill development, alignment with user goals, proficiency level of skills	Rule-based skill matching, JobBERT [99] embeddings, Unsupervised Learning, Reinforcement Learning	Baseline Comparison	Reward, Time taken	Publicly available
Abu-Rasheed et al. (2024) [100]	Learning path, courses, concepts	Multi-domain	-	Conversational multi-step explanations, knowledge graph contextualization	LLM-based chatbot, Knowledge Graphs	User Study	Precision, Recall, and F1, Satisfaction, Answer quality, Response speed, Effectiveness	-
Yang et al. (2023) [101]	Training courses for employees	Talent training and development within enterprises	TransE [102], KGCN [103], KGAT [26]	Meta-graph-based relationships, relational attention modeling, local path importance prediction	Contextualized KG Embedding	Baseline Comparison Ablation Study	Recall@N, F1@N, Hit@N, and NDCG@N, T-test	Only informed that the code would be shared.
Chau (2023) [104]	Courses based on skill requirements from job markets	Higher education	-	Skill-based explanations, Concept-based explanations, DWA-based explanations	SBERT [105], Vector Similarity Search	Online user studies	Generalized Linear Mixed Models, Paired T-test, Interestedness, Unexpectedness, Usefulness, Novelty, Serendipity	-
Zylowski et al. (2024) [106]	Career and educational pathways	Multi-domain	Form-based assistant, Intent-based chatbot	LLM-based conversational explanations	LLM-based assistant, using the Llama-2-13B-chat-GPTQ	User Study	Usability, Trustworthiness, Perceived competence and autonomy, Anthropomorphization, Explainability	-
Wang et al. (2021) [107]	Training courses for employees	Employee training	Weighted Matrix Factorization (WMF) [108] Collaborative Topic Regression (CTR) [109] Neural Matrix Factorization (NeuMF) [110] Deep Matrix Factorization (DeepMF) [111] Light Graph Convolutional Network (LightGCN) [112] Collaborative Deep Learning (CDL) [113] Collaborative Variational Autoencoder (CVAE) [114] DCBVN-O (a variant of their proposed framework)	Latent competency variables, The demand variables	Demand-aware Collaborative Bayesian Variational Network (DCBVN), Demand-aware Collaborative Competency Attentive Network (DCCAN), Autoencoding Variational Inference-based Topic Modeling, Graph-Attentive Networks	Experiments on real-world dataset	Recall@P (R@P), Precision@P (P@P), Mean Average Precision@P (MAP@P)	-

Jiang et al. [91] introduce a novel reinforcement learning-based framework for explainable knowledge concept recommendations in Massive Open Online Courses (MOOCs). The system uses a knowledge graph to capture complex relationships between entities like students, teachers, courses, and knowledge concepts. By framing the recommendation problem as a reinforcement learning task, the system generates recommendations for specific knowledge concepts, rather than entire courses. Additionally, the system provides explainable recommendations by navigating through knowledge graph paths, offering transparency and semantic reasoning for the choices. Evaluation on a real-world MOOC dataset across 14 learning fields showed that the model significantly outperformed baseline methods such as collaborative filtering and Bayesian probabilistic matrix factorization in terms of precision, recall, hit ratio (HR), and NDCG.

Frej et al. [62] propose a job-market-oriented course recommender system that aligns course recommendations with real-time skill demands from the job market. The system uses an unsupervised skill extraction and matching method (SEM) to identify relevant skills from job postings, course descriptions, and resumes. It employs reinforcement learning (RL) to recommend sequential courses that maximize the learner's job marketability by increasing the number of applicable job opportunities. Evaluations demonstrated that RL-based models, particularly Proximal Policy Optimization (PPO), provide efficient and effective recommendations for longer course sequences. The study offers a novel approach by considering both learner goals and market demands.

Abu-Rasheed et al. [100] present an LLM-powered chatbot system designed to support student learning by providing personalized learning recommendations along with conversational explanations. The system leverages a human-curated knowledge graph to contextualize and enhance the LLM's outputs, allowing the chatbot to provide relevant explanations for recommended learning paths. Additionally, the system enables group chats where students can interact with human mentors for further support. Evaluation results from a user study showed high satisfaction with the chatbot's

design and the quality of explanations provided.

Yang et al. [101] propose CKGE, a novel approach for developing an explainable talent training course recommendation system using contextualized knowledge graph embeddings. By constructing a meta-graph that captures both local semantics and high-order relationships between entities, CKGE provides personalized and motivation-aware course recommendations. The system leverages a transformer-based model, KGformer, which incorporates relational attention and structural encoding to enhance the interpretability of recommendations. Evaluation on real-world and public datasets demonstrated the system's superior performance in terms of accuracy and explainability compared to baseline methods.

Chau [104] introduces a career-oriented explainable recommendation engine that aligns college courses with career-relevant skills. The system provides two types of explanations: Concept-based, which extracts skills from course content, and DWA-based, which uses Detailed Work Activities aligned with labor market needs. Using a deep concept extraction model with BERT and BI-LSTM-CRF architectures, the engine leverages Sentence BERT embeddings and ranks skills based on cosine similarity for course recommendation. A diversification algorithm ensures varied skill explanations. Tailored to students' enrollment histories and career aspirations, the system aimed to achieve goals of interestedness, unexpectedness, usefulness, novelty, and serendipity in explainable recommendations. User studies showed that while explanations did not significantly boost course interest, they enhanced perceived relevance, particularly for unexpected recommendations. DWA-based explanations were preferred for aligning courses with career goals, and statistical analyses confirmed that explanations supported more informed decision-making.

Zylowski et al. [106] present an AI-based assistant using a large language model (LLM) for career and educational pathway recommendations, focusing on trustworthiness, usability, and explainability. The study compares the LLM-based assistant to an intent-based chatbot and a form-based assistant, each offering different levels

of interaction and recommendation structures. The LLM-based assistant uses open-ended dialogue to capture user preferences, providing tailored career and educational suggestions. A user study involving 12th-grade students revealed that while the LLM-based assistant facilitated more nuanced and personalized interactions, the quality of recommendations was variable.

Wang et al. [107] present a personalized and explainable employee training course recommendation system through a Bayesian variational approach. The proposed Demand-aware Collaborative Bayesian Variational Network (DCBVN) framework uses variational inference to model employees' current competencies and integrates collaborative filtering for personalized training recommendations. To address challenges in sparse or missing skill profiles, the Demand-aware Collaborative Competency Attentive Network (DCCAN) framework constructs competency graphs and uses graph-attentive networks to infer competencies from neighboring employees. Evaluated on a real-world dataset of learning records and skill profiles, DCBVN and DCCAN demonstrated superior recommendation accuracy over several baseline models and showed robustness in cold-start scenarios.

4. METHOD

The method integrates data collection, model selection, and natural language processing techniques. The goal is to provide personalized and transparent recommendations, ensuring that they are not only accurate but also explainable, thus empowering users with insights into why certain roles and courses are suggested. Figure 4.1 presents the steps taken to implement this work.

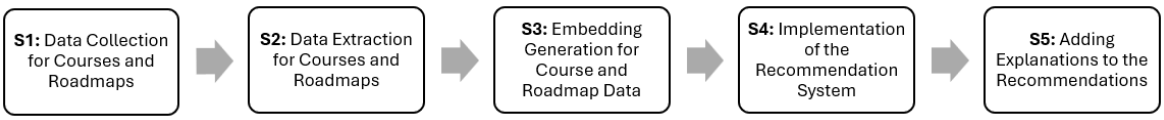


Figure 4.1. Summary of the steps in the implementation process of the XCRS.

The method begins with the collection of relevant data (S1), forming the foundation of the recommendation system. This data encompasses course-related information from Udemy [4] and roadmap structures from `roadmap.sh` [115] that represent the progression of knowledge and skills across various career paths. The system constructs a robust dataset that captures a wide range of learning resources and career trajectories by leveraging Application Programming Interface (API) integration, web scraping techniques, and rigorous data processing (S2).

Following the data collection and extraction phase, the method involves employing state-of-the-art large learning models and their embedding techniques to generate meaningful representations of both user inputs and available resources (S3). These embeddings enable the system to conduct precise similarity searches, matching users with roles and courses that align with their unique profiles.

User input processing is another critical component of the method. The system interprets the user's knowledge, preferences, and curiosities to tailor recommendations. This personalization is further refined through vector similarity searches, which identify

the most relevant roles and courses based on the user's conceptual understanding and learning goals (S4).

To enhance transparency and user trust, explainability is integrated into the recommendation process. Both role and course recommendations are accompanied by explanations that clarify the rationale behind the suggestions. These explanations are generated using advanced language models that process both the user's input and the content of the recommended resources to provide clear, contextually relevant justifications (S5).

The subsequent sections of this chapter detail each step of the method, starting with the data collection methods used to gather and refine course and roadmap data, followed by the processes employed to model, match, and explain the recommendations.

4.1. Course Data Collection

The initial step in building a robust and relevant recommendation system involves gathering a comprehensive dataset of online courses (S1). This dataset serves as the foundation upon which the entire recommendation process is constructed, and its quality is paramount to the accuracy and usefulness of the recommendations generated. The data collection process was meticulously planned and executed to ensure that the dataset not only includes a broad range of courses but also contains rich, detailed information about each course, such as its category, description, and learning objectives. Our strategy encompassed both API integration and web scraping to assemble a diverse and complete collection of course data. The subsequent sections will elaborate on the tools and techniques employed in this process, starting with our integration of Udemy's Affiliate API and the additional data enrichment performed through web scraping.

4.1.1. Udemy Affiliate API Integration and Web Scrapping

The initial phase of our data collection strategy involved leveraging Udemy's Affiliate API to access comprehensive information about public courses available on the Udemy platform. The API provided a structured source of data, including essential metadata such as title, URL, headline, whether the course was paid, and price of the course. To enrich our dataset and capture detailed information beyond the confines of the API, I employed web scraping techniques using the Selenium [116] framework. By programmatically navigating to the URLs associated with each course obtained from the API, I systematically extracted pertinent details directly from the course web pages on Udemy.

4.1.2. Extracted Features

Our approach focused on extracting specific features to construct a comprehensive representation of each course:

- Title and Headline: The course title and headline were extracted to encapsulate the primary focus and theme of each course.
- Category: The course category provided crucial contextual information, aiding in the categorization and organization of the courses.
- "What You'll Learn" Section: The "what you'll learn" section, a key component of Udemy course descriptions, was extracted to capture the learning objectives and outcomes associated with each course.
- Course Description: The complete course description was utilized as a fundamental component, offering a more detailed course content and structured narrative.

Figure 4.2 illustrates the structure of these extracted features for the example course "HTML5 and CSS3 Fundamentals".

Title	HTML5 and CSS3 Fundamentals
Headline	Build your very own website with HTML5 from scratch using HTML5 and CSS3 - designed for complete beginners
Category	Development, Web Development, CSS
What You'll Learn	Know how to use Html tags and build with the most common ones. Learn the use of attributes and common settings. Create CSS stylesheets that control your site design and set them up on your project site. Understand how the elements go together to build each part of the site Finish a complete typical website as part of your lessons.
Description	When it comes to the world of technology, staying ahead of the curve is always a challenge. In the last year one aspect of this – the world wide web – has kicked up a gear with the introduction of HTML5, the newest version of the code that makes the web tick. If you learn HTML5 along with CSS3 (the next level of web design used on all modern websites), you'll have a recipe for success; and this course will show you how. Create a website from scratch with HMTL5 and CSS3 Build a complete functioning website section by section Gain a solid knowledge base of HMTL5and CSS3 for future, more complex projects Control your website's appearance, functionality and navigability Discover how HTML5 and CSS3 can be used in other areas of the web Learn best practice techniques for building websites and web pages

Figure 4.2. Structure of the extracted course data for "HTML5 and CSS3 Fundamentals".

The extracted features were systematically concatenated to form a cohesive textual representation for each course. This composition ensured that the unique characteristics and content of each course were encapsulated within a singular, coherent text.

4.1.3. Data Cleansing and Elimination

To refine and prepare the dataset for subsequent stages of our recommendation system, the following cleansing steps were undertaken:

- Language Selection: Only English-language courses were retained, aligning with the targeted user base for our recommendation system.
- Category Selection: The scope of our recommendation system was narrowed to specific categories deemed relevant to our targeted roles and user profiles. Consequently, courses falling under the categories of Development, IT & Software, and Office Productivity were selectively retained.
- Removal of Duplicates: Duplicate courses were identified and removed to main-

tain data integrity and prevent redundancy.

- Elimination of Missing Information: Courses lacking essential information, such as titles, categories, or descriptions, were excluded from the dataset to ensure the quality and completeness of the data.

Initially, 8180 courses were scraped from Udemy. After applying the aforementioned data cleansing steps, 7727 courses were eliminated, leaving a refined dataset of 453 courses. This dataset now serves as a focused foundation for the subsequent stages of our XCRS. The elimination of duplicates, non-English courses, and the restriction to specific categories enhances the relevance and precision of our recommendation model.

The distribution of the remaining courses across various subcategories within the selected categories is illustrated in Figure 4.3, where it is important to note that these subcategories are all under the broader categories of Development, IT & Software, and Office Productivity.

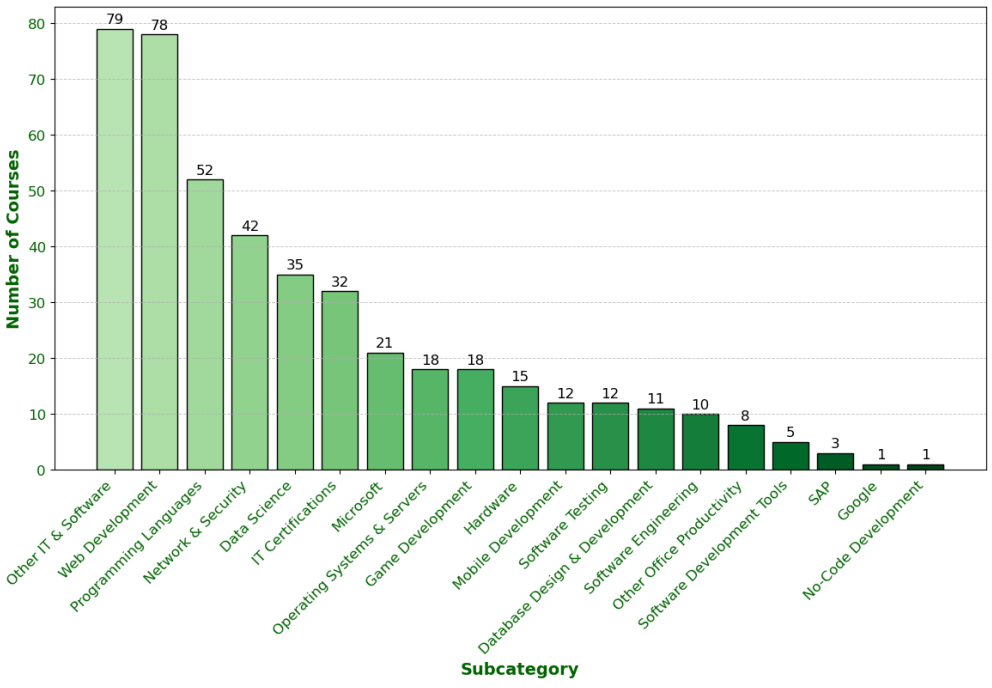


Figure 4.3. Distribution of courses by subcategory under Development, IT & Software, and Office Productivity categories.

4.2. Roadmap Data Collection

Following the acquisition of Udemy course data, our focus extended to enrich the dataset with comprehensive roadmaps essential for shaping the curriculum of our XCRS. During this phase, I engaged with Kamran Ahmad, the proprietor of the `roadmap.sh` website to secure permission to obtain roadmap data for ten distinct IT roles. Ten IT roles were strategically chosen based on their popularity and expansive scope within the IT sector. The selected roles are as follows: AI – Data Scientist, Android Developer, Backend Developer, Blockchain Developer, Frontend Developer, Full-Stack Developer, DevOps Engineer, Game Developer, QA (Quality Assurance) Engineer, UX Designer.

Upon securing permission, I systematically fetched roadmaps for the ten selected IT roles from the `roadmap.sh` website. Each roadmap delineates a structured path encompassing various nodes crucial for the successful attainment of the respective role. Figure 4.4 illustrates the roadmaps of the two roles: AI & Data Scientist and Android Developer.

4.2.1. Roadmap Node Classification and Numbering

Nodes within the roadmaps were classified into two distinct categories to facilitate a hierarchical structure:

- Concepts: Leaf nodes devoid of any sub-nodes.
- Topics: Nodes containing either leaf nodes or other nodes similar to themselves.

Nodes in the roadmap can be either a topic or a concept, and both can be directly connected to a career role. While a topic may have child topics or concepts, concepts cannot have any children, as visualized in Figure 4.5. This is the fundamental rule that distinguishes a topic from a concept.

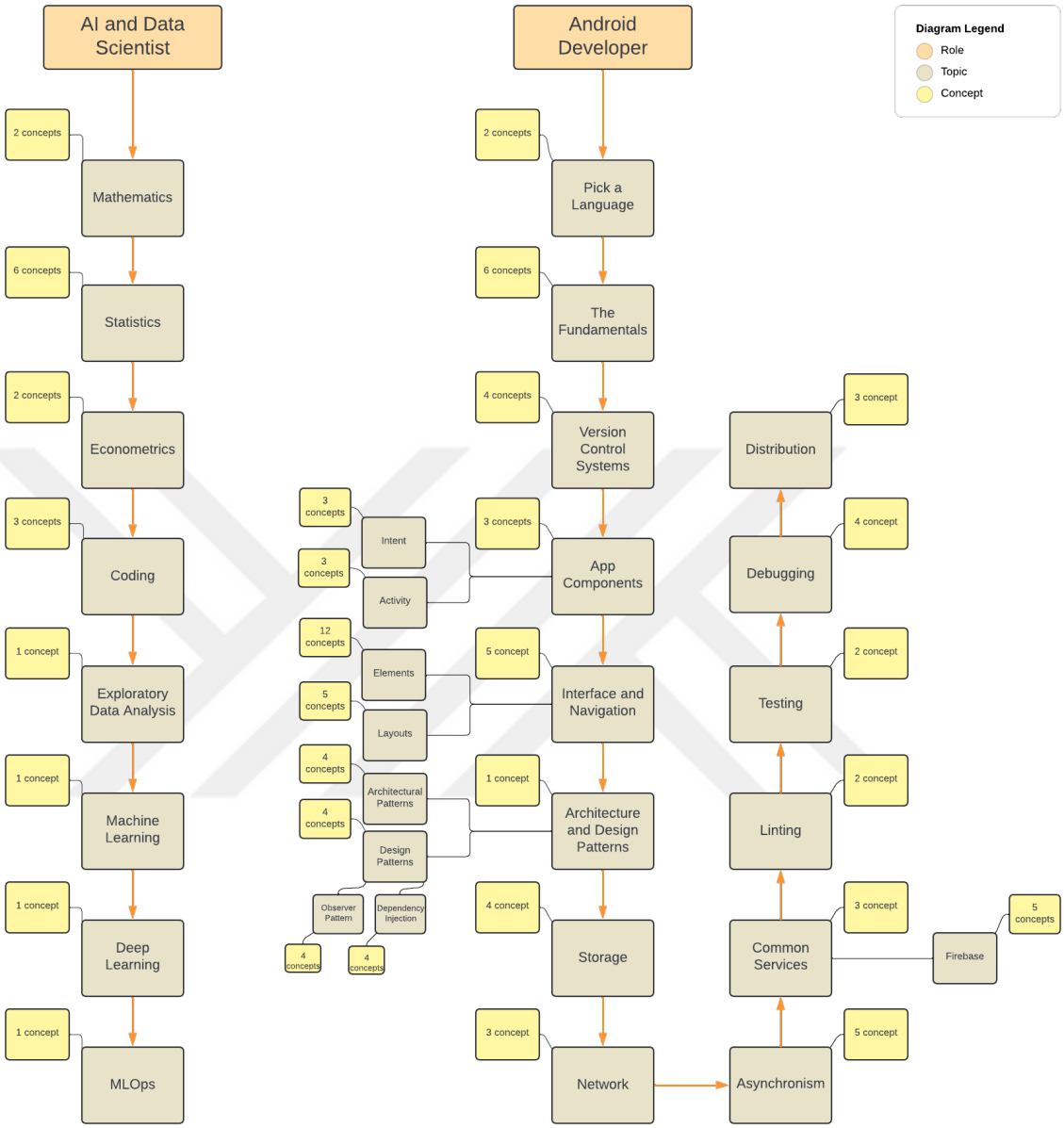


Figure 4.4. Roadmaps of AI & Data Scientist and Android Developer roles.

Furthermore, a systematic numbering scheme was introduced to encode the relationships between roles, topics, and concepts. Each role was assigned a unique number from 1 to 10, sorted alphabetically. Topics were numbered starting with the role number, followed by their order among the topics on the same level. For instance, the first topic under the role 'backend' (number 3) is numbered 301, and subsequent topics are sequentially numbered. To represent the hierarchy, child topics or concepts under each topic are assigned numbers based on their order among the children.

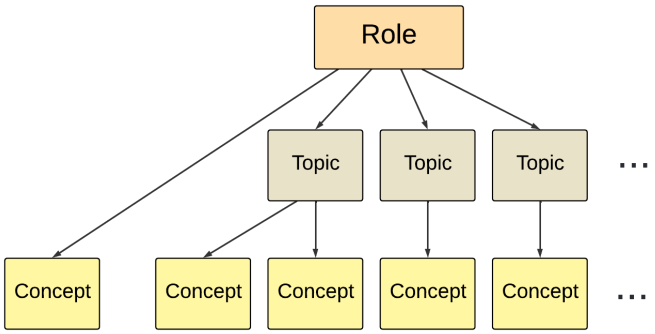


Figure 4.5. Structure of the roadmap data.

Throughout the roadmap data collection, a total of 235 topics and 869 concepts were systematically extracted, providing a comprehensive view of the learning paths associated with each IT role. Additionally, an analysis revealed the distribution of concepts under each role roadmap. The average number of concepts under each role roadmap is 97, showcasing the depth and specificity of the outlined learning trajectories. Figure 4.6 illustrates the variation in the number of concepts associated with each role, highlighting the relative complexity or breadth of knowledge required for different roles within the system.

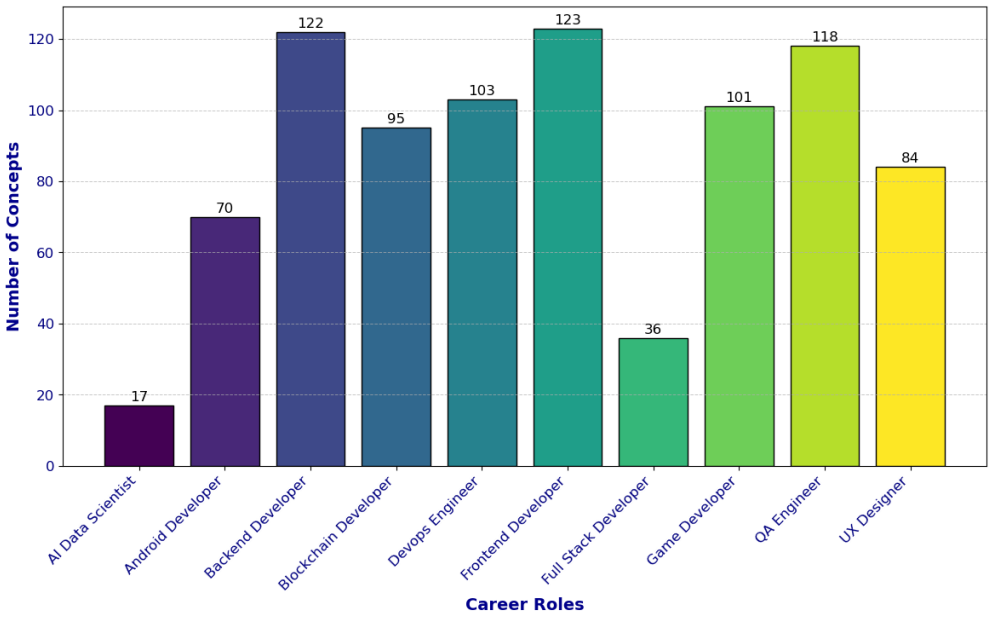


Figure 4.6. Distribution of concept counts across different roles.

4.2.2. Data Organization in DataFrame

The roadmap data was organized into a structured DataFrame with the following columns:

- Id: A unique identifier for each node.
- Name: The name or title associated with the node.
- Content: Descriptive content associated with the node.
- Type: A discriminative column classifying nodes as either 'Topic' or 'Concept.'

This meticulous organization of roadmap data in a tabular form facilitates efficient traversal and analysis, allowing us to establish relationships between roles, topics, and concepts seamlessly.

4.3. Model Selection and Embedding Generation

To facilitate accurate and relevant recommendations, our system employs five well-known LLMs to generate embeddings. The selection of these models was guided by their performance on the Massive Text Embedding Benchmark (MTEB) [117] and the reputation of the leading companies in the field of natural language processing. The latest embedding models available for API calls from the following five companies were chosen: Google [118], VoyageAI [119], OpenAI [120], MistralAI [121], and Cohere [122]. Specifically, the models utilized are *'text-embedding-004'* from Google, *'voyage-large-2'* from VoyageAI, *'text-embedding-3-large'* from OpenAI, *'mistral-embed'* from MistralAI, and *'embed-english-v3.0'* from Cohere. These models are leveraged for their state-of-the-art capabilities in generating high-quality embeddings that capture the semantic essence of textual data, which is crucial for the recommendation process.

The selected embedding model APIs were used for courses and concepts, which were scraped earlier from Udemy and `roadmap.sh`, in advance of the XCRS application, to save time on initialization. For courses, embeddings were generated from the

composed text as mentioned in Section 4.1.2, and for each concept, content fields were used. Therefore, embeddings of each course and each concept were created by five different models and kept in different DataFrames, as depicted in Figure 4.7. Lastly, every embedding was checked to ensure that none were empty and that each concept and course had corresponding embeddings from all five models.

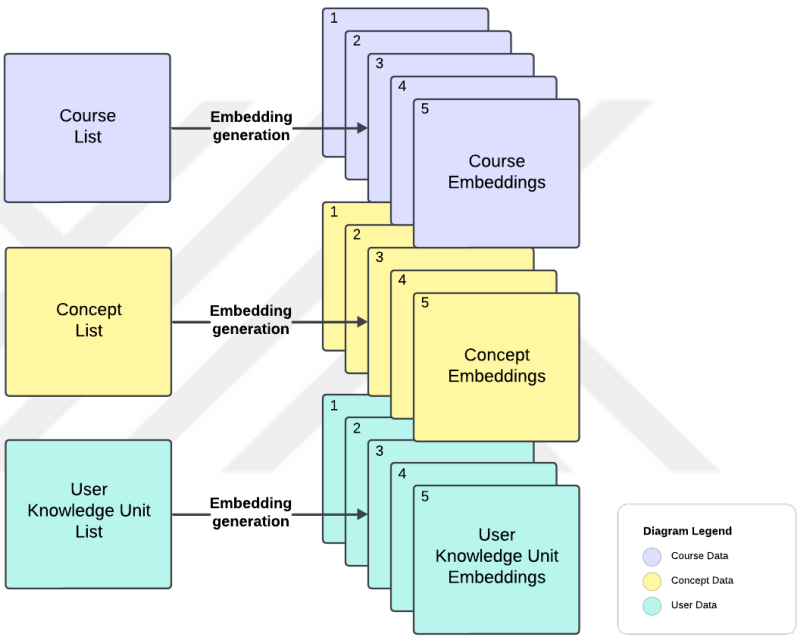


Figure 4.7. Embedding generation for courses, concepts, and user knowledge units using five LLMs.

4.4. User Input Processing

The user input processing stage is crucial for tailoring the recommendations to the user's specific preferences and experiences. Users provide their knowledge and interests as we call knowledge units. In this context, knowledge units refer to any form of educational content or learning elements, including courses, concepts, topics, subjects, technologies, or theorems. These knowledge units help frame the user's learning experiences and guide the system in making personalized recommendations based on their engagement with these elements. The user assigns these knowledge units into four

distinct categories, which allow the system to better understand the user's background and preferences:

- Courses and Concepts I Enjoyed: This category includes courses the user took, skills the user mastered, and topics, concepts, or subjects that the user learned and found enjoyable or liked. In short, it encompasses the knowledge units that the user positively engaged with.
- Courses and Concepts I Found Neutral: This category covers courses, skills, topics, concepts, or subjects that the user encountered but felt indifferent about. The user neither particularly enjoyed nor disliked these knowledge units. They represent areas where the user's interest or satisfaction was average.
- Courses and Concepts I Didn't Enjoy: This category includes courses, skills, topics, concepts, or subjects that the user learned but did not enjoy or found uninteresting. These are the knowledge units that the user had a negative experience with, either due to a lack of interest, difficulty, or other reasons.
- Courses and Concepts I'm Curious About: This category captures the user's areas of curiosity and interest for future learning. It includes courses, skills, topics, concepts, or subjects that the user has not yet explored but is eager to learn more about. This category helps identify new areas for the user to expand their knowledge.

By categorizing their knowledge and interests into these four distinct groups, users provide a comprehensive view of their educational background and preferences. This approach ensures that the recommendation system can deliver personalized and relevant suggestions that align with the user's past experiences and future aspirations.

These categories were carefully chosen to capture a wide spectrum of the user's educational experiences and preferences, serving several key purposes:

(i) Holistic Understanding: By including categories for neutral and disliked courses, the system gains a nuanced understanding of the user's experiences. This helps

avoid recommending similar courses that the user may not enjoy and focuses on areas that have a higher likelihood of being positively received.

(ii) Comprehensive Knowledge Mapping: Including skills, topics, and concepts alongside courses ensures that the system gathers a richer and more detailed picture of the user's educational background. This comprehensive approach helps the system consider all relevant knowledge areas, broadening the range of potential recommendations.

(iii) Future Learning Interests: Including the "Curious About" category allows the system to suggest courses that align with the user's future learning goals. This proactive approach helps users expand their knowledge in areas they are interested in, rather than only reinforcing what they already know.

These categories were selected to understand not only what courses the user has taken and their level of familiarity with different topics, but also how they felt about those experiences. Whether they liked, disliked, or felt neutral about a course provides critical information for modeling each user's educational journey. Additionally, knowing what users are curious about but have not yet explored gives the system the ability to support their future learning ambitions. This holistic view of interest, curiosity, and emotional response allows the system to model the user's tendencies, preferences, and divergences effectively, resulting in a more personalized recommendation experience.

Each knowledge unit provided by the user is sent to the LLM APIs to generate embeddings. This process mirrors the one used for the scraped courses and roadmap concepts, as detailed in Section 4.3. Consequently, five different embeddings are generated for each user-provided knowledge unit as shown in Figure 4.7. These embeddings are essential for accurately matching user interests with relevant courses and concepts.

4.5. Vector Similarity Search

To generate accurate and relevant recommendations for career roles and courses, I selected vector similarity search to work with LLM embeddings. This process enables

the system to measure the similarity between various embeddings, helping to identify patterns and relationships between user-provided knowledge units and potential recommendations. Several similarity metrics are used to quantify the closeness between vectors, each offering specific advantages depending on the context of the data and application. Below are some of the most commonly used similarity metrics, along with their formulas and characteristics.

Cosine Similarity. Cosine similarity measures the cosine of the angle between two non-zero vectors, focusing on their orientation rather than magnitude [123]. The formula is

$$S_{Cosine}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \tag{4.1}$$

where $\mathbf{x} \cdot \mathbf{y}$ refers the dot product of vectors $\mathbf{x}$ and $\mathbf{y}$ while $\|\mathbf{x}\|\|\mathbf{y}\|$ denotes the product of their magnitudes. Cosine similarity values range from -1 to 1, where a score of 1 implies the vectors are in the same direction, 0 means they are orthogonal, and -1 indicates they point in opposite directions. However, in the context of information retrieval, the cosine similarity between two documents is typically constrained between 0 and 1, as term frequencies are non-negative.

Dot Product Similarity. By calculating the cosine of the angle between two vectors, dot product similarity measures their alignment without applying normalization. Its formula is

$$S_{DotProduct}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}. \tag{4.2}$$

This metric provides a raw measure of similarity that accounts for both the orientation and magnitude of the vectors. It is useful when the magnitude of the vectors plays a significant role in the comparison; otherwise, cosine similarity is a more suitable option.

Minkowski Distance and Variations. Apart from cosine similarity, the most widely used distance functions for both scalars and vectors are different variants of the Minkowski distance [124]. Its formula is

$$D_{\text{Minkowski}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}. \tag{4.3}$$

When $p = 1$, it becomes the Manhattan distance, also known as taxicab distance, or L1-norm, i.e.,

$$D_{\text{Manhattan}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{4.4}$$

and when $p = 2$, it is called the Euclidean distance or L2-norm, thus;

$$D_{\text{Euclidean}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{4.5}$$

If the square-root is expelled from the equation, it becomes L2-squared distance, i.e,

$$D_{\text{L2-squared}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} (x_i - y_i)^2. \tag{4.6}$$

The Manhattan distance determines the distance between two points by moving strictly along horizontal and vertical lines, similar to navigating a grid. In contrast, the Euclidean distance measures the shortest, direct path between two points in Euclidean space. Manhattan distance is especially relevant for tasks that involve movement along structured grids, such as in navigation or routing scenarios. While the Euclidean distance is effective when magnitude differences are important, it can be sensitive to scaling. For instance, if the magnitudes of the vectors vary significantly, the Euclidean distance may overestimate or underestimate the actual similarity. Also, another variation of Euclidean distance, L2-squared distance, is used frequently due to being computationally less expensive since it omits the square root calculation.

Jaccard Similarity. Jaccard similarity [125] is used to compare the similarity between finite sets, commonly for binary vectors. The formula is

$$S_{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.7}$$

where $|A \cap B|$ represents the intersection of sets A and B, and $|A \cup B|$ their union. Intuitively, this yields a similarity of 1 when both sets are identical, as their intersection and union would be equal in size, and a similarity of 0 for completely disjoint sets. Additionally, it's important to note that these measures are only properly defined for finite sets [124].

Hamming Distance. Hamming distance [126] is used to measure the number of differing positions between two binary vectors. The formula is

$$D_{Hamming}(A, B) = \sum_{i=1}^{n} (A_i \neq B_i). \tag{4.8}$$

The Hamming distance is commonly used to measure the difference between two bit-strings of the same length by counting how many positions have differing values [127].

Among the various similarity metrics, cosine similarity was selected for this project due to its specific advantages in handling high-dimensional, sparse data, which is common in text processing and recommendation systems. Cosine similarity inherently normalizes the vectors, making it invariant to their magnitude. This characteristic is particularly beneficial in our recommendation system, where the scale of user inputs (i.e., knowledge units) may vary widely, but the orientation (context) is what ultimately matters. Two vectors pointing in the same direction will have a similarity score of 1, regardless of their magnitude, allowing for robust comparison even when users have different knowledge levels.

Additionally, cosine similarity is computationally efficient to calculate, especially for large-scale applications requiring numerous similarity calculations. In comparison to Euclidean distance or L2-squared distance, which are sensitive to the magnitudes of vectors, cosine similarity focuses solely on the direction of the vectors, making it ideal for applications in NLP. Despite ongoing debates regarding its interpretation, cosine similarity remains the most widely utilized similarity metric in NLP research [128].

4.5.1. Similarity Matrices

With the embeddings for courses, concepts, and user input prepared, the next step involves calculating the similarity between these embeddings to facilitate accurate recommendations. This is done using cosine similarity, a metric that quantifies the similarity between two non-zero vectors based on the cosine of the angle between them. Figure 4.8 illustrates the relation between courses, concepts, user knowledge units, their embeddings, and calculated similarity matrices.

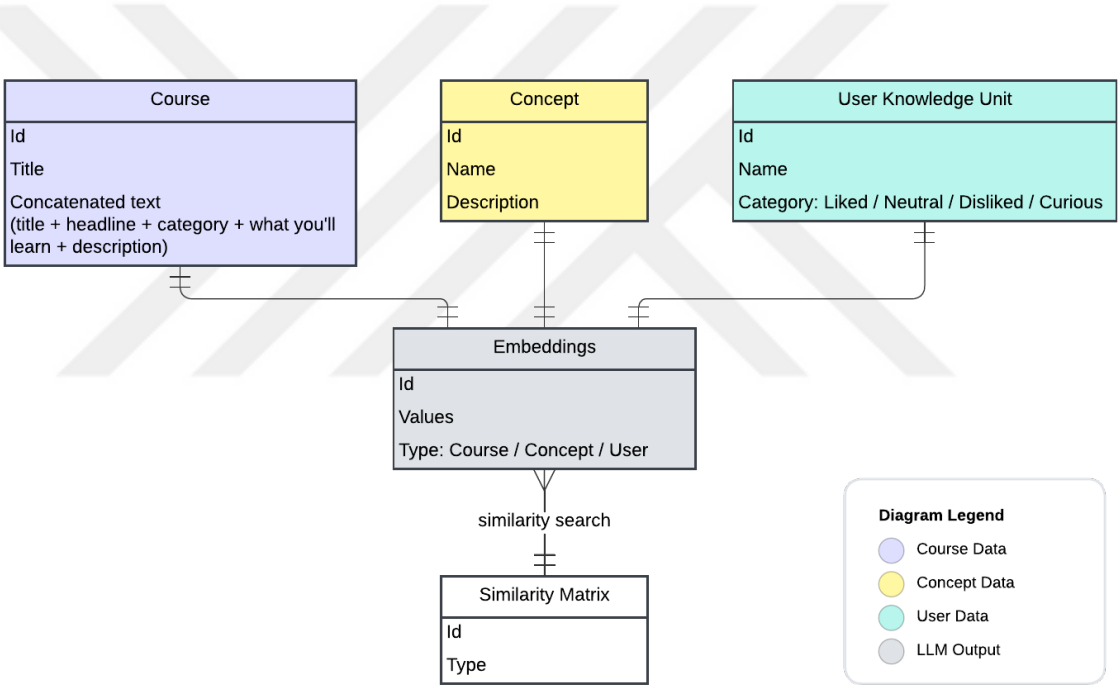


Figure 4.8. Course - Concept - User entity relation diagram.

For both the "Course × Concept" and the "User × Concept" embeddings similarity matrices, the following steps are undertaken: cosine similarity is used to construct the similarity matrices. This method measures the cosine of the angle between vectors, providing a similarity score that ranges from 0 to 1, where 1 indicates identical orientation, and 0 indicates no similarity. For each pair of embeddings (course and concept, or user and concept), the cosine similarity score is calculated, resulting in similarity matrices where each cell represents the similarity score between a specific

pair of embeddings, as illustrated in Figure 4.9. To ensure the accuracy and reliability of these matrices, multiple checks and validations are performed, including verifying the integrity of the embeddings and recalculating any discrepancies. The constructed similarity matrices are crucial components of the recommendation system. They serve as the foundation for matching courses with relevant concepts and aligning user interests with appropriate recommendations, ultimately providing personalized and relevant suggestions.

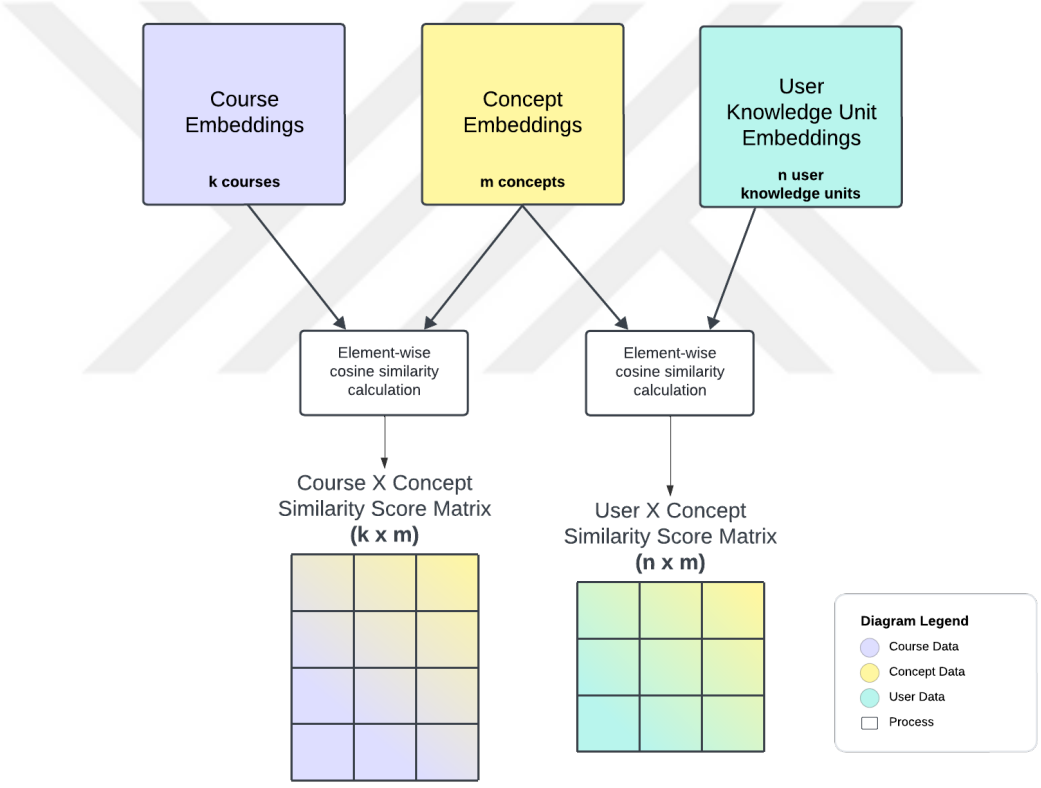


Figure 4.9. Illustration of the process for creating the similarity matrices.

However, it is important to note that similarity scores can differ for each model. This variance means that embedding similarity thresholds should vary from model to model to ensure the most accurate recommendations.

4.5.2. Similarity Score Thresholds

To address the issue of varying similarity scores across different models, I conducted an analysis to select the optimal similarity score thresholds for each LLM. I hypothesized that if the histograms of similarity score data points in the Course Embeddings × Concept Embeddings Similarity Matrices—these being the only matrices available at the beginning—follow a Gaussian distribution, I could derive a generic formulation to apply as a threshold for the similarity matrices of each LLM. Therefore, I examined the distribution of the similarity scores for each model and plotted the histograms of the data points in the similarity matrices. As anticipated, the distributions were Gaussian, allowing us to calculate the mean and standard deviation for each model. Figure 4.10 illustrates the Kernel Density Estimate (KDE) plots of the similarity scores, confirming the Gaussian nature of the distributions across the different LLMs.

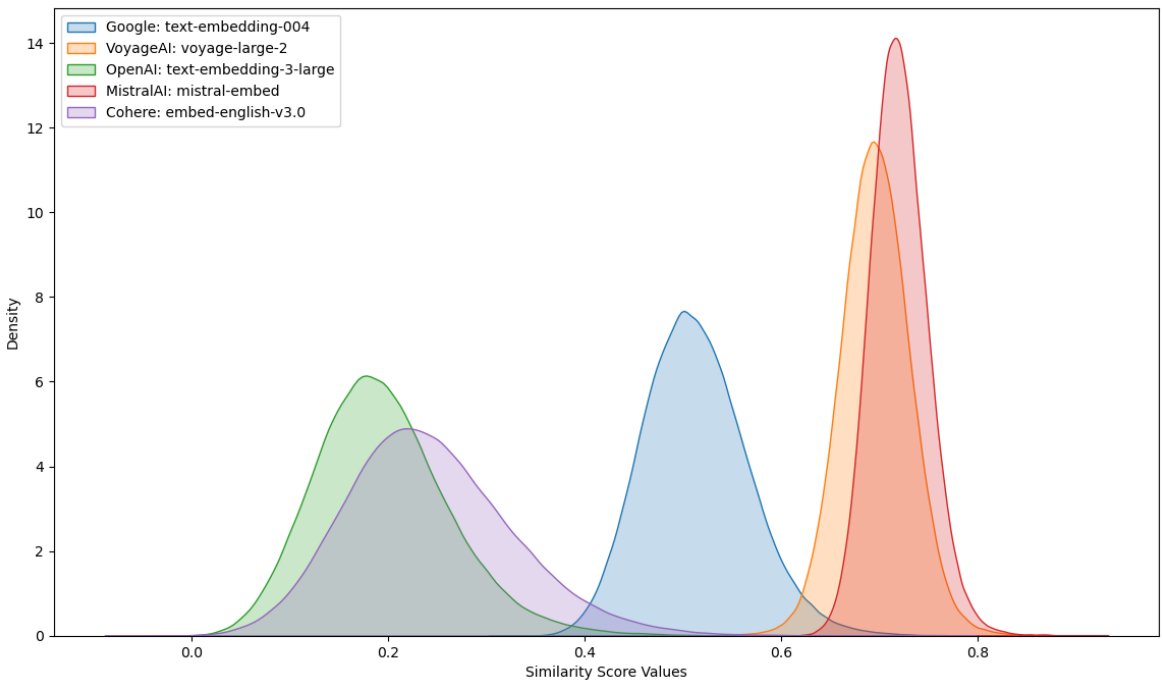


Figure 4.10. Kernel Density Estimate (KDE) plots showing the distribution of similarity scores in the Course × Concept Similarity Matrices across different LLMs.

We then tested our recommendation models using thresholds set at *mean + 2 standard deviations*, *mean + 2.5 standard deviations*, and *mean + 3 standard deviations*. By comparing the results from these different thresholds, we found that the best performance was achieved when the threshold was set at *mean + 2.5 standard deviations*. This threshold provided the most accurate and relevant recommendations, balancing the inclusion of high-quality matches with the exclusion of less relevant ones.

4.6. Role Recommendation

By utilizing the User Embeddings × Concept Embeddings Similarity Matrix, we can identify similar concepts for each knowledge unit provided by the user. These similar concepts are determined using the thresholds defined in Section 4.5.2. Additionally, the category information of the inputs—whether the user likes, is neutral about, dislikes, or is curious about a concept—is preserved with the resulting similar concepts.

Each of these categories plays a critical role in our scoring system for role recommendations, which we refer to as the coverage score. The coverage score is designed to measure the extent to which a user's knowledge aligns with the required concepts for various career roles outlined in the roadmap data. By leveraging this score, we can assess how well the user's existing knowledge and interests cover the necessary concepts for different career roles. This allows us to prioritize roles that are most aligned with the user's strengths and interests, ensuring that the recommendations are both relevant and achievable. Furthermore, it helps to eliminate roles that may not be a good fit, thereby narrowing the focus to those roles that offer the best potential match. Based on this score, and by excluding roles with a coverage score lower than 0.67, a maximum of three career roles are determined to recommend to the user as depicted in the flowchart for role recommendation in Figure 4.11.

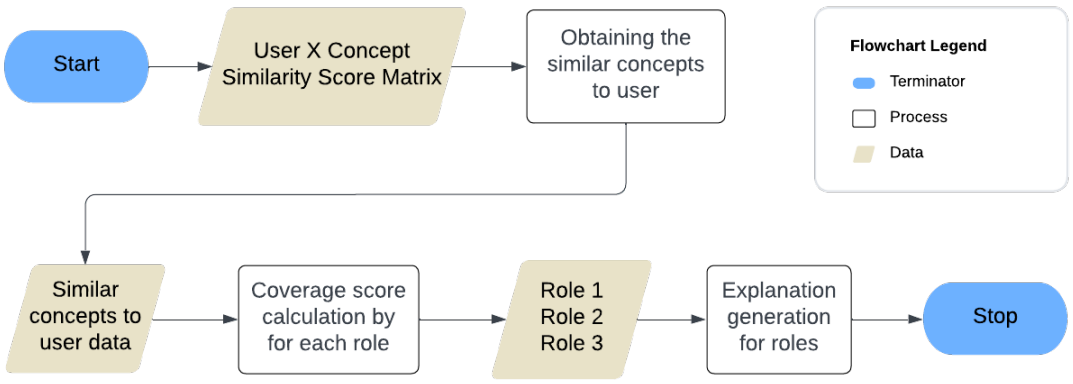


Figure 4.11. Overview of the role recommendation flow.

The following section will delve into the Role Coverage Score in detail, explaining its calculation and the rationale behind its use in our recommendation system.

4.6.1. Role Coverage Score

As its name suggests, coverage score allows us to understand how much the user covers which role and, therefore, which role they are closest to. To understand this, we first named categories as *TookAndLiked*, *TookAndNeutral*, *TookAndDisliked*, and *Curious* for the sake of convenience in expression. Each category is assigned a coefficient: 0.75 for TookAndLiked, 0.5 for TookAndNeutral, -0.25 for TookAndDisliked, and 1 for Curious. Then, for each career role, concepts that belong to it are determined. When a user has that concept under one of these categories, it affects the score of that role by the coefficient. Total scores for each career role are calculated by summing the coefficients of the associated concepts. To normalize these points, we divide the total score for each role by the total number of concepts required for that role.

The total coverage score $S(R_i)$ for a career role R_i is calculated as

$$S(R_i) = \frac{\sum_{j=1}^{n_i} \text{Coef}(C_j^{R_i})}{n_i} \tag{4.9}$$

where:

- $C_j^{R_i}$ represents each concept associated with the role R_i.
- Coef($C_j^{R_i}$) is the coefficient assigned to $C_j^{R_i}$ based on the user's interaction category.
- n_i is the total number of concepts required for the role R_i.

This formula calculates a normalized score that reflects how well the user's knowledge and interests align with the requirements of each career role. By summing the coefficients for all relevant concepts and normalizing by the total number of required concepts, the coverage score offers a clear and quantifiable measure of role alignment, guiding users toward career paths where their current knowledge and interests are most applicable.

A potential problem can arise if the user inputs the same knowledge unit into two different categories, or if two different knowledge units from different categories are found to be similar to the same concept. This creates a conflict in the scoring process. To mitigate this issue, we established a priority system for categories during score calculation. The higher-priority category will always be retained, and the lower-priority one will be disregarded. To emphasize the importance of curiosity and enjoyment in influencing recommendations, the priority order is defined as

$$Curious > TookAndLiked > TookAndDisliked > TookAndNeutral. \tag{4.10}$$

After calculating the normalized points, we multiply these points by 100. The resulting points theoretically range from -25 to 100. To better represent coverage scores on a scale from 0 to 100, we apply a custom activation function to map the values.

The core of this custom activation function is the sigmoid function, which is commonly used in neural networks. The standard sigmoid function is given by $\sigma(z) = \frac{1}{1+\exp(-z)}$, where z is the input to the function. In this custom activation function, the sigmoid function is modified by incorporating a scaling factor k and a shift b. This results in the input to the sigmoid function being transformed into $k \cdot (x - b)$. After applying the sigmoid function, the result is scaled by a factor of 100, and the

final output is rounded to two decimal places. The mathematical expression for this transformation is

$$f(x) = \text{round}\left(\frac{100}{1 + \exp(-k \cdot (x - b))}, 2\right). \tag{4.11}$$

where $k = 0.2$ and $b = 25$.

Thus, the applied custom activation function for the final coverage scores for each career role is

$$f(x) = \text{round}\left(\frac{100}{1 + \exp(-0.2 \cdot (x - 25))}, 2\right). \tag{4.12}$$

The shift parameter $b = 25$ translates the input x horizontally, meaning that the midpoint of the sigmoid function, where the output is 50, occurs at $x = 25$. The scaling factor $k = 0.2$ determines the steepness of the sigmoid curve. A smaller value of k leads to a smoother, less steep transition from 0 to 100, while a larger value would result in a steeper transition. The output of this function is constrained to values between 0 and 100, providing a normalized range for the input data. The graph in Figure 4.12 illustrates this custom activation function, showing how the output value varies with different inputs.

The mapping of input values to outputs by this function can be understood through the following scenarios: for values of x significantly less than 25, the function's output asymptotically approaches 0, corresponding to the lower bound of the sigmoid curve. Specifically, in the case where all concepts associated with a role are disliked by the user, x becomes -25, resulting in an output near the lower bound as shown in Figure 4.12 with the circle marker. If the user has not entered any knowledge unit input corresponding to the concepts of that role, x will be 0, and the coverage score of that role will be 0.67, as shown in the figure with the square marker. This is why only roles with a coverage score greater than 0.67 are considered during the recommendation

phase. At $x = 25$, the output reaches 50, marking the midpoint of the function where the sigmoid curve exhibits its inflection point, as marked in the figure with a cross marker. If all concepts in the role are known and all are found neutral by the user, x becomes 50, producing an output close to the upper bound of the sigmoid curve as marked in the figure with a star marker. When all concepts in the role are known and liked by the user, x becomes 75, as marked in the figure with a triangle marker, producing an output on the upper bound of the curve. Finally, if all concepts within a role fall under the category of "Curious" for the user, x becomes 100, as depicted in the figure with a diamond marker, causing the output to be the highest possible value. As a result, if the user has mastered all the concepts necessary to progress in that role and at least does not dislike it, it will be considered that he has almost completely covered that role.

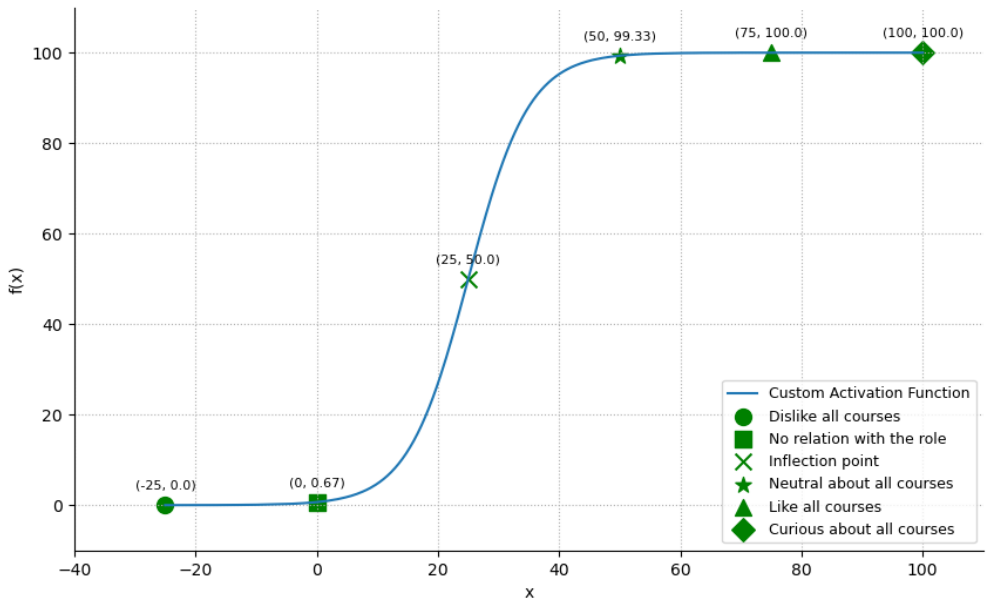


Figure 4.12. Custom activation function which maps values from range [-25,100] to [0,100].

This function ensures that the scores are appropriately scaled and fall within the desired range, providing a clear and standardized measure of how well the user's knowledge aligns with each career role.

The resulting roles obtained by role coverage score calculation are gathered and sent to the explanation generation function, which will be explained in the following section. The detailed flow, including role coverage score and custom activation function of role recommendation, is illustrated in Figure 4.13.

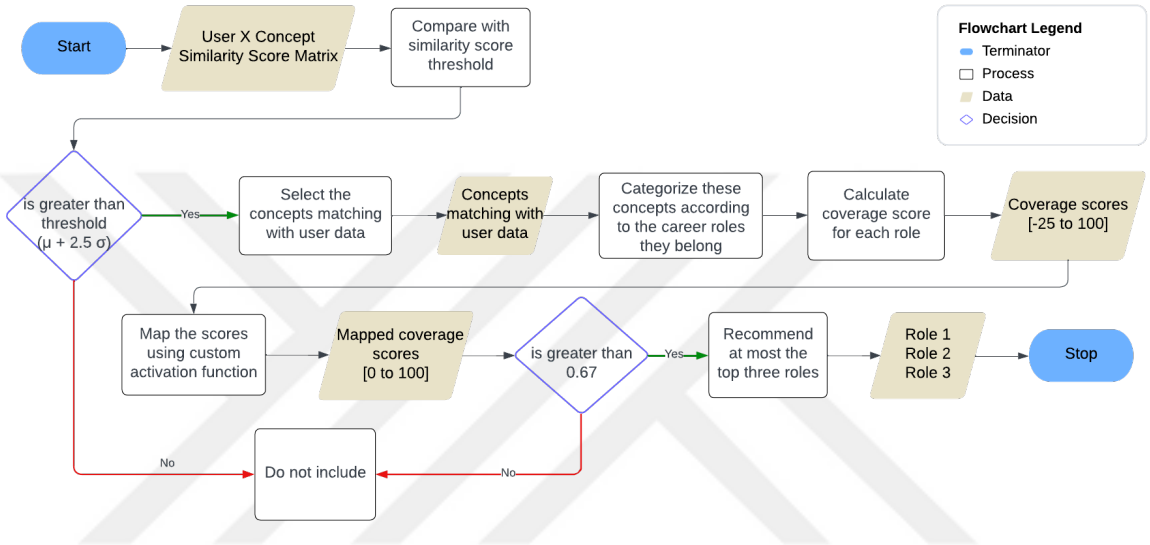


Figure 4.13. Detailed process flow for the role recommendation.

4.6.2. Role Explanation Generation

The role explanation generation in our system is designed to provide personalized explanations for the career roles recommended to the user. This function operates by leveraging the user's input, which consists of a list of concepts categorized into four distinct categories as mentioned in Section 4.6.1: *TookAndLiked*, *TookAndNeutral*, *TookAndDisliked*, and *Curious*.

For each recommended role, the system generates explanations based on the user's concept list. The explanation process considers two primary dimensions: the concepts the user is already familiar with (regardless of whether they liked, are neutral about, or disliked them) and the concepts the user is curious about. If the user's knowledge covers a broad range of topics under a specific career role, the explanation generation

emphasizes these topics. Conversely, if the user's knowledge is more specific and limited to certain concepts, the explanation focuses on these concepts instead.

The generated explanations follow a structured format. For known concepts, the explanation begins with the phrase "I assume that you are familiar with," followed by the relevant topics or concepts. For concepts the user is curious about, the explanation starts with "I can see that you are willing to learn," followed by the corresponding topics or concepts. This approach ensures that the explanations are tailored to the user's knowledge and interests, thereby enhancing the relevance of the recommended roles.

However, since the names of some concepts might be case insensitive, abbreviated, or not immediately clear, a Retrieval-Augmented Generation (RAG) [129] technique is applied to enhance the clarity and quality of the explanations. This is achieved by integrating the "GPT-4o" model from ChatGPT [130] via API, which refines the explanations through customized prompts.

The customized prompt provided to the GPT-4o model includes specific instructions to:

(i) Convert the initial explanations into standard English, correcting any unclear or meaningless parts.
(ii) Expand abbreviations related to computer science into their full forms.
(iii) Paraphrase the sentences to make them clearer, summarizing and merging key concepts where appropriate.
(iv) Ensure the sentences effectively explain why the user is well-suited for the recommended role.

The explanations for up to three recommended roles are processed in batches to optimize time and computational resources. Once the refined explanations are returned by the GPT-4o model, they are matched with the corresponding roles and passed on

to the subsequent Course Recommendation component of the system. As shown in Figure 4.14, the process for generating role explanations begins with selected roles and matched concepts with user data, followed by template-based explanation generation and enhancement through an API call to ChatGPT.

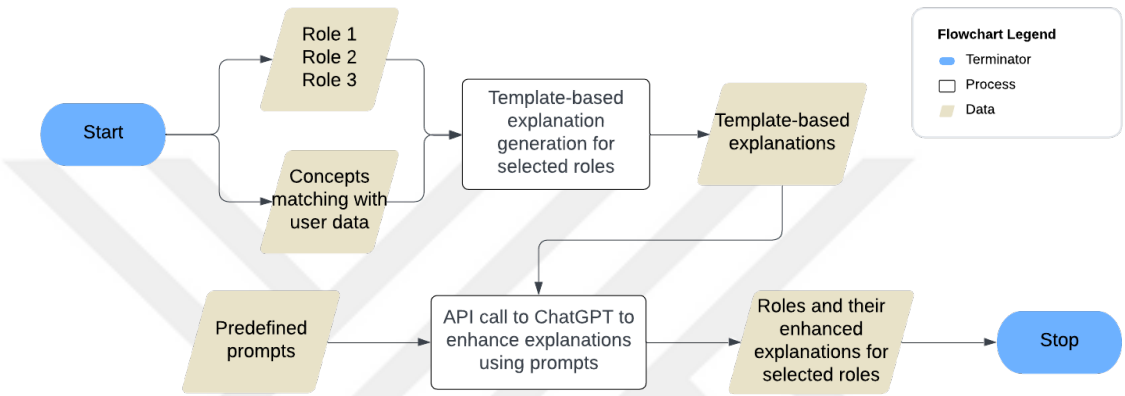


Figure 4.14. Flow of explanation generation for the role recommendation.

This approach not only enhances the transparency and comprehensibility of the role recommendations but also ensures that the explanations are contextually aligned with the user's specific knowledge and aspirations.

4.7. Course Recommendation

The Course Recommendation component is designed to identify and recommend specific courses tailored to the user's needs based on the roles recommended by the Role Recommendation component. This process begins by retrieving the user's concept data alongside the three recommended roles.

For each recommended role, the system assesses the user's familiarity with the concepts associated with that role. The user's familiarity is measured by comparing the concepts they know against the total set of concepts within the role's roadmap. If the user is familiar with more than 30% of the role's concepts, they are considered to be

at an intermediate or advanced level, rather than a beginner. In such cases, the system selects concepts located at the later stages of the role's roadmap, assuming these are areas where the user could benefit from further learning. If the user is familiar with less than 30% of the role's concepts, they are considered to be at a beginner level. In such cases, the system selects concepts located at the earlier stages of the role roadmap. In both scenarios, the system subtracts the concepts already known to the user from the total set of concepts associated with the role. The remaining, unfamiliar concepts are then selected as targets for course recommendations.

For each selected concept, the system identifies and retrieves three courses that are most similar to the concept in question by leveraging the Course × Concept similarity matrix. This matrix enables the system to pinpoint the most relevant courses. The identified courses are then aggregated for each role, with their similarity scores and selection counts (the number of times a course is recommended for a role) recorded. To maintain diversity in the recommendations, any duplicate courses are removed during this aggregation process. The detailed flow of the course recommendation is depicted in Figure 4.15.

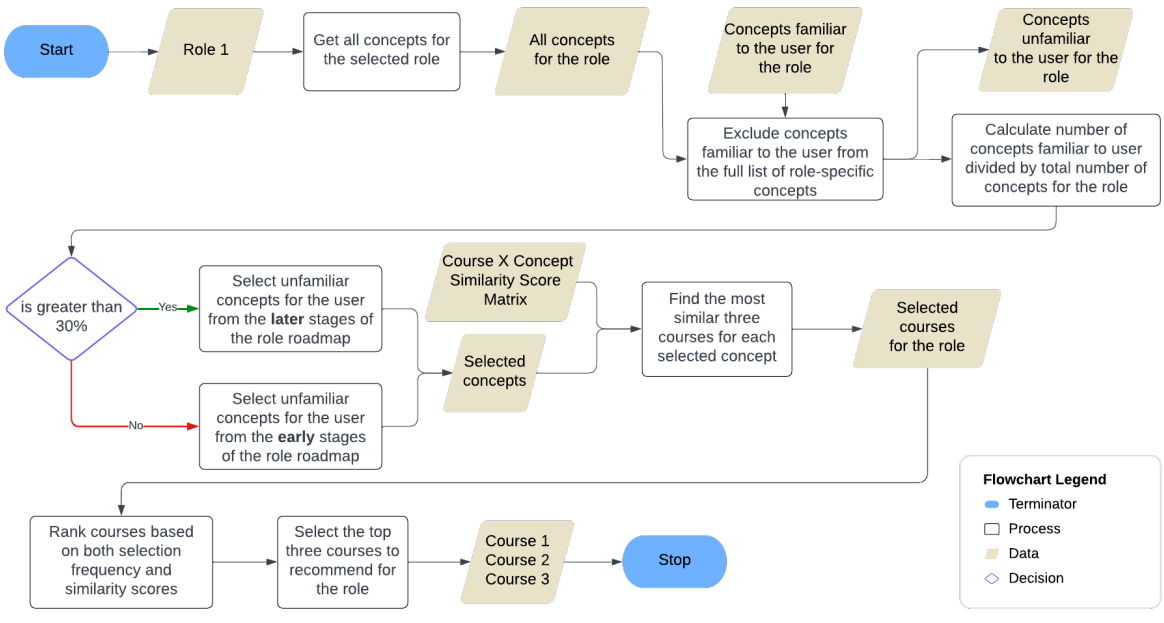


Figure 4.15. Detailed process flow for the course recommendation.

The final step involves sorting the gathered courses based on their selection count and similarity scores. The top three courses, ranked by both the frequency of selection and their similarity to the selected concepts, are determined as the final recommendations. These courses are further sorted by their similarity scores to ensure that the most relevant courses appear first. The selected courses are then passed to the Course Explanation Generation component, where detailed explanations for each course recommendation are generated. This process will be elaborated upon in the subsequent section.

4.7.1. Course Explanation Generation

The Course Explanation Generation process enhances the transparency and comprehensibility of the course recommendations provided by the system. This process begins by taking as input the set of courses matched to the recommended roles and the concepts within those roles that the user has not yet mastered. The primary objective is to generate concise, informative explanations that justify why each course is relevant and beneficial for the user's progress in the recommended role.

To achieve this, key information about each course, including the title, headline, description, and the "What You Will Learn" section, is concatenated into a single field. This field provides a comprehensive summary of the course content and learning outcomes. The system then prepares a structured prompt, which includes the concatenated course information, the list of concepts within the role that the user has not covered, and the role name. This prompt is sent via an API call to the GPT-4o model of ChatGPT. The prompt instructs ChatGPT to:

- Infer the key concepts from the provided course information.
- Explain why the user should take those courses to advance in the $<role\ name>$ role, utilizing these inferred concepts.
- Avoid including or explicitly mentioning the key concepts in the explanation.
- Where possible, connect these explanations to the concepts the user is missing:

<remaining concepts for the user>.

The explanations for the three courses are processed in batches, similar to the role explanation generation, to optimize time and computational resources. Each generated explanation is returned by ChatGPT as a brief paragraph. These explanations are then matched with the corresponding courses and assembled into a package that is subsequently passed on to the application layer. This package, containing the recommended courses and their explanations, is then presented to the user through the user interface.

Through this approach, the course recommendations become more personalized and actionable for the user. By providing clear, concise explanations that directly link the recommended courses to the user's learning gaps and career goals, the system not only guides the user toward the most relevant educational resources but also reinforces the value of each recommendation. This level of customization enhances the overall user experience, helping users make informed decisions about their learning paths and boosting their confidence in the system's recommendations.

5. XCRS WEB APPLICATION

The XCRS is a comprehensive web application that not only serves as a powerful backend recommendation engine but also provides an intuitive user interface for seamless interaction. This chapter outlines the technology stack utilized in the development of the XCRS web application, the user interface components, and the overview of the system architecture.

5.1. Technology Stack

The development of the XCRS web application leveraged a modern technology stack that ensures both scalability and a smooth user experience. On the frontend, the application was built using the Nuxt 3 [131] framework, which is based on Vue.js [132]. Nuxt 3 was selected for its advanced features, such as server-side rendering and static site generation, which contribute to enhanced performance and ease of use. Vue.js, known for its flexibility and ease of integration, was used to build the interactive components of the user interface. The use of TypeScript [133] in the frontend further improved code quality by adding strong typing and error checking, thereby reducing the likelihood of bugs and enhancing development efficiency.

On the backend, Python 3 [134] was the primary programming language, developed within an Anaconda [135] environment to manage dependencies effectively. The choice of Python was driven by its rich ecosystem of libraries, which made it ideal for handling data processing and implementing machine learning models. FastAPI [136] was chosen as the web framework due to its high performance and support for asynchronous operations, making it suitable for building the Representational State Transfer (REST) API that powers the application. To ensure that the application could handle a large number of concurrent connections, Uvicorn [137], a fast Asynchronous Server Gateway Interface (ASGI) server, was employed to run the FastAPI application. Additionally, the backend heavily relied on libraries such as NumPy [138] and

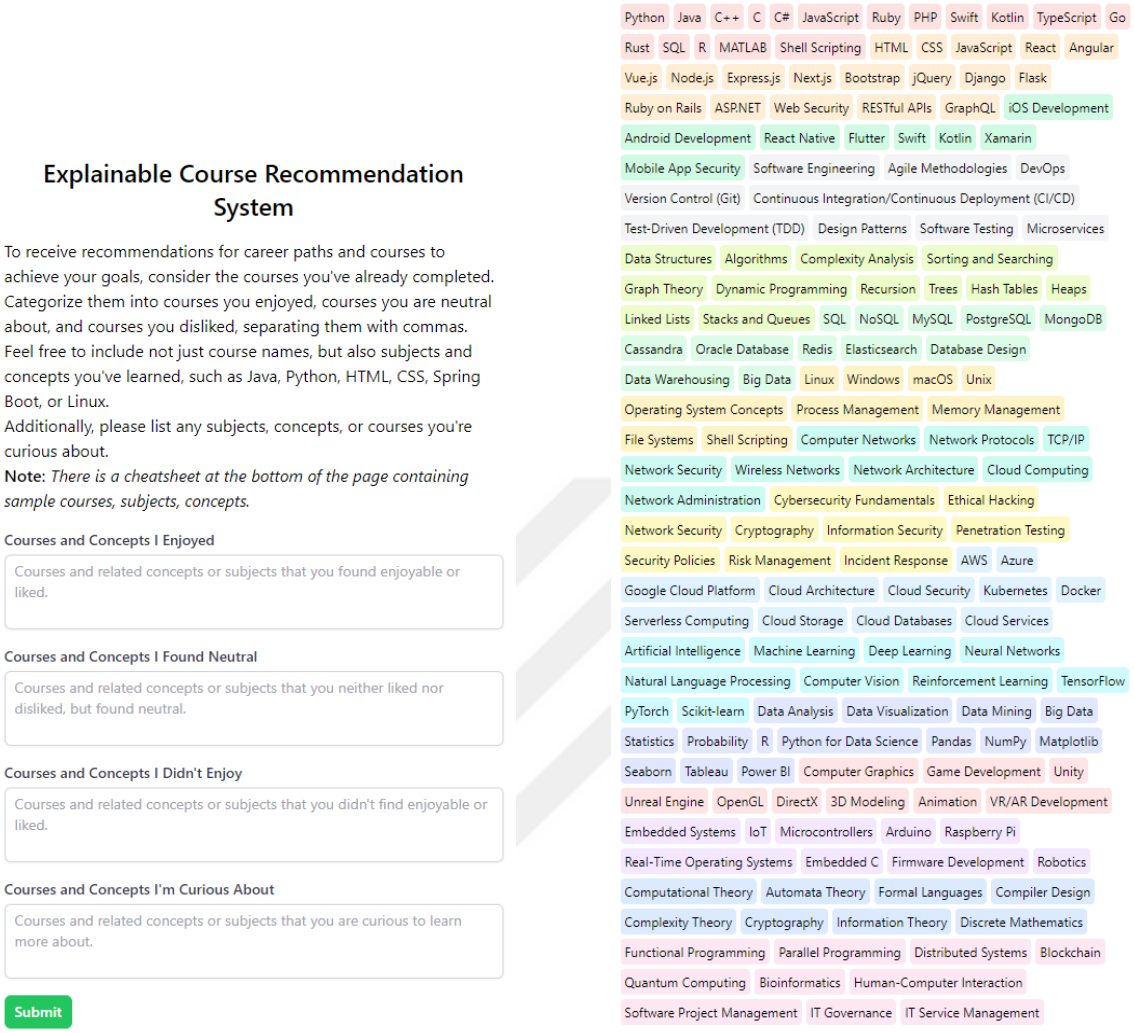
Pandas [139] for data manipulation and numerical computations. The recommendation engine within the XCRS web application integrates several APIs to create embeddings, which are crucial for generating personalized recommendations. The APIs used include Google Generative AI [118], VoyageAI [119], OpenAI [120], MistralAI [121], and Cohere [122]. Each of these APIs offers unique embedding capabilities, ensuring that the system can effectively analyze user inputs and generate relevant recommendations.

5.2. User Interface

The frontend of the XCRS web application is meticulously designed to provide a smooth and engaging user experience across various devices. The application employs a responsive design approach, ensuring that the layout and components adapt seamlessly to different screen sizes and resolutions, making it fully accessible on mobile devices as well as desktops. This responsiveness enhances usability, allowing users to interact with the system conveniently from any device.

The user interface is structured around two primary pages: the Home (Input) Page and the Result (Output) Page.

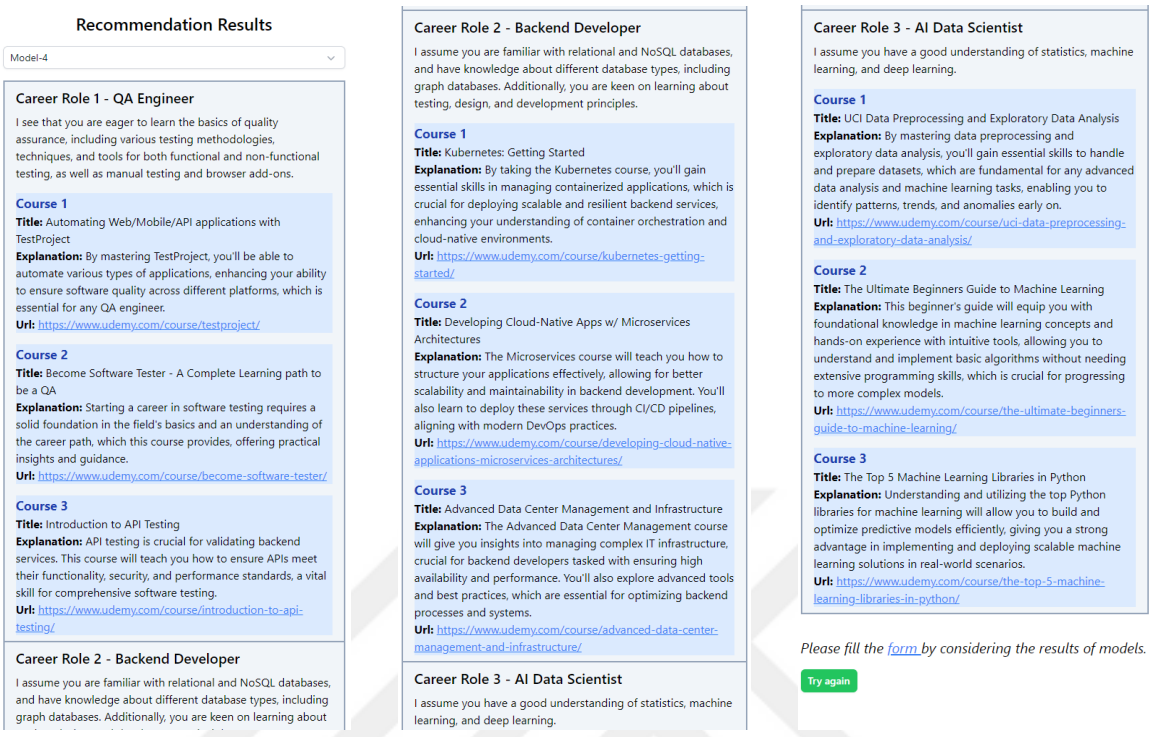
Home Page, Figure 5.1(a): This page serves as the starting point for users, where they input detailed information about the courses and concepts they have taken, liked, disliked, or are curious about. The design is clean and minimalist, focusing on ease of use and clarity. An essential feature of this page is the inclusion of a color-coded cheat sheet, as shown in Figure 5.1(a), which categorizes information according to different broad topics. This cheat sheet is visually designed to remind users of the names of courses, concepts, and knowledge units relevant to their inputs, aiding in the quick identification and selection of options.



(a) (b)

Figure 5.1. (a) Home page and (b) course cheat sheet of the XCRS web app.

Result Page, Figure 5.2: After processing the input data, the system directs users to the result page, where the recommendations are displayed. This page presents three career role recommendations along with three course recommendations for each role. Each recommendation is accompanied by a detailed explanation, helping users understand the reasoning behind the suggestions. Additionally, the result page allows users to toggle between different LLMs to explore how various models generate different recommendations. This dynamic model-switching capability offers users a comprehensive view, enabling them to compare and select the most suitable recommendations.



(a) (b) (c)

Figure 5.2. Example result page for career roles of the XCRS web app displaying Model-4 recommendations: (a) the first career role, (b) the second career role, and (c) the third career role.

The application's layout is intuitive, ensuring that users can navigate through the input and output stages effortlessly. The integration of the cheat sheet on the Home Page and the clear presentation of recommendations on the Result Page work together to provide a user-friendly and informative experience.

5.3. Workflow Overview

The operation of the XCRS web application can be summarized through a series of key steps that connect the user inputs to the final recommendations. When a user inputs information about the courses they have taken and concepts they are interested in, the frontend collects this data and sends it to the backend via an API.

Upon receiving the input, the backend processes the information using its four components: embedding generation, role recommendation, course recommendation, and explanation generation, as illustrated in Figure 5.3. The embedding generation component is responsible for making API calls to services mentioned in detail in Section 4.3 generating embeddings for the user's input data. These embeddings represent the user's inputs in a format that can be understood and processed by LLMs. The role recommendation and course recommendation components then utilize the embeddings to produce personalized career role and course recommendations.

Once the recommendations are generated, they are sent to ChatGPT for explanation generation. The explanations and recommendations are then combined into a package and sent back to the result page, where they are displayed to the user in an organized and user-friendly manner. The frontend ensures that the recommendations are easy to interpret and interact with, enabling users to effectively explore the suggested career paths and courses. The system also allows users to switch between different models to view alternative recommendations, providing a comprehensive and flexible decision-making tool.

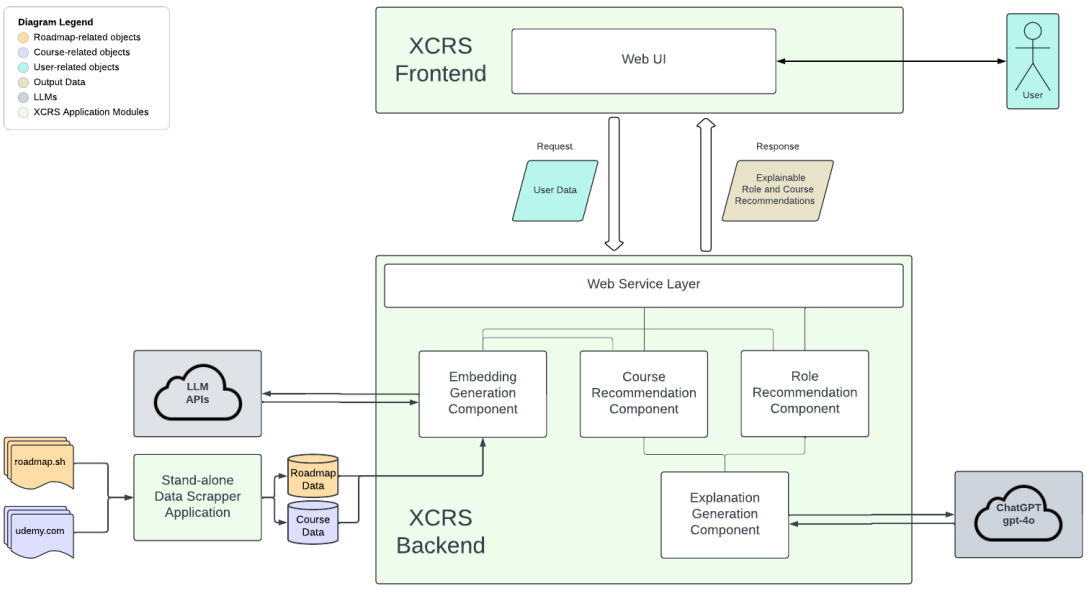


Figure 5.3. System architecture diagram of the XCRS, illustrating key components and data flow for generating personalized recommendations and explanations.

6. EVALUATION

The evaluation of the XCRS is a critical aspect of this research, aimed at assessing various dimensions of the recommendations generated by the system. This chapter details the methodologies and outcomes of the evaluation processes, which include both a user study and an expert judgment study. These complementary approaches provide a comprehensive understanding of the system's performance from multiple perspectives.

The user study focuses on the end-user experience, gauging how well the system meets the needs and expectations of its intended audience. Participants in the user study interacted with the system and provided feedback on various aspects, including the effectiveness, satisfaction, persuasiveness, transparency, and serendipity of the recommendations and explanations, as well as the overall efficiency of the system. This study is designed to capture real-world usage scenarios and identify areas where the system excels or requires improvement.

In parallel, the expert judgment study brings in domain-specific insights to evaluate the technical soundness and practical applicability of the system. Experts in the fields of education, career counseling, and artificial intelligence reviewed the system's outputs, offering a critical assessment of the recommendation logic, the quality of explanations, and the overall coherence of the system's design. Their feedback is instrumental in validating the system's approach and ensuring alignment with best practices in the respective fields. The overall feedback-gathering process is presented in Figure 6.1.

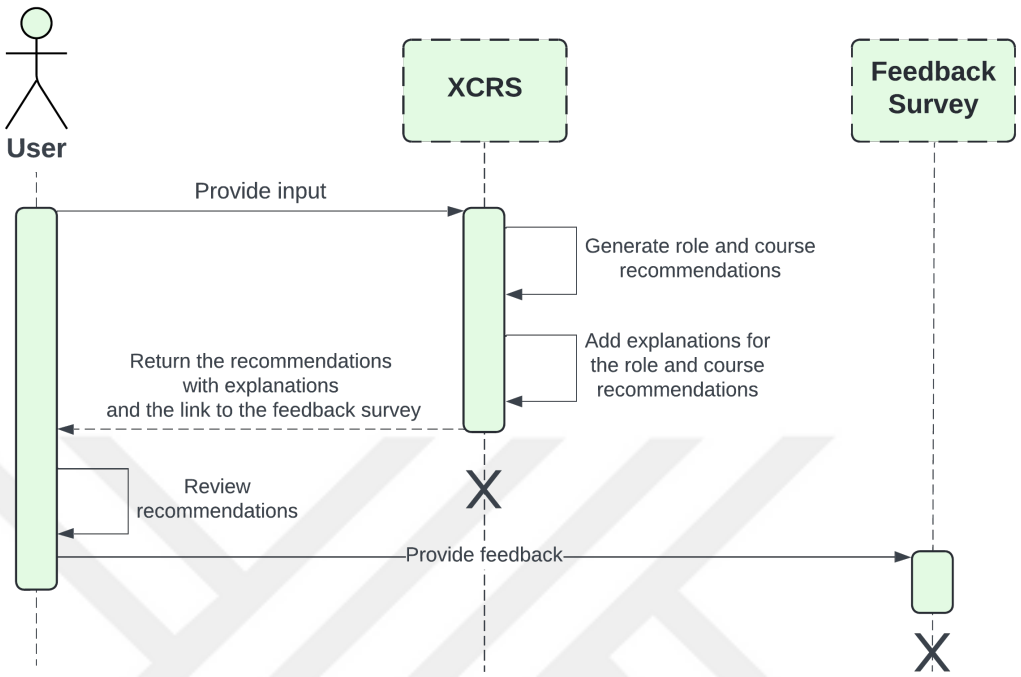


Figure 6.1. Feedback-gathering process used in the evaluation studies.

The evaluation chapter is organized into two main sections: the User Study and the Expert Judgment Study. Each section presents the method, participant demographics, data collection procedures, and key findings from the respective studies. By combining these two evaluation approaches, this chapter provides a robust analysis of the system's performance, highlighting both its strengths and potential areas for enhancement.

The insights gained from this evaluation will not only inform potential improvements to the current system but also contribute to the broader discourse on the development of explainable AI systems in the context of career guidance and education.

6.1. User Study

The User Study was conducted to evaluate the XCRS from the perspective of end-users, focusing on various aspects of the system's performance. The study aimed to assess how well the system meets the needs of its target audience, particularly in terms of the effectiveness and transparency of its recommendations, as well as the overall user experience.

6.1.1. Participants

Participants for the User Study were selected based on availability from the system's target audience, which included both students and professionals in the IT field. The study targeted individuals in computer engineering or related disciplines, such as computer science, management information systems, electrical and electronics engineering, industrial engineering, and mathematics. This approach ensured that the system's recommendations would be relevant to a broad audience within the IT sector, as the scope of XCRS is limited to courses and career roles within the computer engineering domain.

6.1.2. Procedure

The study began with participants interacting with the system by inputting their knowledge units, which included courses, concepts, subjects, and skills. Participants were given time to carefully consider all the concepts they knew and every course they had taken related to the computer engineering and computer science domains. After this reflection, they categorized these knowledge units into four distinct groups:

(i) Knowledge units they had mastered and enjoyed.
(ii) Knowledge units they had mastered and were neutral about.
(iii) Knowledge units they had mastered but did not enjoy.
(iv) Knowledge units they had not yet mastered but were curious about.

Following this input, the system processed the data and returned recommendation results based on five different models, each offering career role and course recommendations tailored to the user's input. Participants were encouraged to carefully examine the results from each model. After reviewing the recommendations, they could access a survey by clicking a form provided at the end of the page.

6.1.3. Survey Structure

The survey was designed to collect both demographic information and detailed feedback on the system's performance. The demographic section gathered data on the participants' educational background, current employment status, and job experience. This information provided context for interpreting the evaluation results (see Appendix A for the detailed survey questions).

The core of the survey focused on evaluating the system's recommendations and overall user experience across several dimensions:

(i) Usefulness of Recommendations: Participants were asked to rate the usefulness of career role and course recommendations provided by each of the five models. This section aimed to identify which model delivered the most relevant and helpful suggestions.

(ii) Model Satisfaction: Participants were asked to identify the model with which they were most satisfied, providing insights into the effectiveness of each model in meeting user expectations.

(iii) System Outputs Evaluation: The survey included several questions that assessed the system's outputs based on the model participants considered the best. These questions covered:

- Effectiveness: How satisfied participants were with the relevance of the role and course recommendations.
- Satisfaction: Participants' likelihood of using the system to assist with their career planning and recommending it to peers.

- Persuasiveness: The likelihood that participants would consider enrolling in a course recommended by the system.
- Transparency: How easy participants found the explanations provided for the recommendations.
- Efficiency: The ease of navigating and using the system to obtain the necessary recommendations.
- Serendipity: Whether the system helped participants explore courses outside of their usual preferences.

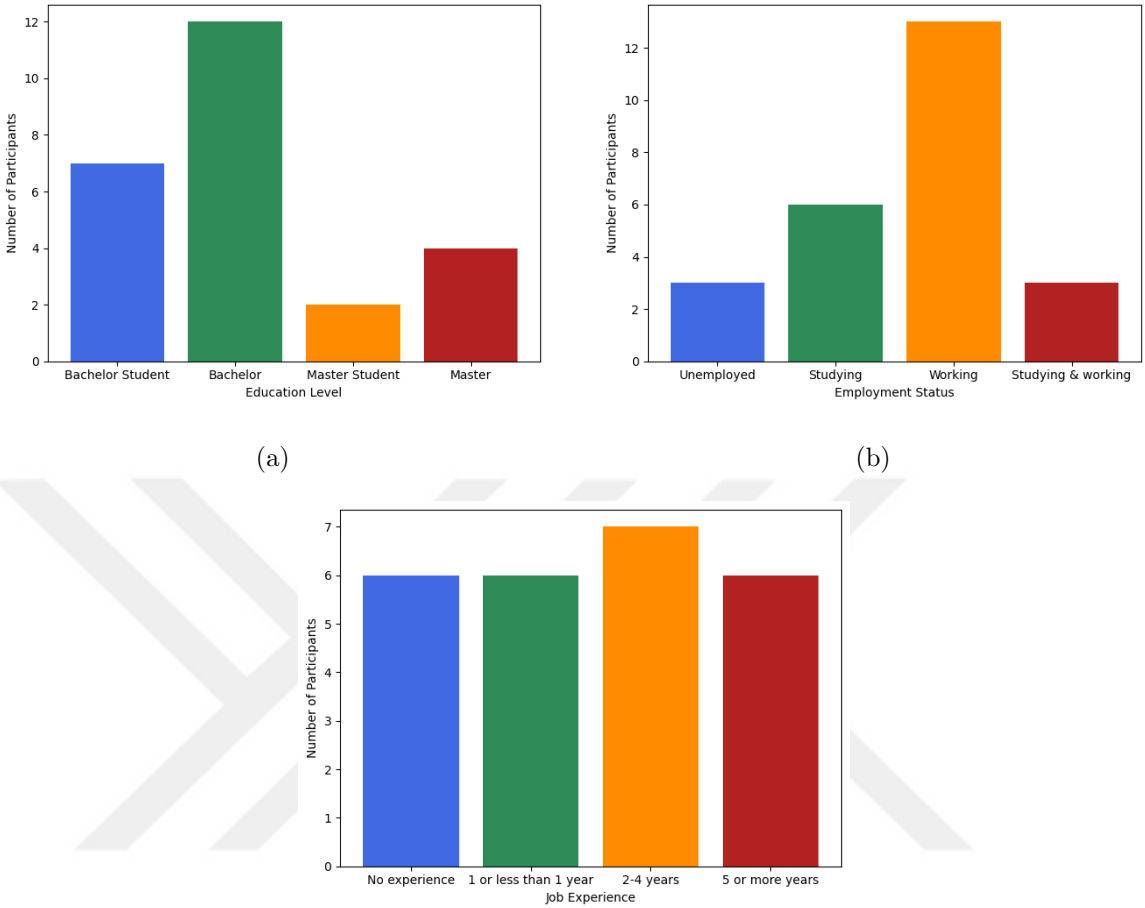
(iv) Overall Experience: Participants were invited to share any additional feedback or comments about their experience with the recommendation system, offering qualitative insights that could guide future improvements.

6.1.4. Participants' Demographics

The survey was completed by 25 participants, offering a snapshot of the demographic makeup of the study group and their feedback on the system's performance. Participants were a mix of students and professionals within the IT field, with a significant portion currently engaged in computer engineering, computer science, or closely related disciplines. Figure 6.2 provides a combined overview of the participants' education level, employment status, and job experience.

In terms of education, 12 participants held a Bachelor's degree, while another 7 were currently enrolled in Bachelor's programs. A smaller segment of the group consisted of individuals pursuing or holding Master's degrees, as shown in Figure 6.2(a).

Most participants had an academic background aligned with the system's focus, with 24 individuals confirming their studies in fields such as computer engineering, computer science, or related areas like management information systems, electrical and electronics engineering, industrial engineering, and mathematics. The remaining participant had a background in Physics, which, while not directly within the system's primary focus, still bears relevance to the technical and analytical domains.



(a) (b)

(c)

Figure 6.2. Results of demographic questions: (a) education level, (b) employment status, and (c) job experience.

Regarding employment status, 13 participants were currently working, 6 were studying, 3 were both studying and working, and 3 were unemployed, as illustrated in Figure 6.2(b). Of those employed, 16 were working within the IT sector, underscoring the relevance of the system's recommendations to their professional paths.

When considering job experience, the survey revealed a diverse range among participants. As presented in Figure 6.2(c), six participants reported having no professional experience, while another six had one year or less. Seven participants had between two and four years of experience, and six participants had five or more years of experience in the field.

6.1.5. Survey Results

Participants were asked to rate the usefulness of the recommendations provided by each of the five models, which correspond to different LLMs created by various companies. The models used were as follows: text-embedding-004 by Google, voyage-large-2 by VoyageAI, text-embedding-3-large by OpenAI, mistral-embed by MistralAI, and embed-english-v3.0 by Cohere. It is important to note that participants were not provided with any information about the specific LLMs; instead, the models were simply labeled as Model 1, Model 2, Model 3, and so forth during the survey.

These ratings, collected on a scale from 1 to 5, were converted to percentage values for better readability, as shown in Table 6.1. It is evident that participants found mistral-embed by MistralAI particularly effective for course recommendations, achieving the highest average rating of 3.72, corresponding to 74.4%. In terms of career role recommendations, voyage-large-2 by VoyageAI slightly outperformed the others with a 72.8% usefulness rating. Overall, the results suggest that while some models had slight advantages in specific areas, all models were generally perceived as useful by the participants.

Table 6.1. Average ratings and percentage values for the usefulness of recommendations provided by each model. The highest scores are highlighted, and the second-highest scores are underlined.

Model (LLM)	**Career Roles (Avg. Rating / %)**	**Courses (Avg. Rating / %)**
text-embedding-004 (Google)	3.52 / 70.4%	3.40 / 68%
voyage-large-2 (VoyageAI)	3.64 / 72.8%	3.32 / 66.4%
text-embedding-3-large (OpenAI)	3.32 / 66.4%	3.40 / 68%
mistral-embed (MistralAI)	3.60 / 72%	3.72 / 74.4%
embed-english-v3.0 (Cohere)	3.56 / 71.2%	3.52 / 70.4%

When asked which model's recommendations they were most satisfied with, the participants showed preferences that slightly diverged from the average ratings of usefulness. The most satisfaction was reported with text-embedding-004 by Google, which was selected by 8 participants as delivering the most satisfactory recommendations. This was followed by mistral-embed by MistralAI, favored by 6 participants, and embed-english-v3.0 by Cohere, chosen by 5 participants. Although voyage-large-2 by VoyageAI and text-embedding-3-large by OpenAI had fewer participants identifying them as the most satisfying, these results indicate a diversity of preferences among users, with no single model overwhelmingly dominating in both usefulness and satisfaction. Figure 6.3 shows the distribution of participants' satisfaction with each model.

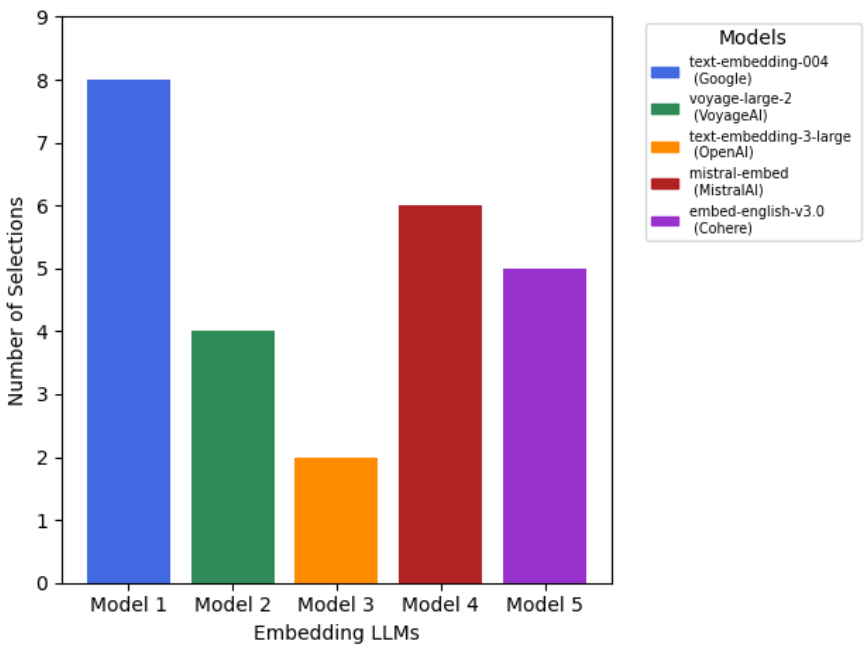


Figure 6.3. Distribution of participants' satisfaction across different large language models used in the study.

This slight divergence between the highest-rated models for usefulness and those that received the most satisfaction votes suggests that other factors, such as perceived relevance, clarity, or personal alignment with the recommendations, might play a significant role in overall user satisfaction.

In addition to assessing the usefulness of the recommendations, participants were asked to evaluate various aspects of the system's outputs, focusing on effectiveness, persuasiveness, transparency, efficiency, serendipity, and satisfaction. The responses to these questions provide deeper insights into how users perceived the overall performance and usability of the XCRS. The following sections summarize the key findings, with references to corresponding figures for visual representation.

Effectiveness. Q1) How satisfied are you with the relevance of the role and course recommendations?

Participants were asked how satisfied they were with the relevance of the role and course recommendations provided by the system. The results indicate that the majority of participants found the recommendations to be quite relevant, with an average satisfaction rating of 4 out of 5, i.e., 80%. Specifically, 10 participants rated their satisfaction at 4, and 8 participants gave the highest rating of 5 (Figure 6.4).

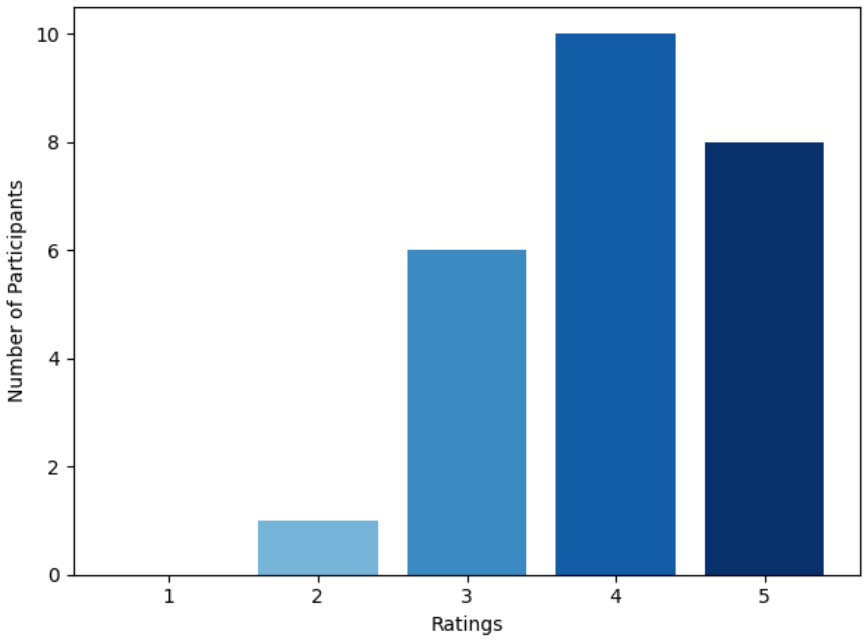


Figure 6.4. Distribution of participant satisfaction regarding the relevance of role and course recommendations provided by the system.
1: Not satisfied at all, 5: Very satisfied.

Persuasiveness. Q2) How likely are you to consider a course that was recommended to you?

When asked how likely they were to consider a course recommended by the system, participants responded with an average rating of 3.8, i.e., 76%. This suggests that the system was relatively persuasive in encouraging users to consider the suggested courses. The majority of participants rated this question at 4, with 13 people providing this rating, followed by 5 participants who rated it at 5 (Figure 6.5).

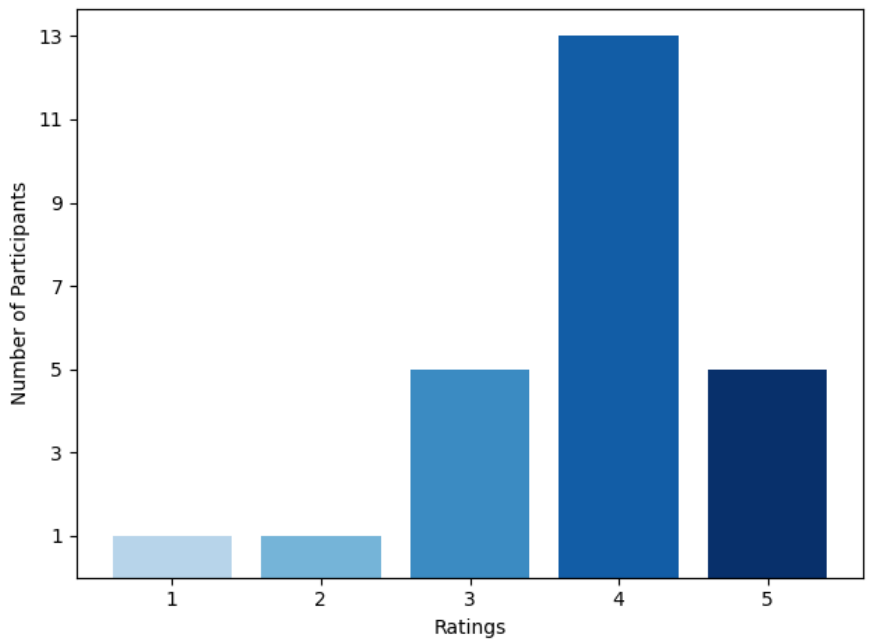


Figure 6.5. Participants' likelihood of considering a course recommended by the system.

1: Not likely at all, 5: Very likely

Transparency. Q3) How easy do you find the explanations provided for the recommendations to understand?

The system's transparency was evaluated based on how easy participants found the explanations provided for the recommendations to understand. This aspect received

one of the highest ratings, with an average of 4.48, i.e., 89.6%. All participants rated this either 4 or 5, with 13 giving a rating of 4 and 12 giving the highest rating of 5 (Figure 6.6). This strong positive response highlights the system's effectiveness in clearly communicating the rationale behind its recommendations. While this question primarily aims to measure the system's transparency, as one of the key purposes of explanations in recommender systems [28], it is essential to note that the clarity of these explanations may also influence other dimensions of the user experience.

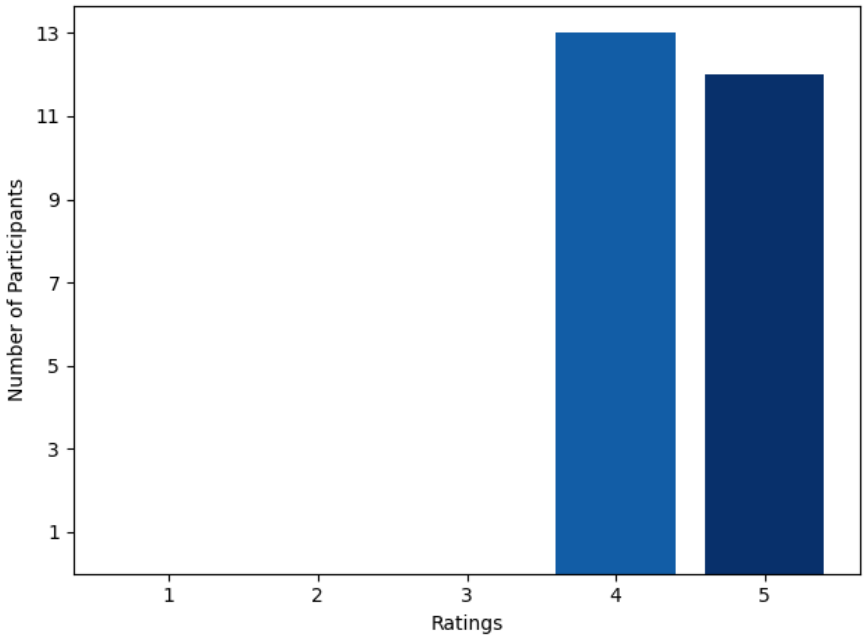


Figure 6.6. Participants' ratings on the ease of understanding the explanations provided for the system's recommendations.

1: Very difficult, 5: Very easy

Efficiency. Q4) How easy is it to navigate and use the system to get the recommendations you need?

Regarding the ease of navigating and using the system to obtain the necessary recommendations, participants gave an average rating of 4.44, i.e., 88.8%. The majority of participants found the system very easy to use, with 15 participants giving it the

highest rating of 5 (Figure 6.7). Only 2 participants rated it below 4, indicating a generally high level of usability.

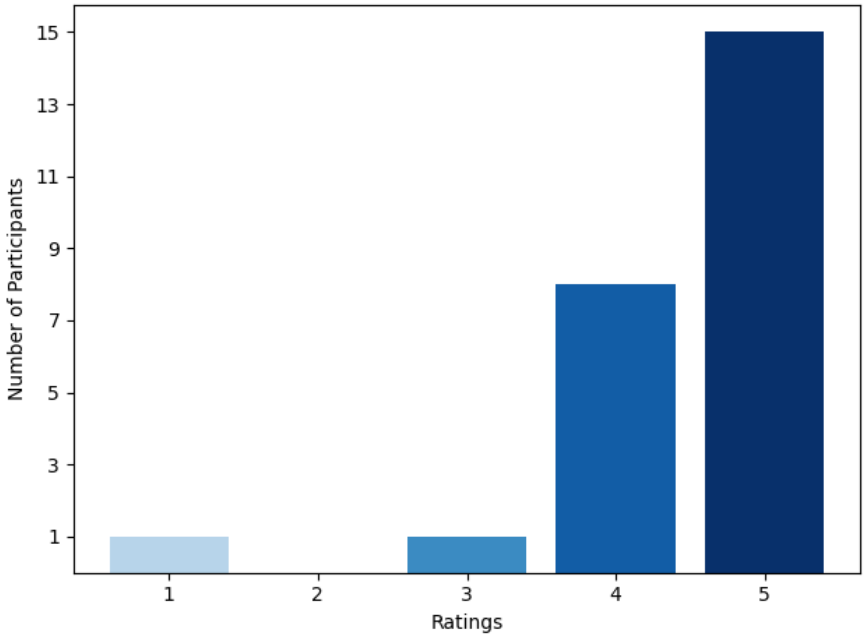


Figure 6.7. Participants' ratings on the ease of navigating and using the system to obtain necessary recommendations.

1: Very difficult, 5: Very easy

Serendipity. Q5) Do you feel that the system helps you explore courses outside of your usual preferences?

To gauge whether the system helped users explore courses outside of their usual preferences, participants were asked if they felt the system introduced them to new options. A significant 84% of participants (21 people) responded positively, indicating that the system successfully broadened their horizons (Figure 6.8).

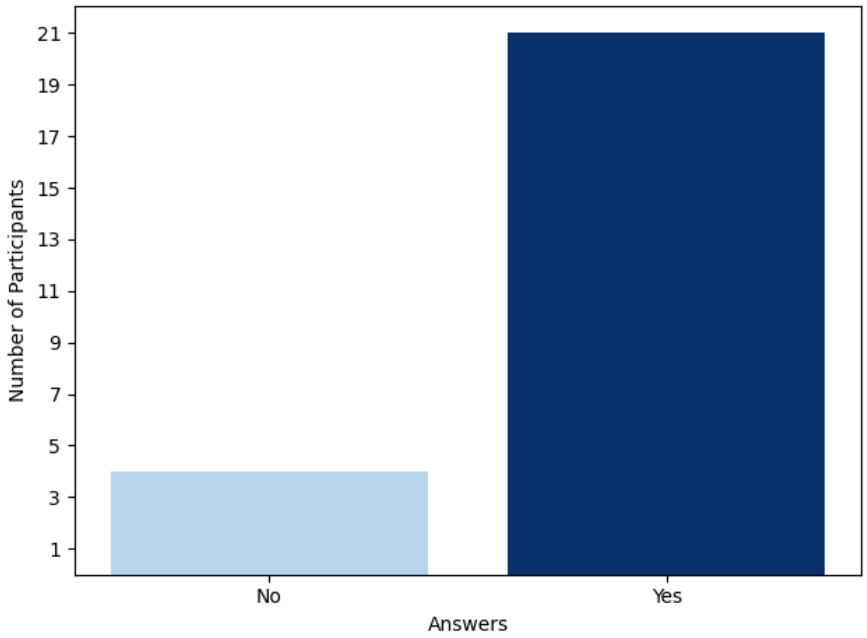


Figure 6.8. Participants' responses on whether the system helped them explore courses outside of their usual preferences.

Satisfaction - Likelihood to Use. Q6) How likely are you going to use the system to help with your career plan?

Participants were also asked how likely they were to use the system to assist with their career planning. The average rating for this question was 3.48, i.e., 69.6%, indicating a moderate level of satisfaction. The distribution of responses shows that while 12 participants rated their likelihood of using the system at 4, a smaller group of participants expressed lower likelihoods, with 2 participants rating it at 1 and 4 participants at 2 (Figure 6.9).

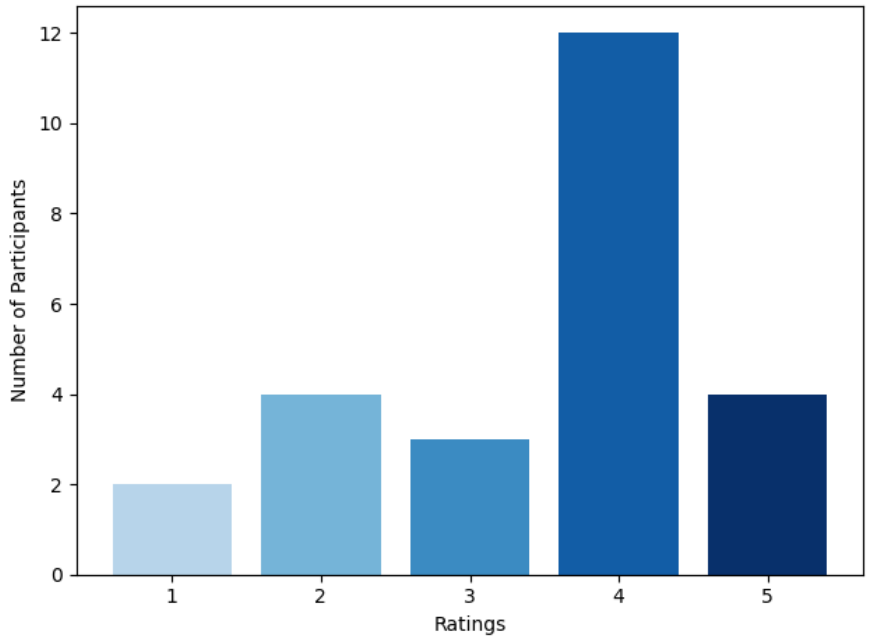


Figure 6.9. Participants' likelihood of using the system to assist with their career planning.

1: Not likely at all, 5: Very likely

Satisfaction - Likelihood to Recommend. Q7) How likely are you going to recommend the system to your peers?

Finally, participants were asked how likely they were to recommend the system to their peers. The average rating for this question was 3.88, i.e., 77.6%, suggesting that most participants would consider recommending the system. Ten participants gave the highest rating of 5, while 7 rated it at 4 (Figure 6.10). The lower ratings were fewer, with 1 participant rating it at 1 and 3 participants at 2, showing that while the system was generally well-received, there is still room for improvement in encouraging users to recommend it more strongly.

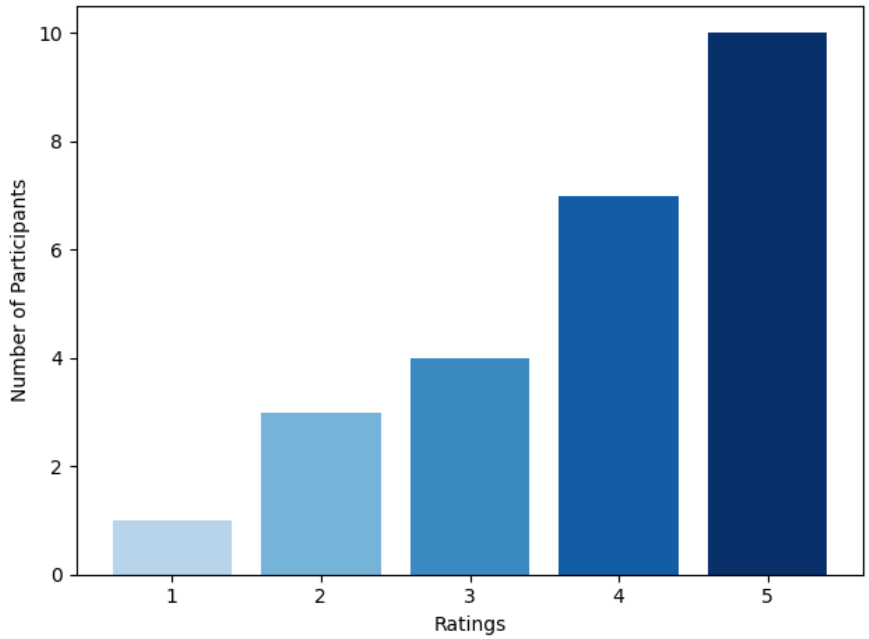


Figure 6.10. Participants' likelihood of recommending the system to their peers. *1: Not likely at all, 5: Very likely*

These results provide a comprehensive view of the system's performance from multiple perspectives, highlighting areas where it excelled and others where it could be further refined. The high scores in transparency and efficiency suggest that the system is user-friendly and clearly communicates its recommendations, while the moderate satisfaction ratings indicate opportunities for enhancing user engagement and perceived relevance.

6.1.6. Statistical Analysis of Survey Data

To gain deeper insights into the survey responses, I performed several statistical analyses, beginning with the Shapiro-Wilk test [140] to assess the normality of the data. The results indicated that the data did not follow a normal distribution, prompting the use of non-parametric tests to analyze the variations in responses across different participant groups. The following sections detail the tests conducted, along with the key findings.

Normality Test. The Shapiro-Wilk test was applied to the data from the six linear scale questions. The null hypothesis (H_0) of the Shapiro-Wilk test posits that the data is normally distributed. However, with p-values less than 0.01 for all questions, the null hypothesis was rejected at a 99% confidence level, indicating that the data is not normally distributed. Consequently, non-parametric tests were deemed appropriate for further analysis.

Group Comparisons. Given the non-normal distribution of the data, the Kruskal-Wallis H test [141] was employed to examine differences in responses across groups based on participants' education level, employment status, and job experience. Participants were categorized into relevant groups within each dimension to identify any significant differences in their responses to the survey questions.

Participants were categorized into three groups for education level dimension: those currently enrolled in a Bachelor's degree program, those who have completed a Bachelor's degree, and those enrolled in or holding a Master's degree. The Kruskal-Wallis H test results for the education status dimension, as shown in Table 6.2, revealed a significant difference in responses only for the Persuasiveness question (Q2), with a p-value of 0.0234, indicating a 97% confidence level. This suggests that participants' education level influenced how persuasive they found the course recommendations. For the other questions—Effectiveness, Transparency, Efficiency, Serendipity, Satisfaction (Likelihood to Use), and Satisfaction (Likelihood to Recommend)—no significant differences were observed between the groups, with p-values above 0.05 threshold.

Table 6.2. Kruskal-Wallis H test results for the education level dimension.

Question	**Statistic**	**p-value**	**Result**	**Confidence L.**
Effectiveness (Q1)	5.9859	0.0501	No significant difference	95%
Persuasiveness (Q2)	7.5095	0.0234	Significant difference	97%
Transparency (Q3)	0.7399	0.6908	No significant difference	95%
Efficiency (Q4)	0.2962	0.8623	No significant difference	95%
Serendipity (Q5)	1.2959	0.5231	No significant difference	95%
Satisfaction (Likelihood to Use, Q6)	5.5652	0.0619	No significant difference	95%
Satisfaction (Likelihood to Rec., Q7)	4.5680	0.1019	No significant difference	95%

For employment status, participants were divided into four groups: Unemployed, Studying, Working, and Studying and Working. The Kruskal-Wallis H test results for employment status, as summarized in Table 6.3, no significant differences were found in any of the questions, with p-values consistently above the 0.05 threshold. This uniformity in responses suggests that the system's performance, as perceived by the participants, is robust across different employment contexts, whether the users are unemployed, studying, working, or balancing both study and work.

Table 6.3. Kruskal-Wallis H test results for the employment status dimension. Values highlighted in yellow indicate statistically significant differences.

Question	**Statistic**	**p-value**	**Result**	**Confidence L.**
Effectiveness (Q1)	4.3278	0.2282	No significant difference	95%
Persuasiveness (Q2)	7.4516	0.0588	No significant difference	95%
Transparency (Q3)	0.6765	0.8787	No significant difference	95%
Efficiency (Q4)	1.4296	0.6986	No significant difference	95%
Serendipity (Q5)	3.1209	0.3734	No significant difference	95%
Satisfaction (Likelihood to Use, Q6)	3.8654	0.2764	No significant difference	95%
Satisfaction (Likelihood to Rec., Q7)	5.5944	0.1331	No significant difference	95%

For job experience, participants were divided into four groups: having no job experience, 1 or less than 1 year of experience, 2 to 4 years of experience, and 5 or more years of experience. The Kruskal-Wallis H test results for job experience, as summarized in Table 6.4, a significant difference was observed in the Persuasiveness question (Q2), with a p-value of 0.0128, suggesting that participants' level of professional experience influences how persuasive they found the system's course recommendations. However, for other questions—Effectiveness, Transparency, Efficiency, Serendipity, and both Satisfaction—no significant differences were detected, indicating a consistent perception of the system's effectiveness and usability across varying levels of job experience.

Table 6.4. Kruskal-Wallis H test results for the job experience dimension. Values highlighted in yellow indicate statistically significant differences.

Question	**Statistic**	**p-value**	**Result**	**Confidence L.**
Effectiveness (Q1)	4.5163	0.2108	No significant difference	95%
Persuasiveness (Q2)	10.8046	0.0128	Significant difference	98%
Transparency (Q3)	0.7399	0.8638	No significant difference	95%
Efficiency (Q4)	1.0222	0.7959	No significant difference	95%
Serendipity (Q5)	2.5714	0.4625	No significant difference	95%
Satisfaction (Likelihood to Use, Q6)	6.1253	0.1057	No significant difference	95%
Satisfaction (Likelihood to Rec., Q7)	6.0869	0.1075	No significant difference	95%

These findings suggest that the XCRS provides a generally consistent user experience across diverse educational backgrounds, employment statuses, and levels of job experience. The significant differences found in the Persuasiveness dimension indicate that factors such as education and job experience may play a role in how users perceive the recommendations' ability to influence their decisions.

Friedman Test. To further investigate whether there were any significant differences in how participants rated various aspects of the XCRS, a Friedman test [142] was conducted. The Friedman test is a non-parametric test used to detect differences in responses across multiple related groups, making it suitable for evaluating the consistency of participants' ratings across the different questions used to assess the system.

The Friedman test was applied to the responses to all the questions that evaluated the system, which included aspects such as effectiveness, satisfaction, persuasiveness, transparency, efficiency, and serendipity. The null hypothesis (H0) for the Friedman test posits that there is no significant difference between the responses to the different questions. The results of the test yielded a statistic of 27.9194 with a p-value of 0.0000377. This p-value is well below the conventional significance threshold of 0.05, leading to the rejection of the null hypothesis at a 99% confidence level. This finding indicates that there is a significant difference in how participants rated the various aspects of the system.

6.2. Expert Judgment Study

To further evaluate the XCRS, an expert judgment study was conducted involving three distinguished professionals with diverse backgrounds and expertise. The experts included:

- An academic specializing in LLMs and active in the education sector.
- An academic whose research focuses on Recommender Systems.
- A senior manager in the telecommunication sector with extensive experience mentoring IT professionals and a deep understanding of the industry's workforce needs.

These experts were selected for their ability to evaluate the system from different perspectives, ensuring a comprehensive assessment of the XCRS. Each expert's unique viewpoint allowed the system to be examined through a multidisciplinary lens.

6.2.1. Study Design and Method

The expert judgment study was designed to parallel the user study but with the added depth and rigor appropriate for professionals with advanced knowledge in their respective fields. Three distinct personas were created, each representing a different background within the IT field. Figures containing knowledge unit information for each persona are shown in Figure 6.11.



Computer Science Student (Junior)

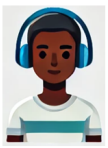
Courses and Concepts I Enjoyed:
Digital Circuits, Linux, GitHub, Computer Security, Object Oriented Programming, Computer Organization, Solidity, Rust, Database Systems, Computer Operating Systems, Redis, PostgreSQL

Courses and Concepts I Found Neutral:
Intr. to Computer Engineering and Ethics, Data Structures, Python, C++, C

Courses and Concepts I Didn't Enjoy:
Technical Communication for Computer Engineers, Numerical Methods, Mathematics I, Discrete Mathematics, Engineering Mathematics, Basics of Electrical Circuits, Introduction to Electronics, Probability and Statistics

Courses and Concepts I'm Curious About:
Nodejs, AWS, Cloud, React

(a)



Graphic Design Graduate

Courses and Concepts I Enjoyed:
Introduction to Graphic Design Programs, Drawing for Graphic Design, Form and Structure, Typography, Computer-Aided Design, Digital Typography, Web Design, Graphic Design, Adobe Illustrator, HTML, CSS

Courses and Concepts I Found Neutral:
Desktop Publishing, Illustration, Digital Illustration, Social Media Design, Photography, SQL, GitHub, Git, Adobe Photoshop

Courses and Concepts I Didn't Enjoy:
Design Psychology, History of Graphics Design, Sociology of Art, Philosophy of Art, Image Aesthetics

Courses and Concepts I'm Curious About:
Angular, React

(b)



Computer Engineering Graduate

Courses and Concepts I Enjoyed:
Introduction to Information Systems, Digital Circuits, Numerical Methods in Computer Engineering, Linear Algebra and Applications, Computer Architecture, Object Oriented Programming, Computer Organization, Microprocessor Systems, Computer Engineering Design I, Intro. to Artificial Intelligence, Computer Vision, Natural Language Processing, NumPy, PyTorch

Courses and Concepts I Found Neutral:
Physics, Intr. to Computer Engineering and Ethics, Calculus, Discrete Mathematics, Engineering Mathematics, Data Structures, Introduction to Electronics, Database Systems, Computer Operating Systems, Intro. to Machine Learning, Deep Learning, Java, Python, C++, C, Linux, SVN, SQL, TCP/IP, NoSQL

Courses and Concepts I Didn't Enjoy:
Digital Electronic Circuits, Signal and Systems for Computer Engineers, Basics of Electrical Circuits, Computer Graphics, Robotics, Probability and Statistics, Memory Management, Heaps, Recursion, Shell Scripting

Courses and Concepts I'm Curious About:
Software Quality and Testing, Functional Programming, React, JavaScript, Design Patterns

(c)

Figure 6.11. Personas used in the expert judgment study: (a) computer science student, (b) graphic design graduate, and (c) computer engineering graduate.

These personas were carefully crafted by analyzing relevant undergraduate programs to ensure their knowledge and backgrounds were realistic, albeit fictional. The experts were provided with detailed information about each persona's mastered concepts, neutral concepts, disliked concepts, and areas of curiosity.

The expert judgment study began by providing personas to the XCRS, which then generated role and course recommendations using five different LLMs. The experts were tasked with thoroughly reviewing these recommendations and evaluating the system based on various criteria, including relevance, accuracy, quality of explanations, and overall effectiveness. The evaluation aimed to capture both quantitative and qualitative insights, with the experts offering their professional judgment on the system's performance across the different personas.

The core of the expert evaluation focused on several key dimensions to assess the system's performance and the quality of its recommendations:

(i) Model Satisfaction: Similar to the user study, experts were asked to identify which model's recommendations they were most satisfied with for each persona. Additionally, they were asked to select the models they found most reliable overall, offering insights into the comparative performance of the different LLMs used in the system.

(ii) Relevance and Accuracy of Recommendations: Experts were asked to evaluate how relevant and accurate they found the recommended roles and courses for each persona, considering both the personas' known concepts and completed courses, as well as their background and interests. This dimension aimed to determine the system's effectiveness in aligning recommendations with the persona's knowledge and career goals.

(iii) Clarity and Quality of Explanations: Experts assessed the clarity and understandability of the explanations provided for the recommended roles and courses. This evaluation was crucial in determining whether the system could effectively communicate the rationale behind its recommendations to users.

(iv) Overall Effectiveness: The experts were also asked to rate the overall effectiveness of the system in recommending suitable roles and courses for each persona. This dimension provided insights into the system's ability to deliver comprehensive and relevant recommendations that align with users' professional development needs.

(v) Qualitative Feedback and Suggestions: Beyond the quantitative ratings, experts were invited to provide qualitative feedback on the system's performance. This included detailed observations on the relevance and accuracy of recommendations, the quality of explanations, and suggestions for improving the system. The experts also compared the XCRS with other recommendation systems they had used or evaluated, offering valuable perspectives for future enhancements.

The combination of quantitative assessments and qualitative feedback provided a thorough evaluation of the XCRS, highlighting both its strengths and areas for potential improvement. The insights gathered from this expert judgment study are crucial for refining the system to better meet the needs of users with diverse backgrounds and expertise. To provide a clear understanding of the specific areas evaluated, the survey questions used in this expert judgment study are detailed in Appendix B.

6.2.2. Results

The expert judgment study aimed to evaluate the performance of the XCRS across different personas by utilizing the expertise of professionals with deep knowledge in LLMs, Recommender Systems, and IT industry practices. The study focused on the satisfaction levels of the experts with the recommendations provided by five distinct models, as well as their overall assessments of the system's relevance, accuracy, explanatory quality, and overall effectiveness.

Model Satisfaction. Similar to the user study, the experts in the judgment study evaluated the recommendations generated by five models. The models used were the same as in the User Study: text-embedding-004 by Google, voyage-large-2 by VoyageAI,

text-embedding-3-large by OpenAI, mistral-embed by MistralAI, and embed-english-v3.0 by Cohere. The models were labeled as Model 1 through Model 5 during the study to prevent any bias, but the actual model names are used here for reporting the results.

Table 6.5 presents the model preferences of the three experts for each persona, based on the question: *Which model's recommendations were you most satisfied with for the persona?* For the junior-level computer science student persona (Persona A), two experts preferred the recommendations generated by text-embedding-3-large by OpenAI (Model 3), while one expert favored mistral-embed by MistralAI (Model 4). For the graphic design graduate persona (Persona B), the experts were divided, with one expert each preferring the recommendations of voyage-large-2 by VoyageAI (Model 2), text-embedding-3-large by OpenAI (Model 3), and embed-english-v3.0 by Cohere (Model 5). Similarly, for the computer engineering graduate persona (Persona C), preferences were again varied, with one expert each selecting text-embedding-004 by Google (Model 1), text-embedding-3-large by OpenAI (Model 3), and mistral-embed by MistralAI (Model 4).

Table 6.5. Model preferences of the three experts for each persona, displaying the selected models for Persona A, B, and C.

Model 1: text-embedding-004 (Google) Model 2: voyage-large-2 (VoyageAI) Model 3: text-embedding-3-large (OpenAI) Model 4: mistral-embed (MistralAI) Model 5: embed-english-v3.0 (Cohere)

	Computer Science Student	**Graphic Design Student**	**Computer Engineering Graduate**
Expert 1	Model 3	Model 3	Model 3
Expert 2	Model 4	Model 5	Model 1
Expert 3	Model 3	Model 2	Model 4

In addition to evaluating models for each persona, the experts were asked to select up to two models they were most satisfied with overall. The results of this question are visualized in Figure 6.12, which shows the distribution of model selections by experts. The figure highlights a clear preference for text-embedding-004 by Google (Model 1) and mistral-embed by MistralAI (Model 4), each receiving two votes, while text-embedding-3-large by OpenAI (Model 3) received one vote. These results suggest that while individual personas may lead to varied model preferences, overall satisfaction tends to cluster around certain models.

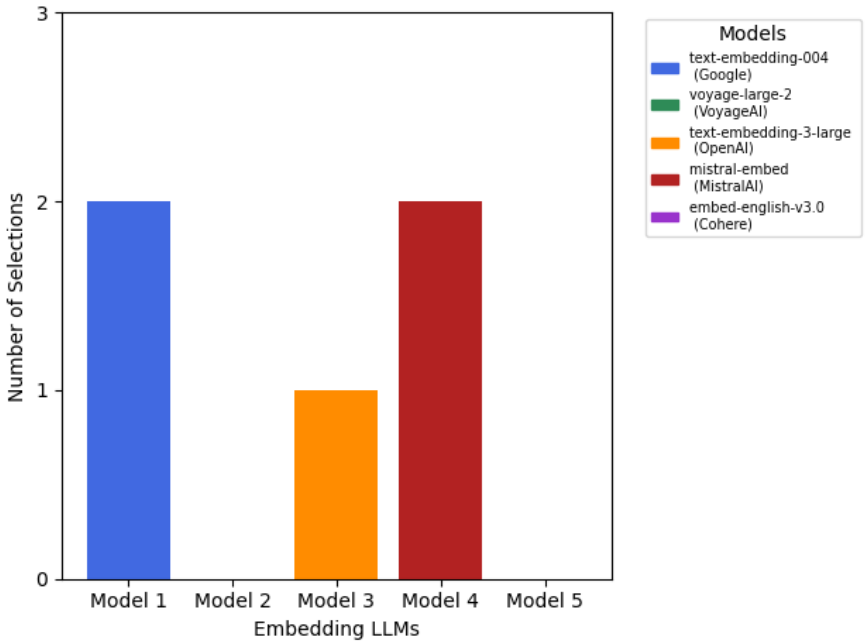


Figure 6.12. Distribution of model selections by experts, showing the total number of selections for each model across all personas.

The following analysis delves into the system's performance in its most refined form, focusing on the results produced by the best-performing LLM. Expert evaluations were based on key aspects such as relevance, accuracy, quality of explanations, and overall effectiveness of the recommendations. The experts provided ratings for seven quantitative questions using a Likert scale [143] (from 1 to 5), assessing the system's ability to generate suitable roles and courses for three distinct personas. These ratings,

presented in Table 6.6, offer a numerical summary of the system's performance across these key dimensions, providing valuable insights into how well the recommendations align with the personas' needs.

Table 6.6. Expert ratings on evaluation dimensions.

Dimension	**Scale**		**Expert 1**			**Expert 2**			**Expert 3**			**Average for each dimension**
			Personas									
	1 ...	**... 5**	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	
Relevance - Role	Not relevant	Very relevant	2	2	5	5	3	5	4	3	5	3.78
Relevance - Course	Not relevant	Very relevant	3	2	5	4	4	4	4	4	5	3.89
Accuracy - Role	Not accurate	Very accurate	5	3	4	5	3	5	4	3	5	4.11
Accuracy - Course	Not accurate	Very accurate	5	4	4	4	4	4	5	4	5	4.33
Quality of Explanations - Role	Not clear and understandable	Very clear and understandable	5	5	3	4	2	5	5	4	5	4.22
Quality of Explanations - Course	Not clear and understandable	Very clear and understandable	5	5	4	5	4	5	5	4	5	4.67
Overall Effectiveness	Not effective	Very effective	5	3	5	5	4	5	4	4	5	4.44
Average for experts' ratings for persona			4.29	3.43	4.29	4.57	3.43	4.71	4.43	3.71	5	4.21
Average for experts' ratings				4			4.24			4.38		

Relevance. Q1-Q2) How relevant do you find the recommended roles/courses for the persona's knowledge, considering only known concepts and completed courses, excluding personal interests?

The experts found the relevance of the recommended roles for the personas' knowledge to be generally high, with an average rating of 3.78 out of 5 (75.56%). The majority of ratings were concentrated in the higher end, with four ratings of 5, indicating strong alignment with the personas' knowledge. The relevance of the recommended courses was rated slightly higher, with an average of 3.89 out of 5 (77.78%). This suggests that the system was particularly effective in recommending courses that matched the personas' known concepts and completed courses (Figure 6.13).

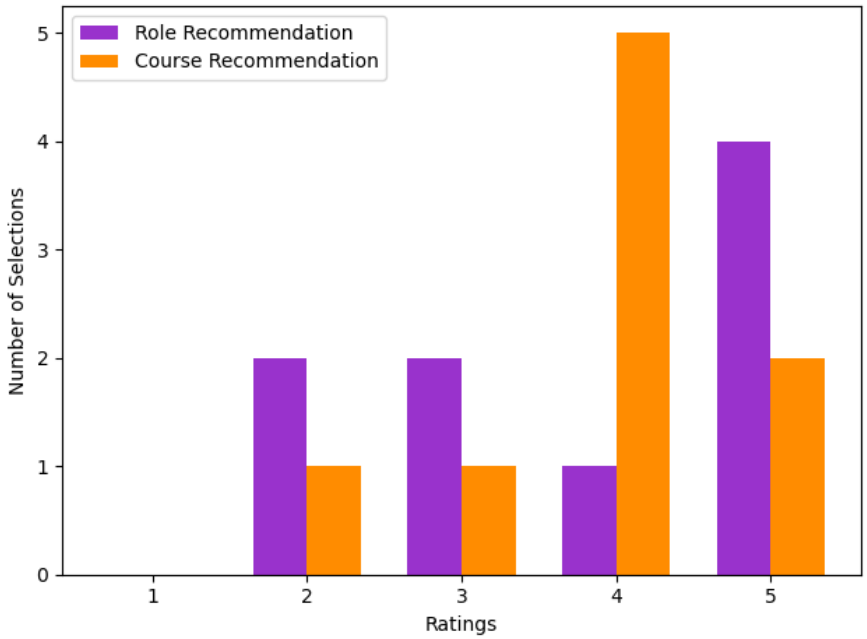


Figure 6.13. Experts' ratings on the relevance of role and course recommendations. *1: Not relevant at all, 5: Very relevant*

Accuracy. Q3-Q4) How accurate do you find the recommended roles/courses for the persona when considering both their background and interests?

When considering both the personas' backgrounds and interests, the accuracy of the recommended roles received an average rating of 4.11 out of 5 (82.22%). The experts consistently rated this aspect highly, with four ratings of 5 and no ratings below 3, indicating strong confidence in the system's ability to match roles with the personas' profiles. The accuracy of the recommended courses was rated even higher, with an average of 4.33 out of 5 (86.67%). This reflects the system's effectiveness in integrating both background knowledge and interests into the course recommendations, ensuring that they are not only relevant but also well-suited to the personas' career aspirations (Figure 6.14).

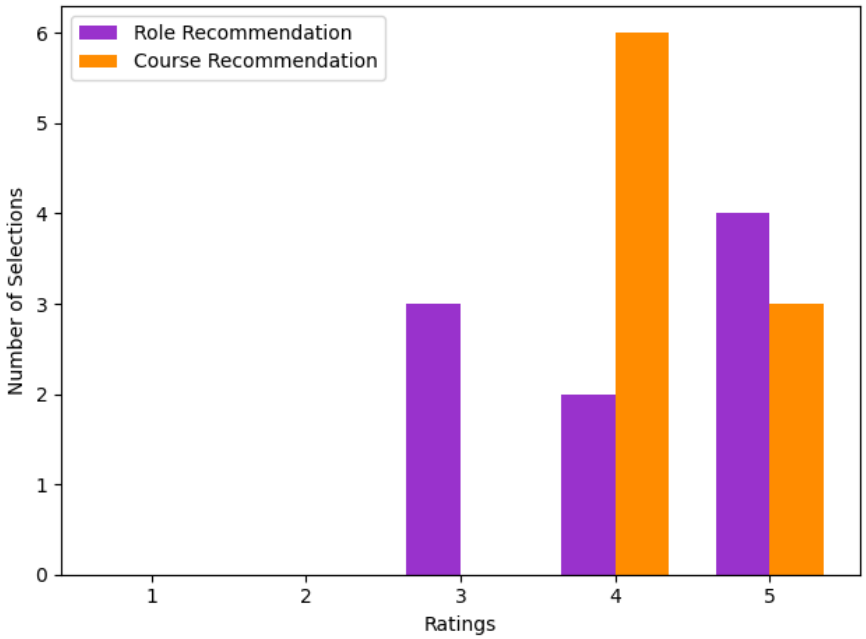


Figure 6.14. Experts' ratings on the accuracy of role and course recommendations. *1: Not accurate at all, 5: Very accurate*

Quality of Explanations. Q5-Q6) How clear and understandable are the explanations provided for the recommended roles/courses for the persona?

The clarity of the explanations provided for the recommended roles was also evaluated positively, with an average rating of 4.22 out of 5 (84.4%). The experts appreciated the system's ability to clearly communicate the reasoning behind the role recommendations, which is crucial for building user trust and understanding. The explanations for the recommended courses were rated even higher, with an average of 4.67 out of 5 (93.33%), indicating that the system was highly effective in making the course recommendations easy to understand and actionable (Figure 6.15).

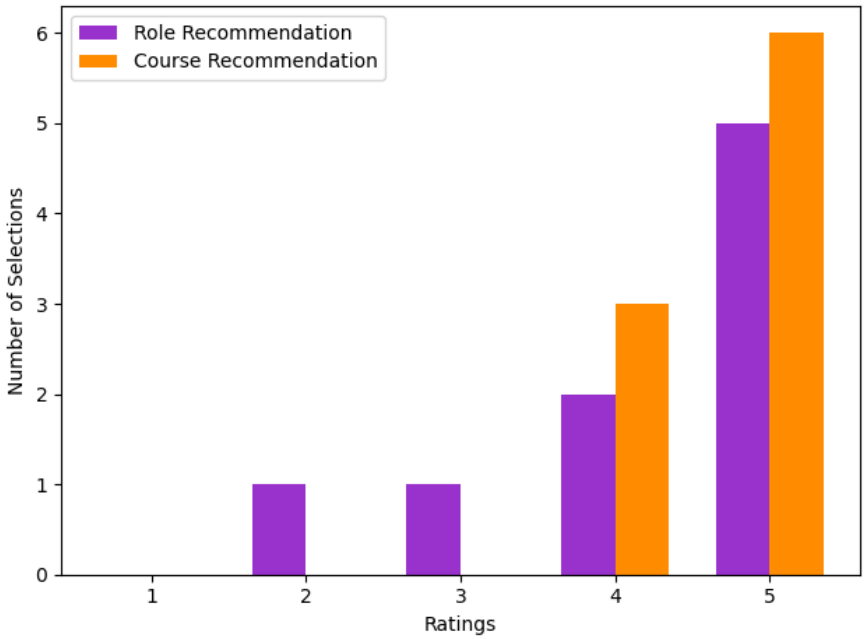


Figure 6.15. Experts' ratings on the quality of explanations for the role and course recommendations.

1: Not clear and understandable at all, 5: Very clear and understandable

Overall Effectiveness. Q7) How effective do you find the system overall in recommending suitable roles and courses for the persona?

Finally, the overall effectiveness of the system in recommending suitable roles and courses for the personas received an average rating of 4.44 out of 5 (88.89%). This high rating underscores the system's capability to deliver comprehensive and relevant recommendations that align with the users' professional development needs (Figure 6.16).

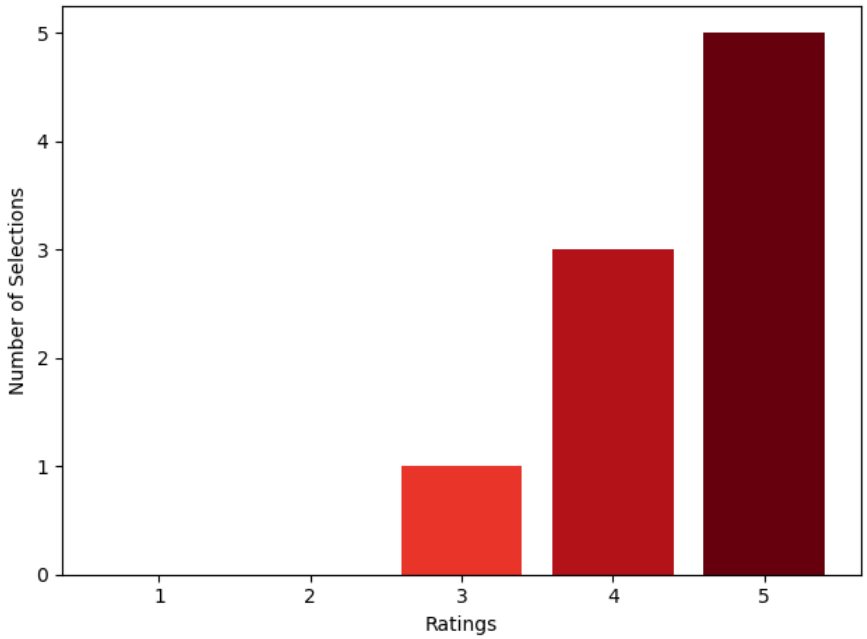


Figure 6.16. Experts' ratings on the system's overall effectiveness.
1: Not effective at all, 5: Very effective

These results demonstrate that the XCRS performs exceptionally well in delivering relevant, accurate, and clear recommendations, particularly in the context of course recommendations. The system's ability to integrate and articulate the relationship between the user's background, interests, and the recommended roles and courses is a key strength, as evidenced by the high ratings across multiple dimensions. The figures accompanying this section further illustrate the distribution of expert responses, highlighting the system's strengths in these critical areas.

6.2.3. Qualitative Feedback and Suggestions

The qualitative feedback provided by the experts offered valuable insights into the strengths and areas for improvement of the XCRS. The experts evaluated the system's performance based on the relevance and accuracy of the recommendations, the quality of the explanations, and the overall effectiveness. Their feedback also included suggestions for enhancing the system and reflections on the models' reliability.

The experts generally found the recommendations to be relevant and accurate, although the effectiveness varied depending on the persona and the balance between background and interests. Expert 1 noted that for Persona A and Persona B, the recommendations were less relevant when personal interests were excluded, particularly highlighting Model-3 for its ability to incorporate interests effectively. However, for Persona C, which had a more diverse background, the system made recommendations that were closely aligned with the persona's educational history, resulting in higher relevance but slightly lower accuracy when interests were considered less.

Expert 2 provided a more critical view, noting that the recommendations for Persona C were particularly well-suited but pointed out that some course recommendations, such as SSL, seemed unexpected and less relevant. Expert 3 also praised the system's performance for Persona C but highlighted that Persona B, with a background less aligned with computer science, received less relevant and somewhat forced recommendations. This expert suggested that the system should consider not making recommendations when the user's profile does not closely align with the available career roles or at least acknowledge when a recommendation is weaker.

The experts generally agreed that the explanations provided by the system were clear and sufficient, particularly for Persona A and Persona B. Expert 1 mentioned that while the explanations for Persona C were slightly weaker, they were still understandable. Expert 2 observed that the quality of explanations tended to correlate with the quality of the recommendations themselves—when the recommendations were strong, the explanations were strong as well, but when the recommendations were less fitting, the explanations felt forced. Expert 3 echoed this sentiment, confirming that the explanations were adequate and clear, making it evident why certain roles or courses were recommended.

In terms of overall effectiveness, Expert 1 suggested that the system was generally effective, especially for Persona A and Persona C, but noted a gap in the recommendations for Persona B, where expected roles such as UX Designer or Game Developer

were missing. Expert 2 highlighted that at least one of the five models consistently provided good recommendations, affirming the system's effectiveness. Expert 3 offered a broader perspective, suggesting that the system could be even more effective if it incorporated a mechanism to express confidence in its recommendations, potentially refraining from making recommendations when confidence is low.

The experts provided several constructive suggestions to enhance the system:

(i) Balancing Background and Interests: Expert 1 recommended adjusting the balance between background and interests in the recommendation algorithm, suggesting a 60% background and 40% interest split. They also proposed a dynamic approach where the system could shift the balance based on whether the user's interests align with their background, potentially offering feedback if the system detects a misalignment.

(ii) Course Recommendation Criteria: Expert 2 emphasized the need to prioritize courses with broader coverage of relevant concepts, potentially integrating additional input from the user to clarify whether they prefer recommendations based on their background or their interests. They also suggested presenting the concepts associated with each course recommendation to the user.

(iii) Diversity of Models: Expert 3 proposed retaining multiple LLMs in the final system rather than relying on a single "best" model. By allowing users to receive recommendations from 2-3 models, the system could provide more diverse options, thereby enhancing user satisfaction and the perceived value of the recommendations.

When asked to compare the XCRS with other recommendation systems, the experts generally noted the novelty and effectiveness of the XCRS. Expert 2 suggested that the system might offer distinct advantages over existing platforms like Udemy, particularly in its approach to integrating content and providing personalized recommendations. While the other experts did not have direct comparisons to other systems, they recognized the potential of the XCRS to set a new standard in educational and

career recommendation systems.

In terms of model reliability, Model-3 (text-embedding-3-large by OpenAI) emerged as a strong performer, particularly for Persona A, with Expert 1 explicitly stating that it provided the most relevant and accurate recommendations. Experts 2 and 3 identified Model-1 (text-embedding-004 by Google) and Model-4 (mistral-embed by MistralAI) as reliable across different personas, with Model-3 also performing well in specific cases. However, Model-5 (embed-english-v3.0 by Cohere) was criticized for its weaker performance, particularly when it recommended roles that seemed inconsistent with the persona's background.

7. DISCUSSION

This chapter presents a comprehensive analysis of the evaluation findings for the XCRS, aiming to elucidate the system's capabilities, limitations, and areas for enhancement. By synthesizing results from both a user study and an expert judgment study, the discussion delves into the system's performance in generating course and career role recommendations based on large language model (LLM) embeddings. The discussion reflects on key themes, including user experience, model efficacy, and expert insights, situating these findings within the broader context of explainable AI systems in educational and career recommendation domains.

The evaluation results underscore XCRS's strengths, particularly in terms of transparency, usability, and relevance of recommendations. The user study, involving participants from various IT-related backgrounds, reveals that the system's design effectively supports users in navigating the recommendations and comprehending the accompanying explanations. High satisfaction ratings for transparency (89.6%) and efficiency (88.8%) highlight the system's success in not only conveying the rationale behind its recommendations but also providing an intuitive interface that aligns with user expectations.

Parallel insights emerged from the expert judgment study, which provided an in-depth critique from domain experts in education, recommender systems, and industry practice. The expert evaluations affirmed the system's ability to generate recommendations that are both relevant and accurate, particularly within the context of course recommendations. Experts noted that XCRS effectively integrates users' educational backgrounds and interests, thereby facilitating precise and personalized guidance. Moreover, the experts commended the system's explanation clarity, which enhances the interpretability and transparency of the recommendations, contributing to the overall perceived effectiveness of XCRS.

The user study results provided a multifaceted evaluation of the recommendations generated for both career roles and courses. Across key dimensions—effectiveness, persuasiveness, transparency, efficiency, serendipity, and satisfaction—the XCRS received generally positive feedback.

In terms of effectiveness, participants rated the relevance of the recommendations highly (80%), indicating that the system's LLM embeddings effectively aligned with their backgrounds and interests. Persuasiveness was similarly strong (76%), suggesting that the recommendations were compelling enough for users to consider new courses or career roles. These results highlight the system's potential to influence users' educational and career pathways.

Transparency, as previously noted, was a standout aspect of XCRS, with participants finding the explanations clear and comprehensive, contributing to a higher level of trust and engagement. The user interface was well received, as reflected in the high efficiency ratings, supporting seamless navigation through the recommendation process. The system also demonstrated strong serendipity (84%), guiding users toward exploring new and diverse courses outside their usual preferences.

Satisfaction levels varied, with a 69.6% likelihood of future use and a 77.6% likelihood of recommending the system to others. While these ratings suggest general contentment, the moderate score for personal use indicates room for further refinement to enhance engagement and perceived relevance.

To further clarify, this thesis used several evaluation dimensions from [28], which are effectiveness, persuasiveness, transparency, efficiency, and satisfaction, to assess XCRS. While these dimensions primarily focus on explanations in recommender systems, our evaluation expands to include the system's overall functionality. For example, effectiveness in [28] refers to decision-making support, but in our study, it measures the perceived relevance of the recommendations, thereby indirectly supporting decision-making. Similarly, efficiency in [28] relates to decision speed, whereas by asking about

the ease of use in our study, we are indirectly measuring how quickly users can make decisions with the system.

The statistical tests applied in the user study—Shapiro-Wilk, Kruskal-Wallis H, and Friedman tests—validated the robustness of the XCRS evaluation. The Shapiro-Wilk test's confirmation of non-normality justified the use of non-parametric tests, ensuring that the analysis aligned with the data's characteristics. The Kruskal-Wallis H test revealed that while user demographics such as education level and job experience influenced perceptions of persuasiveness, other dimensions, like transparency and efficiency, were perceived consistently across diverse groups. This highlights both the system's broad applicability and the need for targeted improvements in persuasive capability.

The Friedman test's identification of significant differences across user perceptions of various system dimensions emphasizes the need to balance the overall user experience. While transparency and efficiency received high praise, focusing on enhancing satisfaction and persuasiveness could lead to increased engagement and likelihood of use. These insights underscore how statistical analyses can guide targeted refinements in recommendation systems like XCRS to address diverse user needs more effectively.

Insights from the expert judgment study largely corroborate the findings of the user study, offering a complementary perspective that reinforces the key strengths of the XCRS. The experts, representing a range of relevant fields, evaluated the recommendations across multiple personas, focusing on relevance, accuracy, explanation quality, and overall effectiveness. These evaluations are consistent with user feedback, particularly in the high ratings for transparency and the system's ability to provide accurate and contextually relevant recommendations. Both studies noted that course recommendations, in particular, were highly aligned with user needs, indicating the robustness of XCRS in understanding and mapping user knowledge units to appropriate educational pathways.

The qualitative feedback from the experts underscores the importance of refining the balance between background knowledge and personal interests in the recommendation process. While the system generally performed well, especially in providing clear explanations and effective recommendations for users with strong alignments to the computer science domain, it faced challenges when dealing with less traditional profiles. The experts' suggestions for improvement, particularly in course selection criteria and the diversity of models, offer valuable guidance for enhancing the XCRS. By addressing these areas, the system can improve its ability to cater to a broader range of users and provide more personalized and confident recommendations.

Prior work [36] emphasizes the growing role of LLMs in enhancing the personalization of recommendation systems. Another study [144] underlines that LLMs introduce new opportunities for recommendation systems by offering zero-shot and few-shot recommendation capabilities. Due to their nature, they are a probable solution to cold-start problems in recommendation systems. These models are capable of generalizing to unseen candidates because of their extensive pre-training on factual knowledge, domain expertise, and common-sense reasoning, allowing them to make accurate recommendations even for items or users they haven't encountered before [36]. The XCRS builds on this paradigm by successfully leveraging LLM embeddings to offer top-k recommendations, coupled with explanations that are accessible and meaningful to end-users. According to the categorization of existing literature, XCRS fits into multiple frameworks: it aligns with the 'LLM Embeddings + Recommendation Systems' model as described by Wu et al. [36], where LLMs are used to extract features for traditional models; it also corresponds to Zhao et al.'s [37] category of 'LLMs act as a recommender,' where LLMs directly perform recommendation functions; and it fits into the 'LLMs for list-wise recommendations' framework defined by Chen et al. [38], enhancing ranking and feature engineering within traditional recommendation systems.

Furthermore, explainable recommendation systems, as examined by Wang et al. [39, 41], Chen et al. [40], and others, have been shown to enhance user trust and engagement through clear explanations that highlight the alignment between user pref-

erences and recommended items. The strong user and expert responses to the transparency of XCRS resonate with these findings, underscoring the importance of explanation quality in fostering an informed and satisfactory user experience.

The comparative analysis of LLMs within XCRS revealed distinct user and expert preferences, indicating that different models excel in various recommendation contexts. For instance, mistral-embed by MistralAI performed best for course recommendations, while voyage-large-2 by VoyageAI excelled in career role recommendations. This suggests that while each model performs well in specific scenarios, there is no single model that universally outperforms others across all recommendation types.

Interestingly, user satisfaction did not always align perfectly with usefulness scores. text-embedding-004 by Google, despite not having the top rating for usefulness in either career or course recommendations, was the model with which users reported the highest overall satisfaction. This divergence highlights the importance of considering both objective metrics (e.g., accuracy, relevance) and subjective user preferences (e.g., satisfaction, perceived clarity) when evaluating recommendation models. Expert feedback mirrored these findings, showing varied model preferences across different personas. While text-embedding-3-large by OpenAI received consistent recognition for its relevance in particular scenarios, mistral-embed by MistralAI and text-embedding-004 by Google were highlighted as reliable performers across multiple contexts. This distribution of preferences across both user and expert evaluations underscores the value of model diversity within XCRS, suggesting that providing users with access to recommendations from multiple models may enhance the overall experience by catering to varied needs and preferences.

7.1. Comparative Analysis with the Related Work

When comparing XCRS to other recommendation systems summarized in Table 3.1, several key differentiators emerge, particularly in its approach to explainability and the use of LLM embeddings. While many systems, such as those developed by Vo

et al. [7] and Striebel et al. [88], apply knowledge graphs (KGs), collaborative filtering, and data mining techniques, XCRS uniquely leverages LLM embeddings to provide transparent and contextually rich explanations for both career role and course recommendations. This approach enables a more flexible, user-specific alignment of educational background and interests compared to template-based explanations or KG-based recommendations.

Other systems, such as those by Jiang et al. [91], and Frej et al. [62], use reinforcement learning to enhance recommendation precision and align skills with market needs. Although XCRS does not directly focus on market demand alignment, it provides diverse recommendations that balance user interests and educational profiles through LLMs, broadening its relevance to both career guidance and academic development.

Recommendation models like those presented by Abu-Rasheed et al. [100], and Zylowski et al. [106] employ LLMs within conversational interfaces, aiming to enhance user experience through dialogue. However, XCRS's distinction lies in offering recommendations from multiple top LLMs, allowing users to compare results across different models and offering a more varied and flexible recommendation experience, rather than being limited to a single model's conversational outputs.

XCRS also stands out in its ability to provide comprehensive, explainable recommendations that cater to both education and career contexts, unlike more domain-specific systems such as the one proposed by Yang et al. [101] (focused on talent training) or Chau [104] (aligning courses to career activities). By leveraging multiple LLMs, XCRS aims to enhance both the accuracy and the perceived relevance of its recommendations, offering a balance between the exploration of new pathways (serendipity) and relevance to user profiles.

In contrast to the model proposed by Wang et al. [107], which uses Bayesian variational networks for employee training recommendations, XCRS's multi-model LLM approach provides explainability directly tied to user inputs in educational and career

trajectories, ensuring transparency and personalized guidance without a heavy reliance on predefined competency graphs or domain-specific knowledge.

Overall, while the systems in Table 3.1 showcase various strengths—ranging from market-oriented skill alignment to conversational explanations—XCRS differentiates itself by combining LLM embeddings with explainable recommendations across diverse educational and career roles. Its emphasis on user-centered explanations, coupled with access to multiple LLMs, offers a robust and transparent recommendation experience that can cater to varied user preferences and needs, situating it effectively within the evolving field of explainable AI-driven recommendation systems.

7.2. Scope and Limitations

The scope of XCRS focuses on providing course and career role recommendations primarily within the IT domain. This design choice was made to address the needs of a clearly defined target audience, including students and professionals in computer science, engineering, and related disciplines. All courses, roadmap data, and the application interface are in English, which allows for standardized content processing but may limit accessibility for non-English speakers. While this scope allows for a more tailored recommendation experience within a global IT audience, it also inherently limits the applicability of the system to users outside the IT field. Consequently, the system's performance and insights drawn from evaluations may not generalize to broader educational or career contexts, such as humanities, social sciences, or other domains.

One significant limitation of the system lies in the number of career roles covered and the reliance on community-created roadmaps. The career paths and steps in the roadmaps are largely user-generated, which can result in varying levels of accuracy and comprehensibility. Some roadmaps may not fully capture the current demands of certain roles or may contain steps that are unclear or potentially misleading to users. However, the open-source and community-supported nature of these roadmaps offers a unique advantage: as users contribute their experiences and knowledge, the content

continuously evolves and improves, much like other collaborative platforms. As a result, XCRS is expected to improve over time in tandem with the `roadmap.sh` project, benefiting from the ongoing refinement and enhancement of these career roadmaps.

From a technical perspective, while the system provides transparent explanations for recommendations, the current framework for explainability could be further refined to better cater to users with varying levels of domain knowledge. Additionally, the selection of LLMs within XCRS was guided by factors such as availability, computational efficiency, and the provision of APIs to generate embeddings. The embedding models used in XCRS are from some of the most prominent companies in the LLM field. However, this selection does not encompass all available models, and other models not included in this study might offer different recommendation quality or better alignment with diverse user needs. Future work could explore additional models to enhance recommendation performance and user satisfaction.

7.3. Threats to Validity

In evaluating the XCRS, several threats to validity must be considered to ensure that the findings are reliable, meaningful, and generalizable. To identify and categorize these threats, I followed the definitions provided by Wohlin et al [145]. This section addresses potential biases and limitations that could affect the integrity of the study's results. Where possible, strategies implemented to mitigate these threats are discussed, highlighting the steps taken to enhance the validity and robustness of the evaluation.

Internal Validity. The internal validity of this evaluation may be influenced by participants' familiarity with recommendation systems and their previous experiences with educational platforms. Such background knowledge could shape perceptions of XCRS's effectiveness and usability, introducing potential biases in survey responses. For instance, users with more experience using recommendation systems might possess different expectations or biases compared to those unfamiliar with similar tools. To mitigate this, a protocol was established to ensure consistent conditions for all participants

in both the user and expert judgment studies. I was present throughout the entire study, either face-to-face or online, to explain the system and address any questions according to the protocol. This ensured that all participants received the same level of information about the system and its usage, minimizing disparities in understanding.

Additionally, although the language of the user study and expert judgment study was English, the native language of all participants was not English, posing a potential threat to internal validity. However, all participants had graduated from universities where the language of instruction was English, and their education levels were sufficient to mitigate any significant comprehension issues during the evaluation process.

External Validity. The external validity of the study is limited by its focus on users primarily within the IT sector, which might constrain the generalizability of the results to other fields or populations. Participants were students and professionals in computer engineering and related disciplines, whose engagement with the recommendations may differ from users in non-technical fields. To address this limitation, efforts were made to include participants with varying educational backgrounds and professional experience within the IT domain, enhancing the representativeness of the feedback within this scope. However, to further strengthen external validity, future studies should aim to include participants from a broader range of fields, such as the humanities or social sciences. This would provide a more comprehensive assessment of XCRS's applicability across diverse user contexts.

Additionally, the roadmap data used in the XCRS was sourced from a community-contributed GitHub project, which presents another challenge to external validity. Since the data is crowd-sourced, its alignment with real-world career development needs may not always be precise. Nonetheless, the roadmap project chosen is one of the most highly regarded resources on GitHub, ranked as the 7th most-starred project out of 28 million, which lends credibility to its use. Its active community and widespread adoption help mitigate some concerns, though future iterations of XCRS could benefit from using more rigorously validated data sources to further improve the accuracy and

reliability of the recommendations.

Construct Validity. Construct validity is concerned with the alignment of evaluation metrics to the key constructs of interest, such as effectiveness, transparency, persuasiveness, satisfaction, and efficiency. While these metrics were carefully chosen to capture essential aspects of user experience, they may not fully encompass all factors influencing participants' decision-making in career and course selection. Additionally, experimenter expectancy could threaten construct validity by unintentionally shaping participant responses. To mitigate this, the consistent use of a protocol minimized unintentional communication, ensuring that participants did not receive cues or guidance that could influence their perceptions. Furthermore, open-ended questions in the survey allowed for qualitative feedback, providing richer insights into the user experience and uncovering additional factors that structured metrics may not have captured. This strategy supported construct validity by broadening the range of responses and maintaining neutrality throughout the feedback process.

Conclusion Validity. Conclusion validity concerns the degree to which the study's findings accurately reflect the relationships being investigated and the soundness of inferences drawn from the data. One potential threat arises from the limited statistical power of the study due to the relatively small number of participants, which may reduce the ability to detect subtle but meaningful differences in user responses across various dimensions. Additionally, while non-parametric tests like the Kruskal-Wallis H and Friedman tests were appropriately used to account for the non-normal distribution of survey data, their sensitivity might be limited when interpreting complex user feedback. Moreover, variations in participant backgrounds and their subjective interpretations of survey questions could introduce noise into the results, affecting the precision of conclusions. To address these concerns, the study emphasized transparent data analysis and careful interpretation of the statistical outcomes, recognizing the limitations inherent in small sample sizes and diverse user experiences. Future work can enhance conclusion validity by expanding the participant pool and refining survey instruments to ensure more precise and consistent measurement of constructs.

8. CONCLUSION

The research presented in this thesis introduced and evaluated the XCRS, a novel framework designed to provide both career role and course recommendations tailored to users in the IT sector. The system integrates state-of-the-art LLMs for embedding generation and employs an explainable recommendation approach to enhance user trust and understanding.

XCRS distinguishes itself by leveraging course data scraped from Udemy and structured roadmaps from `roadmap.sh`, encompassing a diverse range of IT roles and learning resources. This comprehensive dataset serves as a robust foundation for the recommendation system. The core components of XCRS—course and roadmap data collection, embedding generation, role recommendation, course recommendation, and explanation generation—were designed to provide personalized recommendations tailored to the user's background, preferences, and future learning interests.

The use of five advanced LLMs—Google's text-embedding-004, VoyageAI's voyage-large-2, OpenAI's text-embedding-3-large, MistralAI's mistral-embed, and Cohere's embed-english-v3.0—enabled a precise generation of embeddings for both user inputs and knowledge units. Cosine similarity was employed as the metric to match these embeddings effectively, providing accurate role and course recommendations. The system's explainability was achieved through a RAG approach, utilizing GPT-4o via ChatGPT to enhance the clarity and relevance of explanations provided for both career roles and courses.

The XCRS makes several key contributions to the field of educational and career guidance, supported by significant data points:

Development of XCRS. A unique course and career role recommendation system was developed specifically for the IT sector, utilizing LLM embeddings to ensure that rec-

ommendations are not only relevant but also explainable. The system successfully provides users with transparent justifications for both role and course suggestions, addressing a critical gap in current recommendation systems.

Evaluation of Multiple LLMs. Five different LLMs were employed to generate embeddings and subsequently evaluated for their performance in the role and course recommendations. The comparative analysis revealed that mistral-embed by MistralAI demonstrated superior performance in course recommendations with a usefulness score of 74.4%, while voyage-large-2 by VoyageAI was the top performer in career role recommendations, achieving a usefulness rating of 72.8%. Additionally, the model text-embedding-004 by Google was noted for its user satisfaction, indicating the importance of considering both usefulness and user preference when selecting an embedding model.

Creation of a Publicly Shared Dataset. The dataset created for this study was scraped from Udemy and filtered to contain 453 courses across the categories of Development, IT & Software, and Office Productivity. The original dataset of 8180 courses underwent cleansing and elimination to ensure relevance and quality, which improved the precision of the recommendations and contributed to the robustness of XCRS.

Comprehensive Evaluation through User and Expert Judgment Studies. A user study with participants from diverse IT-related backgrounds demonstrated high satisfaction levels in transparency (89.6%) and efficiency (88.8%), with effectiveness and persuasiveness scoring 80% and 76%, respectively. The study also revealed that users found the system serendipitous (84%) in guiding them toward new learning opportunities, while satisfaction ratings indicated potential areas for improvement in user engagement and likelihood to use (69.6%) and recommend (77.6%) the system. The expert judgment study confirmed these findings, with experts highlighting the system's accuracy, contextual relevance, and well-structured explanations. Feedback from domain experts suggested further refinement in the course selection process and the diversification of models to improve recommendation relevance for non-traditional profiles. The qualitative comments underscored the need to enhance the balance between back-

ground knowledge and personal interests, ensuring that recommendations are not only accurate but also aligned with diverse user profiles and goals.

Comparative performance of LLMs within XCRS was analyzed, revealing that mistral-embed by MistralAI performed best for course recommendations, achieving a usefulness score of 74.4%, while voyage-large-2 by VoyageAI was found most effective for career role recommendations with a usefulness score of 72.8%. However, user satisfaction highlighted text-embedding-004 by Google as a model preferred by users for overall experience, reflecting the balance between objective metrics and subjective user preferences in model performance.

In summary, the XCRS makes significant contributions to the field of educational and career recommendations. Offering explainable, transparent, and personalized recommendations enhances user experience and provides insights into role and course selection aligned with individual learning paths. The methodology and results indicate the effectiveness of LLM embeddings in delivering relevant content, and the use of RAG for generating contextually meaningful explanations adds value to the overall recommendation process. The evaluation of XCRS, through both quantitative metrics and qualitative feedback, underscores its potential to support users in navigating complex educational and career landscapes, while also providing a foundation for future work to refine and expand its capabilities.

8.1. Future Work and System Improvements

While the XCRS demonstrates strong performance and user satisfaction, several areas present opportunities for future improvements and expansion. One area is the expansion of the scope of recommendations. Currently, XCRS is focused on providing suggestions for courses and roles within the IT sector based on specific roadmap structures. Broadening the system to encompass recommendations for other professional domains and integrating a wider variety of knowledge sources would enhance its applicability to a more diverse range of user profiles. Additionally, refining the process

for role and course matching by incorporating additional contextual data, such as skill levels, current industry trends, or real-time job market demands, could improve the accuracy and relevance of recommendations. This is particularly important for users with less traditional career paths or those exploring emerging roles in the market.

Another avenue for enhancement involves user input categorization. The current model categorizes input based on whether users enjoy, feel neutral about, dislike, or are curious about specific concepts. Extending this framework to include additional dimensions, such as preferred learning styles (e.g., hands-on or theory-based courses) or available learning time, would enable even more precise and tailored recommendations. Furthermore, advancing explainability techniques remains a priority. While the present use of RAG with GPT-4o produces clear and meaningful explanations, exploring alternative methods like generating visual explanations or incorporating interactive feedback mechanisms could further improve user trust and the comprehensibility of the recommendations. Finally, conducting longitudinal user studies would provide valuable insights into the system's long-term impact. While the existing user and expert studies have yielded significant findings, observing how users' learning journeys and career paths evolve over extended periods based on the XCRS recommendations would deepen our understanding of its effectiveness and influence in real-world applications.

In conclusion, the XCRS demonstrates significant potential in bridging the gap between recommendation systems and explainability, offering users both actionable career and course guidance and transparent, personalized explanations. Its contributions to the IT educational field are reinforced by its robust dataset, comprehensive use of LLM embeddings, and extensive evaluation through user and expert studies. The findings from this work pave the way for future advancements in explainable AI and recommendation systems, fostering enhanced user trust, engagement, and empowerment in learning and career development.

REFERENCES

1. Isinkaye, F. O., Y. O. Folajimi and B. A. Ojokoh, "Recommendation Systems: Principles, Methods and Evaluation", *Egyptian Informatics Journal*, Vol. 16, No. 3, pp. 261–273, 2015.

2. Tinder, "Tinder: Dating, Make Friends & Meet New People", `https://tinder.com/`, accessed on September 25, 2024.

3. Shah, D., "By The Numbers: MOOCs in 2021", `https://www.classcentral.com/report/mooc-stats-2021/`, accessed on September 25, 2024.

4. Udemy, "Udemy: Online Courses - Learn Anything, On Your Schedule", `https://www.udemy.com/`, accessed on January 15, 2024.

5. Edx, "Edx: Free Online Courses by Harvard, MIT, & more", `https://www.edx.org/`, accessed on September 25, 2024.

6. Coursera, "Coursera: Online Courses & Credentials From Top Educators", `https://www.coursera.org/`, accessed on September 25, 2024.

7. Vo, N. N., N. H. Vu, T. A. Vu, Q. T. Vu and B. D. Mach, "CRS: a Hybrid Course Recommendation System for Software Engineering Education", *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*, pp. 62–68, Pittsburgh, PA, USA, 2022.

8. Khalid, A., K. Lundqvist and A. Yates, "Recommender Systems for MOOCs: A Systematic Lterature Survey (January 1, 2012–July 12, 2019)", *International Review of Research in Open and Distributed Learning*, Vol. 21, No. 4, pp. 255–291, 2020.

9. Ricci, F., L. Rokach and B. Shapira, "Recommender systems: Introduction and

Challenges", *Recommender Systems Handbook*, pp. 1–34, Springer, New York, NY, 2015.

10. Adomavicius, G. and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734–749, 2005.

11. Herlocker, J. L., J. A. Konstan, L. G. Terveen and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems", *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 5–53, 2004.

12. Su, X. and T. M. Khoshgoftaar, "A Survey of collaborative Filtering Techniques", *Advances in Artificial Intelligence*, Vol. 2009, No. 4, 2009.

13. Bobadilla, J., F. Ortega, A. Hernando and A. Gutièrrez, "Recommender Systems Survey", *Knowledge-Based Systems*, Vol. 46, pp. 109–132, 2013.

14. Lang, K., "NewsWeeder: Learning to Filter Netnews", *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, pp. 331–339, 1995.

15. Van Meteren, R. and M. Van Someren, "Using Content-Based Filtering for Recommendation", *Proceedings of the Machine Learning in the New Information Age*, Vol. 30, pp. 47–56, Barcelona, Catalonia, Spain, 2000.

16. Pazzani, M. J. and D. Billsus, "Content-based Rcommendation Systems", *The Adaptive Web: Methods and Strategies of Web Personalization*, Vol. 4321, pp. 325–341, Springer, 2007.

17. Colombo-Mendoza, L. O., R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández and J. J. Samper-Zapater, "RecomMetz: A Context-Aware Knowledge-Based Mobile Recommender System for Movie Showtimes", *Expert Systems with Applications*, Vol. 42, No. 3, pp. 1202–1222, 2015.

18. Liu, T., Q. Wu, L. Chang and T. Gu, "A Review of Deep Learning-Based Recommender System in E-Learning Environments", *Artificial Intelligence Review*, Vol. 55, No. 8, pp. 5953–5980, 2022.

19. Lepri, B., N. Oliver, E. Letouzè, A. Pentland and P. Vinck, "Fair, Transparent, and Accountable Algorithmic Decision-Making Processes: The Premise, the Proposed Solutions, and the Open Challenges", *Philosophy & Technology*, Vol. 31, No. 4, pp. 611–627, 2018.

20. Chazette, L., W. Brunotte and T. Speith, "Exploring Explainability: a Definition, a Model, and a Knowledge Catalogue", *29th International Requirements Engineering Conference*, pp. 197–208, Notre Dame, IN, USA, 2021.

21. McSherry, D., "Explanation in Recommender Systems", *Artificial Intelligence Review*, Vol. 24, No. 2, pp. 179–197, 2005.

22. Bilgic, M. and R. J. Mooney, "Explaining Recommendations: Satisfaction vs. Promotion", *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, pp. 13–18, San Diego, California, USA, 2005.

23. Herlocker, J. L., J. A. Konstan and J. Riedl, "Explaining Collaborative Filtering Recommendations", *Proceedings of the Conference on Computer Supported Cooperative Work*, pp. 241–250, Philadelphia, Pennsylvania, USA, 2000.

24. Wang, H., F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo, "Ripplenet: Propagating User Preferences on the Knowledge Graph for Recommender Systems", *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 417–426, Torino, Italy, 2018.

25. Chen, H., Y. Li, X. Sun, G. Xu and H. Yin, "Temporal Meta-Path Guided Explainable Recommendation", ArXiv:2101.01433 [cs], 2021.

26. Wang, X., X. He, Y. Cao, M. Liu and T.-S. Chua, "Kgat: Knowledge Ggraph Attention Network for Recommendation", *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958, Anchorage, AK, USA, 2019.

27. Xian, Y., Z. Fu, S. Muthukrishnan, G. de Melo and Y. Zhang, "Reinforcement Knowledge Graph Reasoning for Explainable Recommendation", *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 285–294, Paris, France, 2019.

28. Tintarev, N. and J. Masthoff, "A Survey of Explanations in Recommender Systems", *23rd International Conference on Data Engineering Workshop*, pp. 801–810, Istanbul, Turkey, 2007.

29. Zhang, Y., X. Chen *et al.*, "Explainable Recommendation: A Survey and New Perspectives", *Foundations and Trends® in Information Retrieval*, Vol. 14, No. 1, pp. 1–101, 2020.

30. Zhang, Y., G. Lai, M. Zhang, Y. Zhang, Y. Liu and S. Ma, "Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis", *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 83–92, Gold Coast, Queensland, Australia, 2014.

31. Peake, G. and J. Wang, "Explanation mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems", *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2060–2069, London, United Kingdom, 2018.

32. Radford, A., K. Narasimhan, T. Salimans and I. Sutskever, "Improving Language Understanding by Generative Pre-Training", 2018, `https://openai.com/index/language-unsupervised/`, accessed on September 25, 2024.

33. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding", ArXiv:1810.04805 [cs], 2018.

34. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Weinberger (Editors), *Advances in Neural Information Processing Systems*, Vol. 26, pp. 3111–3119, Lake Tahoe, Nevada, United States, 2013.

35. Mohanty, I., "Recommendation Systems in the Era of LLMs", *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 142–144, Panjim, India, 2024.

36. Wu, L., Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu *et al.*, "A Survey on Large Language Models for Recommendation", *World Wide Web*, Vol. 27, No. 60, 2024.

37. Zhao, Z., W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang and Q. Li, "Recommender Systems in the Era of Large Language Models", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 11, pp. 6889–6907, 2024.

38. Chen, B., X. Dai, H. Guo, W. Guo, W. Liu, Y. Liu, J. Qin, R. Tang, Y. Wang, C. Wu *et al.*, "All Roads Lead to Rome: Unveiling the Trajectory of Recommender Systems Across the LLM Era", ArXiv:2407.10081 [cs], 2024.

39. Wang, N., H. Wang, Y. Jia and Y. Yin, "Explainable Recommendation via Multi-Task Learning in Opinionated Text Data", *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 165–174, Ann Arbor, MI, USA, 2018.

40. Chen, C., M. Zhang, Y. Liu and S. Ma, "Neural Attentional Rating Regression with Review-level Explanations", *Proceedings of the World Wide Web Conference*, pp. 1583–1592, Lyon, France, 2018.

41. Wang, X., X. He, F. Feng, L. Nie and T.-S. Chua, "TEM: Tree-Enhanced Embedding Model for Explainable Recommendation", *Proceedings of the World Wide Web Conference*, pp. 1543–1552, Lyon, France, 2018.

42. Wang, X., Y. Chen, J. Yang, L. Wu, Z. Wu and X. Xie, "A Reinforcement Learning Framework for Explainable Recommendation", *2018 IEEE International Conference on Data Mining*, pp. 587–596, Singapore, Singapore, 2018.

43. Li, L., Y. Zhang and L. Chen, "Personalized Transformer for Explainable Recommendation", *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4947–4957, Online, 2021.

44. Ni, J., J. Li and J. McAuley, "Justifying Recommendations Using Distantly-Labeled Reviews and Fine-Grained Aspects", *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 188–197, Hong Kong, China, 2019.

45. Li, L., Y. Zhang and L. Chen, "Generate Neural Template Explanations for Recommendation", *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, p. 755–764, Virtual Event, Ireland, 2020.

46. Morsomme, R. and S. V. Alferez, "Content-Based Course Recommender System for Liberal Arts Education", *International Educational Data Mining Society*, 2019.

47. Esteban, A., A. Zafra and C. Romero, "A Hybrid Multi-Criteria Approach Using

a Genetic Algorithm for Recommending Courses to University Students", *Proceedings of the 11th International Conference on Educational Data Mining*, pp. 273–279, Buffalo, NY, USA, 2018.

48. Neamah, A. A. and A. S. El-Ameer, "Design and Evaluation of a Course Recommender System Using Content-Based approach", *International Conference on Advanced Science and Engineering*, pp. 1–6, Duhok, Iraq, 2018.

49. Ma, B., Y. Taniguchi and S. Konomi, "Course Recommendation for University Environments", *International Educational Data Mining Society*, 2020.

50. Ray, S. and A. Sharma, "A Collaborative Filtering Based Approach for Recommending Elective Courses", *Information Intelligence, Systems, Technology and Management: 5th International Conference*, pp. 330–339, Gurgaon, India, 2011.

51. Houbraken, M., C. Sun, E. Smirnov and K. Driessens, "Discovering Hidden Course Requirements and Student Competences from Grade Data", *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 147–152, Bratislava, Slovakia, 2017.

52. Elbadrawy, A. and G. Karypis, "Domain-Aware Grade Prediction and Top-N Course Recommendation", *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 183–190, Boston, MA, USA, 2016.

53. Bakhshinategh, B., G. Spanakis, O. Zaiane and S. ElAtia, "A Course Recommender System Based on Graduating Attributes", *Proceedings of the 9th International Conference on Computer Supported Education*, Vol. 2, pp. 347–354, Porto, Portugal, 2017.

54. Esteban, A., A. Zafra and C. Romero, "Helping University Students to Choose Elective Courses by Using a Hybrid Multi-Criteria Recommendation System with Genetic Optimization", *Knowledge-Based Systems*, Vol. 194, No. 105385, 2020.

55. Morsy, S. and G. Karypis, "Will This Course Increase or Decrease Your GPA? Towards Grade-Aware Course Recommendation", ArXiv:1904.11798 [cs], 2019.

56. Ibrahim, M. E., Y. Yang, D. L. Ndzi, G. Yang and M. Al-Maliki, "Ontology-Based Personalized Course Recommendation Framework", *IEEE Access*, Vol. 7, No. 99, pp. 5180–5199, 2018.

57. Gulzar, Z., A. A. Leema and G. Deepak, "PCRS: Personalized Course Recommender System Based on Hybrid Approach", *Procedia Computer Science*, Vol. 125, pp. 518–524, 2018.

58. Jiang, W., Z. A. Pardos and Q. Wei, "Goal-Based Course Recommendation", *Proceedings of the 99th International Conference on Learning Analytics & Knowledge*, pp. 36–45, 2019.

59. Pardos, Z. A., Z. Fan and W. Jiang, "Connectionist Recommendation in the Wild: On the Utility and Scrutability of Neural Networks for Personalized Course Guidance", *User Modeling and User-Adapted Interaction*, Vol. 29, No. 2, pp. 487–525, 2019.

60. Wong, C., "Sequence Based Course Recommender for Personalized Curriculum Planning", *Artificial Intelligence in Education: 19th International Conference*, pp. 531–534, London, UK, 2018.

61. Shao, E., S. Guo and Z. A. Pardos, "Degree Planning with PLAN-BERT: Multi-Semester Recommendation Using Future Courses of Interest", *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 14920–14929, Virtual Event, 2021.

62. Frej, J., A. Dai, S. Montariol, A. Bosselut and T. Käser, "Course Recommender Systems Need to Consider the Job Market", *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*,

pp. 522–532, Washington, DC, USA, 2024.

63. Alario-Hoyos, C., D. Leony, I. Estèvez-Ayres, M. Pèrez-Sanagustìn, I. Gutièrrez-Rojas and C. D. Kloos, "Adaptive Planner for Facilitating the Management of Tasks in MOOCs", *V International Conference on Quality and Accessibility in eLearning*, pp. 517–522, Antigua Guatemala, Guatemala, 2014.

64. Popescu, I., K. Portelli, C. Anagnostopoulos and N. Ntarmos, "The Case for Graph-Based Recommendations", *IEEE International Conference on Big Data*, pp. 4819–4821, Boston, MA, USA, 2017.

65. Zhou, M., A. Cliff, S. Krishnan, B. Nonnecke, C. Crittenden, K. Uchino and K. Goldberg, "M-CAFE 1.0: Motivating and Prioritizing Ongoing Student Feedback During MOOCs and Large on-Campus Courses Using Collaborative Filtering", *Proceedings of the 16th Annual Conference on Information Technology Education*, pp. 153–158, Chicago, Illinois, USA, 2015.

66. De Medio, C., F. Gasparetti, C. Limongelli, M. Lombardi, A. Marani, F. Sciarrone and M. Temperini, "Towards a Characterization of Educational Material: An Analysis of Coursera Resources", T.-T. Wu, R. Gennari, Y.-M. Huang, H. Xie and Y. Cao (Editors), *Emerging Technologies for Education*, pp. 547–557, Rome, Italy, 2017.

67. Samin, H. and T. Azim, "Knowledge Based Recommender System for Academia Using Machine Learning: A Case Study on Higher Education Landscape of Pakistan", *IEEE Access*, Vol. 7, pp. 67081–67093, 2019.

68. Labarthe, H., F. Bouchet, R. Bachelet and K. Yacef, "Does a Peer Recommender Foster Students' Engagement in MOOCs?", *International Conference on Educational Data Mining*, pp. 418–423, Raleigh, NC, USA, 2016.

69. Bouchet, F., H. Labarthe, K. Yacef and R. Bachelet, "Comparing Peer Recom-

mendation Strategies in a MOOC", *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pp. 129–134, Bratislava, Slovakia, 2017.

70. Prabhakar, S., G. Spanakis and O. Zaiane, "Reciprocal Recommender System for Learners in Massive Open Online Courses (MOOCs)", *International Conference on Web-Based Learning*, pp. 157–167, Cape Town, South Africa, 2017.

71. Potts, B. A., H. Khosravi, C. Reidsema, A. Bakharia, M. Belonogoff and M. Fleming, "Reciprocal Peer Recommendation for Learning Purposes", *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pp. 226–235, Sydney, New South Wales, Australia, 2018.

72. Cohen, R., N. Sardana, K. Rahim, D. Y. Lam, M. Li, O. Maccarthy, E. Woo, J. Zhang and G. Guo, "Recommending Messages to Users in Social Networks: A Cross-Site Study", *12th International Conference on Machine Learning and Applications*, Vol. 2, pp. 445–450, Miami, Florida, USA, 2013.

73. Yang, D., M. Piergallini, I. Howley and C. Rose, "Forum Thread Recommendation for Massive Open Online Courses", *Proceedings of the 7th International Conference on Educational Data Mining*, pp. 257–260, London, UK, 2014.

74. Jo, Y., G. Tomar, O. Ferschke, C. P. Rosè and D. Gasevic, "Expediting Support for Social Learning with Behavior Modeling", ArXiv:1605.02836 [cs], 2016.

75. Yang, D., D. Adamson and C. P. Rosè, "Question Recommendation with Constraints for Massive Open Online Courses", *Proceedings of the 8th ACM Conference on Recommender Systems*, pp. 49–56, Silicon Valley, California, USA, 2014.

76. Yang, D., J. Shang and C. P. Rosè, "Constrained Question Recommendation in MOOCs via Submodularity", *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1987–1990,

Shanghai, China, 2014.

77. Macina, J., I. Srba, J. J. Williams and M. Bielikova, "Educational Question Routing in Online Student Communities", *Proceedings of the 11th ACM Conference on Recommender Systems*, pp. 47–55, Como, Italy, 2017.

78. Tan, M. and M. Wu, "An Association Rule Model of Course Recommendation in MOOCs: Based on edX Platform", *European Scientific Journal*, Vol. 14, No. 25, pp. 284–292, 2018.

79. Atalla, S., M. Daradkeh, A. Gawanmeh, H. Khalil, W. Mansoor, S. Miniaoui and Y. Himeur, "An Intelligent Recommendation System for Automating Academic Advising Based on Curriculum Analysis and Performance Modeling", *Mathematics*, Vol. 11, No. 5, pp. 1–25, 2023.

80. Zhang, H., T. Huang, Z. Lv, S. Liu and Z. Zhou, "MCRS: A Course Recommendation System for MOOCs", *Multimedia Tools and Applications*, Vol. 77, No. 6, pp. 7051–7069, 2018.

81. Yin, S., K. Yang and H. Wang, "A MOOC Courses Recommendation System Based on Learning Behaviours", *Proceedings of the ACM Turing Celebration Conference*, pp. 133–137, Hefei, China, 2020.

82. Tzeng, J.-W., N.-F. Huang, A.-C. Chuang, T.-W. Huang and H.-Y. Chang, "Massive Open Online Course Recommendation System Based on a Reinforcement Learning Algorithm", *Neural Computing and Applications*, pp. 1–12, 2023.

83. Symeonidis, P. and D. Malakoudis, "MoocRec.com : Massive Open Online Courses Recommender System", *Proceedings of the Poster Track of the 10th ACM Conference on Recommender Systems*, Vol. 1688, Boston, USA, 2016.

84. Gupta, A. and D. Garg, "Applying Data Mining Techniques in Job Recommender System for Considering Candidate Job Preferences", *2014 International Confer-*

ence on Advances in Computing, Communications and Informatics, pp. 1458–1465, Delhi, India, 2014.

85. Malinowski, J., T. Keim, O. Wendt and T. Weitzel, "Matching People and Jobs: A Bilateral Recommendation Approach", *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Vol. 6, pp. 137c–137c, Kauai, Hawaii, 2006.

86. Qamhieh, M., H. Sammaneh and M. N. Demaidi, "PCRS: Personalized Career-path Recommender System for Engineering Students", *IEEE Access*, Vol. 8, pp. 214039–214049, 2020.

87. Dave, V. S., B. Zhang, M. Al Hasan, K. AlJadda and M. Korayem, "A Combined Representation Learning Approach for Better Job and Skill Recommendation", *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1997–2005, Torino, Italy, 2018.

88. Striebel, J., R. Myers and X. Liu, "Career-Based Explainable Course Recommendation", I. Sserwanga, A. Goulding, H. Moulaison-Sandy, J. T. Du, A. L. Soares, V. Hessami and R. D. Frank (Editors), *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity: 18th International Conference, Virtual Event*, pp. 371–382, 2023.

89. Ghosh, A., B. Woolf, S. Zilberstein and A. Lan, "Skill-Based Career Path Modeling and Recommendation", *IEEE International Conference on Big Data*, pp. 1156–1165, Virtual Event, 2020.

90. Kokkodis, M. and P. G. Ipeirotis, "Demand-Aware Career Path Recommendations: a Reinforcement Learning Approachh", *Management Science*, Vol. 67, No. 7, pp. 4362–4383, 2021.

91. Jiang, L., K. Liu, Y. Wang, D. Wang, P. Wang, Y. Fu and M. Yin, "Reinforced

Explainable Knowledge Concept Recommendation in MOOCs", *ACM Transactions on Intelligent Systems and Technology*, Vol. 14, No. 3, pp. 1–20, 2023.

92. Sarwar, B., G. Karypis, J. Konstan and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms", *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, Hong Kong, Hong Kong, 2001.

93. Rendle, S., C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback", *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, Montreal, Quebec, Canada, 2009.

94. Mnih, A. and R. R. Salakhutdinov, "Probabilistic Matrix Factorization", *Advances in Neural Information Processing Systems*, Vol. 20, pp. 1257–1264, Vancouver, British Columbia, Canada, 2007.

95. Salakhutdinov, R. and A. Mnih, "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo", *Proceedings of the 25th International Conference on Machine learning*, pp. 880–887, Helsinki, Finland, 2008.

96. Song, W., Z. Duan, Z. Yang, H. Zhu, M. Zhang and J. Tang, "Ekar: An Explainable Method for Knowledge Aware Recommendation", ArXiv:1906.09506 [cs], 2022.

97. Mnih, V., "Playing Atari with Deep Reinforcement Learning", ArXiv:1312.5602 [cs], 2013.

98. Schulman, J., F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal Policy Optimization Algorithms", ArXiv:1707.06347 [cs], 2017.

99. Decorte, J.-J., J. Van Hautte, T. Demeester and C. Develder, "JobBERT: Understanding Job Titles through Skills", ArXiv:2109.09605 [cs], 2021.

100. Abu-Rasheed, H., M. H. Abdulsalam, C. Weber and M. Fathi, "Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational Explainability and Mentoring", ArXiv:2401.08517 [cs], 2024.

101. Yang, Y., C. Zhang, X. Song, Z. Dong, H. Zhu and W. Li, "Contextualized Knowledge Graph Embedding for Explainable Talent Training Course Recommendation", *ACM Transactions on Information Systems*, Vol. 42, No. 2, pp. 1–27, 2023.

102. Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data", C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Weinberger (Editors), *Advances in Neural Information Processing Systems*, Vol. 26, pp. 2787–2795, 2013.

103. Wang, H., M. Zhao, X. Xie, W. Li and M. Guo, "Knowledge Graph Convolutional Networks for Recommender Systems", *The World Wide Web Conference*, pp. 3307–3313, 2019.

104. Chau, H. K., *Explainable Course Recommendation: Connecting College Education to Knowledge and Careers Through Skills*, Ph.D. Thesis, University of Pittsburgh, 2023.

105. Reimers, N. and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, Hong Kong, China, 2019.

106. Zylowski, T., N. Sautchuk Patricio, W. Hettmann, K. Anderer, M. Wölfel and P. Henning, "Evaluating Trustworthiness, Usability and Explainability of an Educational Pathway Recommendation System that Uses a Large Language Model", *Proceedings of 16th International Conference on Education and New Learning*

Technologies, pp. 6461–6470, Palma, Spain, 2024.

107. Wang, C., H. Zhu, P. Wang, C. Zhu, X. Zhang, E. Chen and H. Xiong, "Personalized and Explainable Employee Training Course Recommendations: A Bayesian Variational Approach", *ACM Transactions on Information Systems*, Vol. 40, No. 4, pp. 1–32, 2021.

108. Hu, Y., Y. Koren and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets", *2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, Pisa, Italy, 2008.

109. Wang, C. and D. M. Blei, "Collaborative Topic Modeling for Recommending Scientific Articles", *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 448–456, San Diego, California, 2011.

110. He, X., L. Liao, H. Zhang, L. Nie, X. Hu and T.-S. Chua, "Neural Collaborative Filtering", *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, Perth, Australia, 2017.

111. Xue, H.-J., X. Dai, J. Zhang, S. Huang and J. Chen, "Deep Matrix Factorization Models for Recommender Systems.", *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Vol. 17, pp. 3203–3209, Melbourne, Australia, 2017.

112. He, X., K. Deng, X. Wang, Y. Li, Y. Zhang and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation", *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, Virtual Event, China, 2020.

113. Wang, H., N. Wang and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems", *Proceedings of the 21th ACM SIGKDD International Con-*

ference on Knowledge Discovery and Data Mining, pp. 1235–1244, Sydney, Australia, 2015.

114. Li, X. and J. She, "Collaborative Variational Autoencoder for Recommender Systems", *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 305–314, Halifax, NS, Canada, 2017.

115. Ahmed, K., "Developer Roadmaps", `https://roadmap.sh/`, accessed on January 10, 2024.

116. Selenium, "The Selenium Browser Automation Project", `https://www.selenium.dev/documentation/`, accessed on October 23, 2024.

117. Muennighoff, N., N. Tazi, L. Magne and N. Reimers, "MTEB: Massive Text Embedding Benchmark", ArXiv:2210.07316 [cs], 2022.

118. Google, "Google Generative AI Overview", `https://ai.google/discover/generativeai/`, accessed on October 23, 2024.

119. VoyageAI, "VoyageAI: Supercharging Search and Retrieval for Unstructured Data", `https://www.voyageai.com/`, accessed on October 23, 2024.

120. OpenAI, "OpenAI", `https://openai.com/`, accessed on October 23, 2024.

121. MistralAI, "Mistral AI — Frontier AI in Your Hands", `https://mistral.ai/`, accessed on October 23, 2024.

122. Cohere, "Cohere — The Leading AI Platform for Enterprise", `https://cohere.com/`, accessed on October 23, 2024.

123. Singhal, A., "Modern Information Retrieval: A Brief Overview", *IEEE Data Engineering Bulletin*, Vol. 24, No. 4, pp. 35–43, 2001.

124. Ontañón, S., "An Overview of Distance and Similarity Functions for Structured Data", *Artificial Intelligence Review*, Vol. 53, No. 7, pp. 5309–5351, 2020.

125. Jaccard, P., "Etude de la Distribution Florale Dans Une Portion des Alpes et du Jura", *Bulletin de la Societe Vaudoise des Sciences Naturelles*, Vol. 37, No. 142, pp. 547–579, 1901.

126. Hamming, R. W., *Coding and Information Theory*, Prentice-Hall, Englewood Cliffs, 1980.

127. Bookstein, A., V. A. Kulyukin and T. Raita, "Generalized Hamming Distance", *Information Retrieval*, Vol. 5, No. 4, pp. 353–375, 2002.

128. Freestone, M. and S. K. K. Santu, "Word Embeddings Revisited: Do LLMs Offer Something New?", ArXiv:2402.11094 [cs] 2024.

129. Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin (Editors), *Advances in Neural Information Processing Systems*, Vol. 33, pp. 9459–9474, Virtual Event, 2020.

130. OpenAI, "GPT-4 Technical Report", ArXiv:2303.08774 [cs], 2024.

131. Nuxt, "Nuxt: The Intuitive Vue Framework", `https://nuxt.com/`, accessed on October 23, 2024.

132. Vue.js, "Vue.js - The Progressive JavaScript Framework", `https://vuejs.org/`, accessed on October 23, 2024.

133. Bierman, G., M. Abadi and M. Torgersen, "Understanding Typescript", *European Conference on Object-Oriented Programming*, pp. 257–281, Uppsala, Sweden, 2014.

134. Van Rossum, G. and F. L. Drake, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, 2009.

135. Anaconda, "Anaconda Software Distribution", `https://docs.anaconda.com/`, accessed on October 23, 2024.

136. Ramírez, S., "FastAPI", `https://fastapi.tiangolo.com/`, accessed on October 23, 2024.

137. Encode, "Uvicorn: An ASGI web server, for Python.", `https://www.uvicorn.org/`, accessed on October 23, 2024.

138. Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, "Array Programming with NumPy", *Nature*, Vol. 585, No. 7825, pp. 357–362, 2020.

139. Reback, J., W. McKinney, jbrockmendel, J. V. den Bossche, T. Augspurger, P. Cloud, gfyoung, Sinhrks, A. Klein, M. Roeschke, J. Tratner, C. She, W. Ayd, S. Hawkins, T. Petersen, J. Schendel, A. Hayden, M. Garcia, V. Jancauskas, MomIsBestFriend, P. Battiston, S. Seabold, chris b1, h vetinari, S. Hoyer, W. Overmeire, alimcmaster1, M. Mehyar, C. Whelan and T. Kluyver, "Data structures for Statistical Computing in Python", *Proceedings of the 9th Python in Science Conference*, Vol. 445, pp. 51–56, Austin, TX, 2010.

140. Shaphiro, S. and M. Wilk, "An Analysis of Variance Test for Normality", *Biometrika*, Vol. 52, No. 3, pp. 591–611, 1965.

141. Kruskal, W. H. and W. A. Wallis, "Use of Ranks in One-Criterion Variance Analysis", *Journal of the American Statistical Association*, Vol. 47, No. 260, pp. 583–

621, 1952.

142. Friedman, M., "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance", *Journal of the American Statistical Association*, Vol. 32, No. 200, pp. 675–701, 1937.

143. Likert, R., "A Technique for the Measurement of Attitudes", *Archives of Psychology*, Vol. 22, No. 140, pp. 5–55, 1932.

144. Sileo, D., W. Vossen and R. Raymaekers, "Zero-Shot Recommendation as Language Modeling", *European Conference on Information Retrieval*, pp. 223–230, Stavanger, Norway, 2022.

145. Wohlin, C., P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, *Experimentation in Software Engineering*, Springer, Berlin, 2012.

146. Ma, X. and L. Ye, "Career Goal-Based E-Learning Recommendation Using Enhanced Collaborative Filtering and Prefixspan", *International Journal of Mobile and Blended Learning*, Vol. 10, No. 3, pp. 23–37, 2018.

147. Han, J., M. Kamber and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Boston, third edn., 2011.

APPENDIX A: USER STUDY SURVEY QUESTIONS

Demographic questions:

- What is your current education status and highest continuing education degree?
- Is the department you are studying/graduated at university computer engineering, computer science, or a similar department (management information systems, electrical and electronics eng., industrial eng., mathematics, etc.)?
- Current employment status:
- Do you work in the IT sector?
- Year of job experience:

Model-related questions:

- How useful are Model 1's recommendations regarding career roles?
- How useful are Model 1's recommendations regarding courses?
- How useful are Model 1's recommendations regarding career roles?
- How useful are Model 1's recommendations regarding courses?
- How useful are Model 2's recommendations regarding career roles?
- How useful are Model 2's recommendations regarding courses?
- How useful are Model 3's recommendations regarding career roles?
- How useful are Model 3's recommendations regarding courses?
- How useful are Model 4's recommendations regarding career roles?
- How useful are Model 4's recommendations regarding courses?
- How useful are Model 5's recommendations regarding career roles?
- How useful are Model 5's recommendations regarding courses?
- Which model's recommendations were you most satisfied with?

System performance evaluation questions:

- How satisfied are you with the relevance of the role and course recommendations? (Effectiveness)
- How likely are you to consider a course that was recommended to you? (Persuasiveness)
- How easy do you find the explanations provided for the recommendations to understand? (Transparency)
- How easy is it to navigate and use the system to get the recommendations you need? (Efficiency)
- Do you feel that the system helps you explore courses outside of your usual preferences? (Serendipity)
- How likely are you going to use the system to help with your career plan? (Satisfaction - Likelihood to Use)
- How likely are you going to recommend the system to your peers? (Satisfaction - Likelihood to Recommend)
- Considering the entire recommendation experience, is there anything you would like to share?

APPENDIX B: EXPERT JUDGMENT STUDY SURVEY QUESTIONS

For each three persona:

- Which model's recommendations were you most satisfied with for the persona?
- How relevant do you find the recommended roles for the persona's knowledge, considering only known concepts and completed courses, excluding personal interests?
- How relevant do you find the recommended courses for the persona's knowledge, considering only known concepts and completed courses, excluding personal interests?
- How accurate do you find the recommended roles for the persona when considering both their background and interests?
- How accurate do you find the recommended courses for the persona when considering both their background and interests?
- How clear and understandable are the explanations provided for the recommended roles for the persona?
- How clear and understandable are the explanations provided for the recommended courses for the persona?
- How effective do you find the system overall in recommending suitable roles and courses for the persona?

For the overall experience:

- Which models' recommendations were you most satisfied with in overall? (up to two models)
- Can you describe how relevant you find the recommended roles and courses for the persona's knowledge when considering only known concepts and completed courses, and excluding personal interests?

- In your opinion, how accurate are the recommended roles and courses for the persona when considering both their background and interests? Please explain your reasoning.
- How would you describe the quality of the explanations provided for the recommended roles and courses for each persona? Are they clear and sufficient?
- How effective do you believe the system is in recommending suitable roles and courses for each persona? Please provide specific examples or observations.
- What improvements would you suggest for enhancing the role and course recommendation aspects of the system?
- How do the recommendations from our system compare with other recommendation systems you have used or evaluated?
- Do you think any particular model among the five provided more reliable recommendations? If so, which one and why?