Automatic Instructional Feedback for Database Courses in Higher Education: Strategies for Structured Learning Engagement and Mediation

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc. Chukwuka Victor Obionwu

geb. am 20.02.1980 in Enugu

Gutachterinnen/Gutachter

Prof. Dr. rer. nat. habil. Gunter Saake
Prof. Dr.-Ing. Alke Martens
Prof. Dr. rer. nat. Andreas Thor

Magdeburg, den 30.08.2024

Obionwu, Chukwuka Victor:
Automatic Instructional Feedback for Database Courses in Higher Education: Strategies for Structured Learning Engagement and Mediation
DISSERTATION, Otto-von-Guericke University Magdeburg, 2024.

Abstract

The pedagogical effectiveness of an instructional technique can be assessed by its capacity to adapt to the dynamics of interaction that arise from students' learning engagements. This adaptability allows the instruction strategy to cater to the individual needs and learning styles of students, promoting a more personalized and effective learning experience. Additionally, an instruction strategy that can adapt to interaction dynamics fosters a collaborative and interactive learning environment, enhancing student engagement and participation. However, as observed in most course projects and exercise groups, most students' learning engagement events occur in the context of an egocentric interaction and are thus difficult to classify as structured or unstructured in the short run. This difficulty highlights the importance of implementing an instructional strategy that can readily measure and evaluate students' learning progress and engagement. This will allow instructors or an automated system to identify individual students strengths or areas where they may need additional support or guidance. To address this challenge, researchers have developed and implemented strategies based on natural language processing, large language models, recommendation systems, an error class strategy, and a method that enables a retrospective understanding of students' engagement during a programming task. State-of-the-art research efforts indicate that the implementation of automated systems for the early evaluation and restructuring of a student's learning engagement easily addresses academic failure and thus boosts the ability of lecturers to provide timely and proactive interventions with minimal effort. Similarly, this thesis's efforts have the potential to enhance overall learning engagement outcomes. Furthermore, the implementation of the strategies described in this research, either in their entirety or in a modified form, has the potential to enhance academic performance prediction and promote the development of teamwork skills. Furthermore, this approach lays the foundation for the successful implementation of an intelligent agent-mediated learning platform. While the strategies employed and discussed in this work are generalizable to most fields of learning, their current focus is on teaching structured query language.

Zusammenfassung

Die pädagogische Wirksamkeit einer Unterrichtstechnik kann anhand ihrer Fähigkeit bewertet werden, sich an die Interaktionsdynamik anzupassen, die aus den Lernaktivitäten der Schüler entsteht. Diese Anpassungsfähigkeit ermöglicht es der Unterrichtsstrategie, auf die individuellen Bedürfnisse und Lernstile der Schüler einzugehen, was ein personalisierteres und effektiveres Lernerlebnis fördert. Zusätzlich fördert eine Unterrichtsstrategie, die sich an die Interaktionsdynamik anpassen kann, ein kollaboratives und interaktives Lernumfeld, das das Engagement und die Teilnahme der Schüler erhöht. Wie jedoch in den meisten Kursprojekten und Übungsgruppen beobachtet, finden die Lernengagement-Ereignisse der meisten Studierenden im Kontext einer egozentrischen Interaktion statt und sind daher kurzfristig schwer als strukturiert oder unstrukturiert zu klassifizieren. Diese Schwierigkeit unterstreicht die Bedeutung der Implementierung einer Unterrichtsstrategie, die den Lernfortschritt und das Engagement der Schüler leicht messen und bewerten kann. Dies wird es Lehrkräften oder einem automatisierten System ermöglichen, die individuellen Stärken der Schüler oder Bereiche zu identifizieren, in denen sie zusätzliche Unterstützung oder Anleitung benötigen. Um diese Herausforderung zu bewältigen, haben Forscher Strategien entwickelt und implementiert, die auf natürlicher Sprachverarbeitung, großen Sprachmodellen, Empfehlungssystemen, einer Fehlerklassifikationsstrategie und einer Methode basieren, die ein rückblickendes Verständnis des Engagements der Schüler während einer Programmieraufgabe ermöglicht. Modernste Forschungsbemühungen zeigen, dass die Implementierung automatisierter Systeme zur frühen Bewertung und Umstrukturierung des Lernengagements eines Schülers akademische Misserfolge leicht angeht und somit die Fähigkeit der Dozenten stärkt, rechtzeitige und proaktive Interventionen mit minimalem Aufwand bereitzustellen. Ähnlich haben die Bemühungen dieser Dissertation das Potenzial, die allgemeinen Ergebnisse des Lernengagements zu verbessern. Darüber hinaus hat die Umsetzung der in dieser Forschung beschriebenen Strategien, sei es in ihrer Gesamtheit oder in modifizierter Form, das Potenzial, die Vorhersage der akademischen Leistung zu verbessern und die Entwicklung von Teamarbeitfähigkeiten zu fördern. Darüber hinaus legt dieser Ansatz die Grundlage für die erfolgreiche Implementierung einer intelligenten, agentenvermittelten Lernplattform. Während die in dieser Arbeit angewandten und diskutierten Strategien auf die meisten Lernfelder übertragbar sind, liegt der aktuelle Schwerpunkt auf dem Unterricht in Structured Query Language.

Extended Abstract

Context: In conventional educational settings, educators familiarize themselves with various students and their respective competencies through ongoing interactions during the students' sturdy engagement. Over time, the strategies employed by students for interaction undergo transformations, and, in some instances, instructors exhibit limited adaptability in revising their preconceived notions about a student's level of knowledge. Despite the fact that interactions between students and human teachers provide valuable insights into student objectives, abilities, drive, and preferences, teachers often overlook these interactions due to their commitment to other activities, such as research endeavors. Therefore, human instructors may struggle to adjust to the current circumstances of the students they are teaching. In contrast to human tutors, intelligent tutors possess the capability to deduce and retain presumed student knowledge within the student model. Thus, it becomes obvious that there is a need for a pedagogically effective strategy that facilitates automatic adaptation with respect to a student's current objective and engagement. The aspiration is for a system, a pedagogical agent, to respond efficiently, motivate and stimulate students' interest, and thus facilitate the acquisition of given knowledge through instructional feedback.

Instructional feedback encompasses the provision of information or direction to individuals or learners with the aim of aiding them in improving their performance, cultivating their abilities, or enhancing their understanding of a particular subject, task, or issue. The utilization of this strategy holds significant potential and possesses inherent value in the facilitation of learning and growth within training, educational, and other learning environments. Given the unpredictable nature of a student's engagement event, it becomes important to recognize the need for a persistent system that can provide human-like instructional feedback.

Method: To effectively tackle these challenges, techniques based on natural language processing, large language models, recommendation systems, etc. have been adopted to develop and implement systems that address collaboration, instructional feedback, and retrospective provenance evaluation challenges. These techniques have proven to be successful in improving collaboration among users, providing valuable instructional feedback to learners, and evaluating the quality and reliability of information sources. Additionally, the integration of machine learning algorithms has further enhanced the accuracy and efficiency of these systems, making them indispensable tools in various domains such as education, research, and online platforms.

This thesis contributes to the following four major areas:

Collaborative Learning Area. Here, we introduced the learning interaction hierarchy, which affords a method of characterizing and modeling forms for learning engagements. This hierarchy allows us to understand the different levels of interactions that take place during individual learning and team interaction scenarios, which arise during course projects that require teams of students to work together. By categorizing and modeling these forms of engagement, we can gain insights into how learners interact with the course content, instructors, and their peers. This understanding can help in designing effective learning experiences and improving educational outcomes. In this thesis, we contribute a strategy for administering team collaboration, a platform that facilitates it, and a strategy to mediate between two collaboration systems.

Learning Analytic Area. Here, we introduced the error class strategy, a method that allows us to gain a retrospective understanding of students' engagement while solving structured query language tasks and the various problems they encounter during their learning process. By categorizing errors into different classes, we can identify patterns and trends that can inform our teaching strategies and interventions. This retrospective approach not only helps us address individual student needs but also provides valuable insights for curriculum development and instructional improvement. Furthermore, by understanding the types of errors students commonly make, we can tailor our instruction to target those areas and enhance their learning experience. In this area, we contribute the error class strategy and learning analytic dashboard, which utilize the error classes as indicators to both visualize and make sense of students' learning engagements.

Recommendation System Area. Collaboration is a vital component of university education as students unite to pursue shared goals, such as acquiring knowledge in certain subjects or engaging in team projects and group assignments. Collaboration not only fosters academic growth but also helps students develop important skills such as communication, problem-solving, and teamwork. Additionally, successful collaborations can lead to long-lasting professional relationships and networking opportunities that can benefit students in their future careers. Conversely, a failed collaboration not only fails to achieve these goals but also negatively impacts future partnerships. In this research area, we contribute an approach that leverages multiplex partitioning to create and recommend collaboration teams of desired sizes. Furthermore, we contribute a strategy for providing meaningful instructional feedback in the form of slide recommendations during individual online exercise sessions.

Automatic Instructional Feedback. Several research efforts have demonstrated that feedback is a significant factor in enhancing and attaining essential educational objectives, promoting student engagement, and aiding students in sustaining motivation. The primary objective of instructional feedback is to furnish learners with comprehensive information on their knowledge or performance, enabling them to make pertinent enhancements and adjustments. Additionally, it can motivate and encourage students to persevere in their efforts. Students typically receive timely instructional feedback shortly after submitting or completing a task. Immediately following the feedback, learners can compare their recent experiences and actions. This is challenging in both traditional learning environments and online settings. Therefore, we have developed a methodology for the automated evaluation and recommendation of relevant lecture slides in this field. We also implemented a platform that provides insight into student study interaction activities.

Conclusion: In general, the application of learning analytic methodologies can effectively address a variety of learning difficulties. We adopted a strategy that involves analyzing the unique characteristics and engagement levels of students, allowing us to identify those who struggle academically and identify potential contributing factors. The implementation of automated systems for the early evaluation and restructuring of a student's learning engagement potentially addresses study-related academic difficulties. This also boosts the ability of lecturers to provide timely and proactive interventions with minimal effort. This, in turn, has the potential to ultimately improve overall university outcomes. I argued that the implementation of the strategies described in this research, either in their entirety or in a modified form, has the potential to enhance academic performance prediction and promote the development of teamwork skills. Furthermore, this approach lays the foundation for the successful implementation of personalized learning systems.

Keywords: Skill Acquisition, Collaborative Platforms, Team Assessment Strategy, Text Mining, Text Clustering, Instructional Feedback, Learning Analytic, Natural language processing, knowledge extraction, Conversational agents, Technology-Enhanced Learning, Web classroom applications, Social network analysis, Recommendation systems, Pre-trained models, Collaboration in teams, Community detection algorithms, Recommendation system, Dashboards, Text mining, Page ranking, Large language models.

Statement of Authorship

I hereby certify that I have prepared this thesis without the unauthorized assistance of third parties and without the use of resources other than those indicated; external and my own sources used are identified as such. In particular, I have not used the help of a commercial doctoral advisor. Third parties have neither directly nor indirectly received monetary benefits from me for work related to the content of the submitted dissertation. In particular, I did not knowingly:

- - invent any results or conceal contradictory results
- -intentionally misuse statistical procedures to interpret data in an unjustified manner
- - plagiarize other people's results or publications
- -distort the results of other research.

I am aware that violations of copyright law may give rise to injunctive relief and claims for damages by the author as well as criminal prosecution by the law enforcing authorities. The work has not yet been submitted as a dissertation in the same or a similar form either in Germany or abroad and has not yet been published as a whole.

Place, Date Signature

Magdeburg, 21.02.2024 Obionwu Chukwuka Victor

Contents

List of Figures

List of Tables

1. Introduction

The pedagogical effectiveness of an instruction strategy from the perspective of an instructor consists of an evaluation of the course design, content, and, most importantly, its effectiveness in facilitating knowledge acquisition and retention Tomlinson [2008]. For most students enrolled in courses where knowledge of a specific programming language is a requirement, operational competence and skill acquisition while interacting with the course concepts and exercise tasks are major concerns. Frické [2009]; Prasad et al. [2022]. This concern arises because students need to not only understand the theoretical concepts but also be able to apply them practically in programming tasks. Therefore, assessing the pedagogical effectiveness of an instruction strategy should also include an evaluation of students' ability to apply their knowledge in real-world programming scenarios. Balyer and Özcan [2020]; King et al. [2009]; Sinclair et al. [2020]. Instructional feedback is an agency that instructors use to provide students with specific pedagogical guidance on learning techniques, suggest improvements to enhance their programming skills, and gauge the students' understanding of a concept. It entails the delivery of information, guidance, and assessment to the learners to help them understand their performance and thus make improvements. Lipnevich and Smith [2018]; Narciss [2012]. Furthermore, instructional feedback plays a crucial role by encouraging self-regulation, inspiring learners, and, most importantly, nurturing the acquisition of a structured problem engagement strategy. A known problem-enggement strategy is the scientific method, which is a methodical, empirical, controlled, and analytical approach to examining presumed relationships between natural phenomena. Portney and Watkins [2020]. When faced with a problem in their area of expertise, a person who has mastered the scientific method first tries to categorize it into subclasses for which established solutions or methods of analysis exist. A person who has mastered the scientific method selects and applies the appropriate method if the problem optimally matches any of the known subclasses. If the problem remains unclassifiable or does not fit into any existing class, the individual develops a new strategy that incorporates the general problem-solving techniques of the domain. The method schema is updated with the new strategy to address similar problems. Also, if a previously effective method fails, an evaluation needs to be conducted. Using the result of the evaluation, the method schema is further updated to accommodate the new problem. Furthermore, an understanding of why a strategy failed aids in classifying a given problem Mitrovic [2003]; Obionwu et al. [2022b]; Sussman [1973]. The end goal of an instructional feedback strategy is such that a student's problem engagement strategy evolves and conforms to the just described scientific method or similar strategy. An alternative to a structured form of engagement involves trial and error and a significant amount of stress. While it is challenging for every student in a traditional lecture scenario to acquire the scientific method of problem engagement, e-learning platforms provide affordances such as prompt feedback on assignments, course progress, and tests. Thus, learners could potentially remain engaged and committed to their studies Berge [2002]. This work has developed strategies that integrate automatic instructional feedback into e-learning platforms, utilizing natural language processing, language modeling, recommendation systems, and error class detection methods. The interaction of these strategies and the rapid evolution of the AI models they employ make the eventual goal of creating an intelligent agent-mediated learning platform achievable. This platform would provide personalized recommendations and adaptive learning experiences based on individual needs and preferences. It would also enable continuous feedback loops between learners and AI agents, fostering a dynamic and interactive learning environment.

1.1 Research Contributions

In this section, we detail the contributions of this thesis. An overview of the respective challenges, objectives, and strategies taken to address them is given, as they will be well elaborated in the respective chapters. For each of these contributions, a comprehensive literature review on the topic, which provides a solid foundation for further research in the field, was carried out. In the coming chapters, we will also offer deductive insights into the subject matter. The contributions of this thesis are spread into the 4 broad areas as discussed below

1.1.1 Learning Analytic Area

The contributions described in this research area are based on these publications Obionwu et al. [2022b, 2023b,c, 2021a].

Description: The adoption of e-learning pedagogy in educational institutions has become increasingly prevalent, and it has proffered solutions to notable deficiencies that existed in the traditional learning pedagogy. However, it has also opened up new challenges. One such challenge is how to gain retrospective insight into students' learning engagements. A strategy we adopted is the use of error classes. Thus, we are able to understand the extent to which our teaching strategy is effective in reducing errors resulting from exercise task engagement and other learning challenges that may affect knowledge acquisition by individual students in the course of their learning interactions. Additionally, our evaluation revealed that students learning SQL are most likely to encounter syntax errors. Based on this insight, we implemented a tour that guides students while they engage with our system. The tour provides step-by-step instructions on how to avoid common syntax errors and offers helpful tips for troubleshooting. This interactive feature has greatly improved students' understanding and confidence in using SQL, leading to more successful learning outcomes. Furthermore, we have observed that the tour has also increased students' overall engagement and motivation in the course, as they feel supported and empowered to overcome challenges in their learning journey. To ascertain the effectiveness of these strategies, we further implemented a dashboard. The dashboard allows us to track students' progress and performance in real-time, providing valuable insights into their usage of the tour and their overall comprehension of SQL concepts. Additionally, it enables us to identify any areas where students may be struggling or in need of additional support, allowing us to tailor our instruction and interventions accordingly. Overall, the combination of the error class strategy, the tour, and the dashboard has proven to be powerful tools in enhancing student learning and success in SQL. Listed below are our main contributions in this area:

- **Contribution 1:** We conceptualized an intervention technique through a tutorial walk-through that familiarizes students with our learning management system and how to resolve prevalent errors, such as syntax errors, while engaging with their respective exercise tasks.
- **Contribution 2:** We implemented a strategy that affords instructors valuable insights about students' learning progress in courses focused on the structured query language.
- **Contribution 3:** We implemented a dashboard that allows us and students to track study progress and performance in real-time, providing valuable insights into the overall comprehension of SQL concepts.

1.1.2 Collaborative Learning Area

The contributions described in this research area are based on these publications Obionwu et al. [2022a,c,d, 2023f].

Collaborative skills in the workplace are essential for effective teamwork and communication. Therefore, designing activities or projects that foster this skill is crucial. We aim to contribute valuable insights into how collaboration can effectively enhance student performance and knowledge acquisition during SQL task engagement by investigating the design of collaborative project tasks. Furthermore, the findings from this research will strengthen the existing evidence base that supports the positive impact of collaboration on learning and task completion in the context of SQL training courses. To this end, we have designed collaborative tasks and a platform for team collaboration. We have carefully designed the collaborative tasks to foster active participation and knowledge sharing among team members. The platform for team collaboration provides a user-friendly interface that facilitates seamless communication and coordination between team members, allowing them to work together efficiently on SQL tasks. Additionally, we collected data on the performance and satisfaction of participants to evaluate the effectiveness of the collaborative approach and identify areas for improvement in future iterations of the platform. We further observed that teams rarely used the internal chat system. This behavior leads to scattered conversations and difficulty tracking important discussions and decisions. Thus, the unavailability of the interaction data reduces insights into the dynamics of a team's collaboration and how they came about the solutions they present for respective projects. To this end, we have further designed and implemented a mediation strategy between SQLValidator and Telegram, using APIs and Webhooks. Thus, we are able to fetch telegram group messages into and from SQLValidator. Listed below are our main contributions in this area:

- **Contribution 1:** We implemented a platform that supports teamwork and further designed collaborative tasks for SQL teaching courses.
- **Contribution 2:** We implemented a mediation strategy between the SQLValidator system and the Telegram application to enhance collaboration.

1.1.3 Recommendation System Area

The contributions described in this research area are based on these publications Obionwu et al. [2023e, 2022e,g, 2023g, 2024].

The possibility of modeling and abstracting interaction has been the key driver in social network-based research. By understanding how users interact within a social network, researchers can develop algorithms and models that accurately predict user preferences and behaviors. This enables the creation of personalized recommendations, which enhances the user experience and engagement on social platforms. Additionally, modeling interaction patterns can also provide valuable insights into the dynamics of online communities and the spread of information within them. Based on these interaction models, which we derived from student studies, learning-focused engagements, and personality surveys, we investigated the correlation between academic performance and personality traits. The results of the investigation contributed to the development of a utility function that can forecast academic achievement by utilizing student profiles constructed through the implementation of the Big 5 personality model. In order to provide them with an optimal team recommendation, we have devised a strategy that involves further assessing their collaborative effectiveness through individual personality questionnaires. Additionally, we employ community detection using the Leiden algorithm. This algorithm helps identify clusters of students with similar personality traits, allowing us to form teams that have a higher likelihood of working well together. By combining the insights from individual personality questionnaires and community detection, we aim to enhance the overall collaboration and maximize academic performance among students. Listed below are our main contributions in this area:

- **Contribution 1:** Introduces the learner network interaction hierarchy and characterizes the various interaction modeling forms in learner-centered social networks.
- **Contribution 2:** Propose an approach that utilizes models capable of precisely eliciting an individual's personality to facilitate the formation of collaborative teams.

1.1.4 Automatic Instructional Feedback

.

The contributions described in this research area are based on these publications Obionwu et al. [2023a,d, 2021b, 2022b].

An important strategy for reducing academic-related stress during online academic engagements is the integration of real-time instructional feedback. Ensuring the seamless integration of instructional feedback into the existing learning management system poses a major challenge, as it requires careful consideration of compatibility with the platform's infrastructure. Also, it is necessary to take into account the pedagogical objective of the learning engagement activity. Additionally, it is crucial to continuously evaluate and refine the instructional feedback system based on previous feedback offered to respective users and carry out performance evaluations to ensure its effectiveness in enhancing the learning experience. While it is possible to hard-code these requirements, no two students are the same. Thus, it is important to also consider the individual needs and preferences of each student when designing the instructional feedback system. Incorporating personalized features, such as adaptive algorithms, will enable students to receive feedback in a way that best suits their learning style and pace. By taking into account these unique characteristics, the instructional feedback system can truly cater to the diverse needs of students and enhance their overall learning outcomes. An agency for the actualization of such a system is an intelligent conversational agent. The conversational agent demonstrates intelligence by analyzing and interpreting student responses, offering tailored feedback, and modifying its approach according to each student's study engagement progress. This conversational agent can also offer real-time assistance and support, making the learning experience more interactive and engaging. With the integration of an intelligent conversational agent, the instructional feedback system can provide a personalized and dynamic learning environment that maximizes knowledge acquisition and student success. Furthermore, by integrating such agents into our learning management system, we aim to create a supportive and interactive environment that promotes effective learning and reduces the burden of academic-related stress. We are currently progressing in this area.

Listed below are our contributions in this area:

- **Contribution 1:** A strategy for the provision of meaningful instructional feedback during individual online exercise sessions by leveraging the similarity between structured query language (SQL) theory with corresponding exercise tasks and respective SQL keyword analysis.
- **Contribution 2:** A strategy for using conversational agents as an agency for the provision of personalized instructional feedback.

1.2 Research Framework

The SQLValidator platform served as the research platform for this dissertation. SQLValidator, as shown in figure 1.1, is a web-based interactive tool for learning and practicing SQL. In the SQLValidator environment, students can, among other activities, form and test their queries against a database and receive immediate feedback.

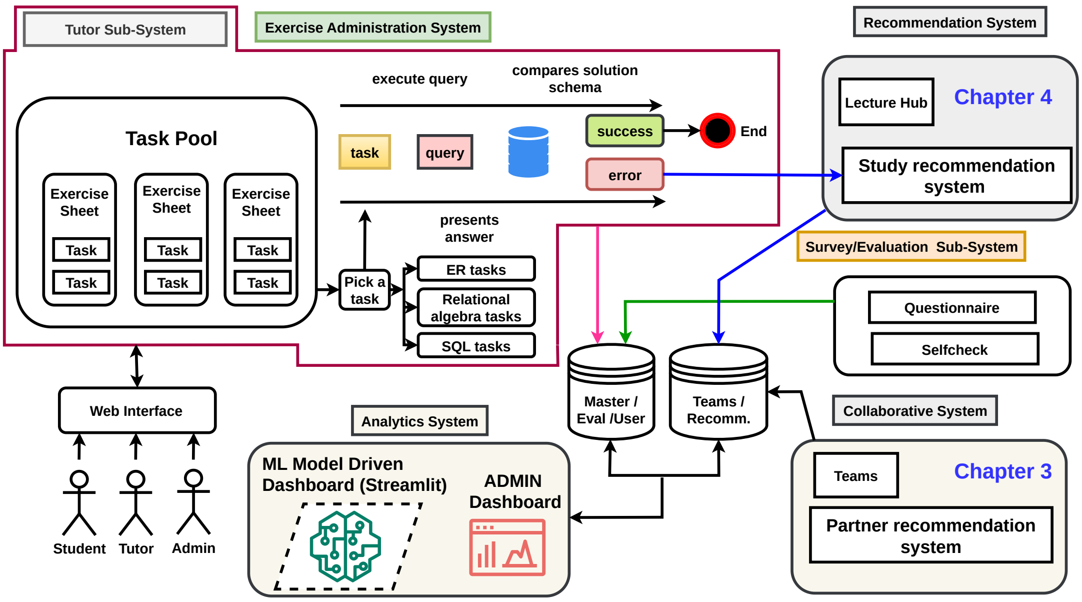


Figure 1.1: The SQLValidator

Current use cases for SQLValidator fall into the following categories:

- **Personal Study:** Students with access to the SQLValidator can freely explore the inbuilt database with tasks from the exercises to prepare for exams or for other purposes.
- **Course Exercises:** The database lecture in Magdeburg comes with several exercise tasks. We administer these exercises using SQLValidator. In this use case, students have the freedom to keep fixing errors in their queries until they achieve the desired outcome. Also, they receive the expected schema and informative feedback during the trial and submission process.
- **Self Checks:** The purpose of these tests is to assist students in evaluating their comprehension of the topics already discussed in the database concept course. Students can repeatedly take the self-check test until they are satisfied. These self-checks primarily use standardized and multiple-choice questions. Additionally, we incorporate tests for SQL query skills that are relevant to evaluating the concepts learned in the course.
- **Questionnaires:** Throughout the semester, we encourage students to assess their learning experience during the course or their interaction with the tools we use to administer the course exercises, such as the SQLValidator system. To administer these evaluations, we directly use our SQLValidator survey subsystem.

1.3 Expected limitations

This research possesses certain limitations that are beyond the researcher's control. The limitations of the research are stipulated constraints Theofanidis and Fountouki

[2018]. The study's limitations stem from its methodology, design, and analysis methods, as noted by Myers et al. Myers et al. [2013]. The study's limitations encompass:

1. The study utilized a quantitative methodology. The selection of technique entails the process of quantifying and measuring a mental phenomenon by attributing qualities to it Wilson [2013]. The quantification of these characteristics is based on self-disclosure. Hence, the study is constrained by the participant's capacity to express their subjective mental state in relation to the quantifiable variables being assessed Wilson [2013].
2. The study utilized convenience sampling. Convenience sampling is a sampling method that selects individuals who are easily accessible without the use of randomization, as noted by Viglia et al. Viglia et al. [2021]. Given their simple accessibility, the participants may exhibit a clustering of shared interests or characteristics that sets them apart from the larger community in a distinctive way. Individuals who possess a higher level of proficiency in utilizing internet technologies are more inclined to participate in the study through online means. The presence of an online affinity creates a bias in the sample and restricts the generalizability of the results to people who are less proficient in using online technology De Quidt et al. [2019]; Sugden [2005].
3. The fact that participation in this study is voluntary suggests that participants' motivation for participating is personal interest in the subject. The informed consent process reveals all relevant information about the study. The proposed study's disclosures encompass the utilization of "chat systems and conversational agents" in conversations and statistical analysis, as well as the identification of two possible causes for sample bias. Initially, individuals who have a strong liking for "chat systems and conversational agents" may be more inclined to voluntarily participate in the study, whereas those who strongly dislike "chat systems and conversational agents" may be more inclined to decline participation. Furthermore, individuals who have a strong inclination towards mathematics may exhibit a higher probability of enrolling in the study, whereas those who have a strong dislike or avoidance of mathematics may exhibit a lower probability. If either of the sample bias situations occurs, it limits the applicability of the results to other situations.
4. The study employed statistical analysis to facilitate understanding the results. The statistical analysis utilizes the properties of probability to generate levels of assurance regarding real outcomes vs. errors. Hence, probability laws restrict research findings as universally applicable facts for the overall population Zyphur and Pierides [2020a,b].

1.4 Organization of the Thesis

In this research endeavor, we have developed technologies that enable the early assessment and reorganization of a student's learning involvement to tackle academic underachievement. This, in turn, enhances the lecturers' capacity to deliver prompt and proactive interventions with minimal effort. We organize the remaining chapters as follows: Chapter 1 gives an overview of the thesis, the contributions, and the thesis organization. Chapter 2 provides comprehensive information on collaboration models and emphasizes the crucial considerations inside the problem space while ensuring instructional efficacy. Chapter 3 elaborates on the structuring and mediation of team collaborations, the development of team tasks, and the conceptualization of a partner recommendation system. In Chapter 4, we describe our learning analytic strategy and our instructional feedback intervention strategy, which focus on improving the learning experience. In chapter 5, we will provide a comprehensive summary of the conclusions and thesis.

2. General Theoretical Background

Social network models are mathematical representations of the relationships and interactions between individuals within a social network. These models aim to capture the patterns and dynamics of social connections, allowing researchers to study various phenomena such as information diffusion, opinion formation, and community detection. By analyzing these models, researchers can gain insights into how social networks evolve over time and how they influence individual behavior and collective outcomes. In this chapter, we present the theoretical background, which consists of the students' network interaction hierarchy Section 2.2, the different network models that can be elicited from their interactions Section 2.3, and a summary of the insights gained from the chapter.

2.1 Model of Interest

Interaction, in its broadest sense, refers to situations in which an individual consciously reorganizes and influences the behaviors of another individual and vice versa. These interactions and behaviors form the basis of a social structure and are therefore fundamental objects of social inquiry and analysis Gillett [2021]; Memon et al. [2015]; Turner [1988]. A social networking platform acts as a new dimension to the traditional social interaction process. These interactions as observed in social networks are no different from the three-way handshake Cerf and Kahn [1974]; Gopalan and Selvan [2008]; Peterson and Davie [2007], i.e., the algorithm used by the Transmission Control Protocol to establish and terminate a connection in the internet. Message exchanges embody the processes involved in establishing and terminating a connection with an entity in a network or interaction. The name, address, and potentially the location of all artifacts or another actor requiring interaction are the most crucial components of these messages. This structure necessitates the knowledge and sharing of each participating element's identity, availability, location, and integrity information for any interaction to occur. Any lack or absence of these elements will lead to a breakdown in communication. Based on this context, one can define interaction as a multi-path relationship between two or more nodes in order to achieve an objective. The two component parts that are needed for these

multi-path relationships to form in a social network are the network structures, which are basically interaction graphs, and the profile of a node Lin et al. [2019].
A network consists of nodes that symbolize actors, and each node is connected to other nodes by interaction edges. We refer to the paths in the network as edges, which represent social interactions like friendship or project collaboration. In the next subsections, we will provide a comprehensive explanation of graphs and their significant characteristics in order to enhance comprehension.

2.1.1 Graphs

A graph G is an ordered pair such that it consists of a set V of vertices and a set E of edges. The vertices represent the entities or objects, while the edges represent the relationships or connections between these entities Wasserman and Faust [1994]. This will be the definition that is followed throughout this chapter. It was also used in the corresponding papers Obionwu et al. [2022e]. The graph structure allows for the representation and analysis of various types of data, such as social networks, student learning interaction networks, etc.

$$\mathbf{G} = (\mathbf{V}(\mathbf{G}), \mathbf{E}(\mathbf{G})) \tag{2.1}$$

where V is a set, whose elements are called vertices or nodes

$$\mathbf{E} \subseteq \{\{\mathbf{x}, \mathbf{y}\} |\ \mathbf{x}, \mathbf{y} \in \mathbf{V} \langle \mathbf{x} \neq \mathbf{y} \rangle\} \tag{2.2}$$

and E, a set of edges which are unordered pairs Brandes [2005].
Another graph **H** is a sub-graph if

$$\mathbf{H} \subseteq \mathbf{G} \Longleftrightarrow \mathbf{V}(\mathbf{G})| \wedge \mathbf{E}(\mathbf{H}) \subseteq \mathbf{E}(\mathbf{G}) \tag{2.3}$$

So, H is a sub-graph of G if and only if the vertex set of H is a subset of the vertex set of G and the edges set of H is a subset of the edges set of G. Fig. 2.1 shows a sample graph and a sub-graph. As can be seen from the two graphs, the vertices

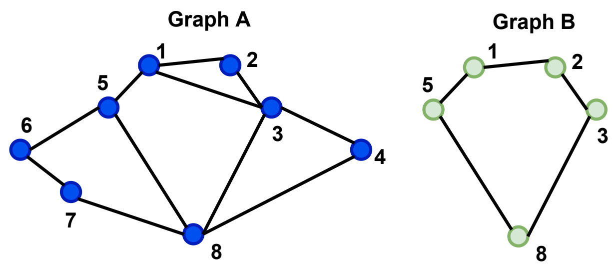


Figure 2.1: Graph and Subgraph

1, 2, 3, 4, 5 and the corresponding edges of Graph B correspond to the vertices 1, 3, 4, 8, 5 and the corresponding edges in Graph A. Hence, Graph B is a subgraph of Graph A. These graphs can be either undirected or directed. For example, we can characterize Facebook's network structure as an undirected graph because its friendship structure is bidirectional, meaning that Alice and Bob's friendship is equivalent to Bob and Alice's friendship. Conversely, we can describe Twitter as a

directed graph, where Alice can follow Bob without Bob following Alice. Directed graphs or digraphs are a set of ”nodes” and a set of directed ”lines” or ”edges” connecting pairs of nodes. We will denote the number of nodes in a digraph by ”g,” the group size. This type of graph is better represented with an (n*n) square matrix $\mathcal{X}^m$,called the adjacency matrix. Given a directed interaction involving a group of 6 nodes or individuals, the square matrix $\mathcal{X}^m$, shown in Fig. 2.2 A, is used to represent the associated interactions. In $\mathcal{X}^m$, $\mathcal{X}^m{}_{\mathrm{ij}}$ designates the status of the relationship between node i to node j. Using a binary representation, we indicate the presence of a tie with “1” and its absence thereof with a “0”. After the establishment of a tie, the frequency of interaction can be indicated using ordinal numbers Gossen et al. [2014]; Harary [1962].
Furthermore, in Fig. 2.2 B, the direction of the tie is represented using a sociogram. The arrowheads indicate the direction of the ties. An arrow pointing from node 5 to node 1 indicates a tie from node 5 to node 1. This is also seen through the sociomatrix: $\mathcal{X}^m_{51} = 1$, but $\mathcal{X}^m_{15} = 0$. The profile information annotates the established paths with details that inform on the attributes of the node and neighbor nodes Akcora et al. [2011, 2013]; Raad et al. [2010]. In the next subsection, we describe the node profile.

A

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	0	0	1	0	0	0
3	0	1	0	0	0	0
4	0	0	0	1	1	0
5	1	0	0	0	0	0
6	1	0	0	0	0	0

B

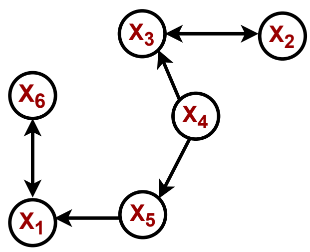


Figure 2.2: Sociogram with six nodes

2.1.2 Node Profile

A node’s profile describes the significant characteristics of the individual it represents. In social networks, these characteristics consist of information about the interactions, behavior, connections, opinions, etc. We model these characteristics within a graph as properties associated with the node. We model the profile information of a node or individual as a set of properties.

$$\mathbf{P} = \{\mathbf{P}_1, ..., \mathbf{P}_{\mathrm{n}}\} \tag{2.4}$$

These properties as identified in Wasserman and Faust [1994] come in several forms: demographic properties, such as age, gender, and location; properties that represent political or religious convictions; properties that encode activities, hobbies, and affiliations; and many other aspects that capture an individual’s preferences Yang et al. [2014]; Zhang et al. [2017]. In offline social networks, it has been observed that nodes with similar profiles, such as those who share similar hobbies, attend the same lectures, or hold similar convictions, tend to interact with each other. For network measures such as centrality, diversity, and density, please refer to Kennedy et al. [2015]; Wasserman and Faust [1994]. We can infer four types of networks from these interactions: the ego, the dyad, the triad, and networks of arbitrary size. These networks form a hierarchy, which we discuss in the next section.

2.2 Network Interaction Hierarchy

This survey presents a classification of these hierarchies. Based on Arnaboldi et al. [2012]; Faust [2010]; Sutcliffe et al. [2012]; Wasserman and Faust [1994], our hierarchy includes four main classes of networks or social units, namely ego networks, dyadic networks, triadic networks, and networks of arbitrary size. As illustrated in Fig. 2.3, these roughly form a hierarchy where networks of arbitrary size are the widest, i.e., most general type, whereas ego networks have the most specific domain. As we move up the hierarchy, the interactions become more restricted, thereby reducing the number of independently varying parameters in the interaction. This constrained degree of freedom allows for simple interaction modeling and instrumentation.

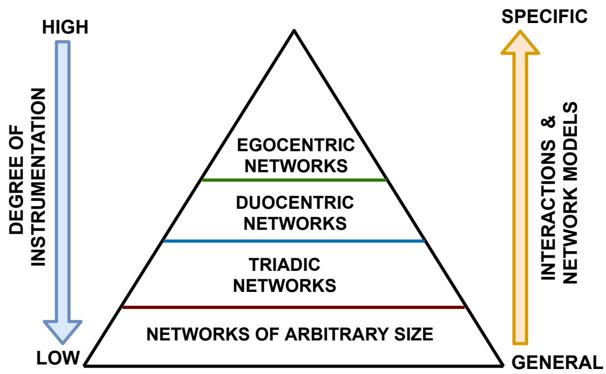


Figure 2.3: Interaction Hierarchy

In the next sections, we will describe the important features of each network in the interaction hierarchy, and the respective models associated with them.

2.3 Learner-Centered Networks

Learner-centered networks prioritize the needs and interests of learners, aiming to create an environment where learners can actively engage in their own learning process, collaborate with peers, and access resources that support their individual learning goals. In this section, we will discuss ego networks, dyad networks, triad networks, and scale-free networks.

2.3.1 Egocentric Networks

Egocentric networks are social networks consisting of a single node, the ego, together with other nodes, the alters, that they interact with, and all the interaction links among those alters. This network can also be described as the neighborhood networks or first-order neighborhoods of an ego Sutcliffe et al. [2012]. The size and degree of ego networks allow for a straightforward analysis of the processes that affect larger networks. Also, the strategies used in sample surveys and most of the techniques

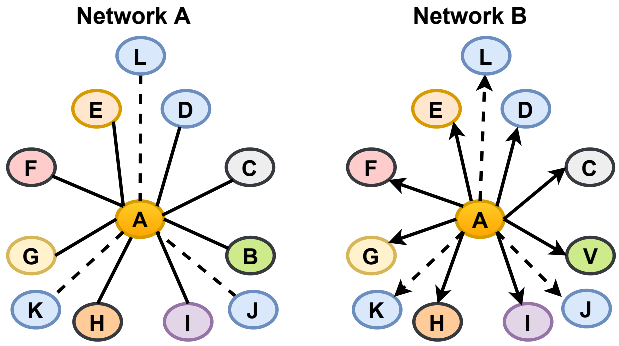


Figure 2.4: Ego network

employed in social network analysis and modeling are fully compatible with ego networks Arnaboldi et al. [2016]; Carolan [2013]; Crossley et al. [2015b]; Tabassum et al. [2018]. As shown in Fig. 2.4, the vertices represent the alters, and the edges represent their connection between the ego and other alters. The ego is at the center of the network, which can be either undirected as in network A or directed as in network B.The arrow direction in network B shows the direction of information flow. Weak ties between the alters, indicated by dotted lines, are also present in an egonetwork. The number of alters present in an ego's network determines the ego's degree, which indicates the size of the network. The alters may be independent or tied, in which case they form a clique. These tied alters increase the likelihood of a consensus during decision-making activities. These ties also give rise to transitive relations, which are conducive to cooperation and the development of trust. The next subsection discusses the features of an egocentric network.

2.3.1.1 Features of an Egocentric Network

Anthropological research that examined the relationships that make up a person's personal social world gave rise to the ego network structure. Also In Forgas et al. [2011]; Hill and Dunbar [2003]; Sutcliffe et al. [2012], the limit for maintaining social relationships was described using a series of concentric circles of acquaintanceship that scales with a consistent ratio close to 3 Arnaboldi et al. [2013]; Guidi [2015]. As we show in Fig. 2.5, the circles, called dunbar circles, represent a hierarchical arrangement of alters with the cumulative sizes of consecutive groups following a scaling ratio of approximately 3. This hierarchy is based on the increasing level of intimacy between these alters and the ego Dunbar et al. [2014]; Dunbar [2008]; Guidi et al. [2021].

The innermost circle represents the support clique consisting of 4–5 individuals. The group of individuals interacts with each other and shares similar interests. They not only identify with one another but are often bound together by shared social characteristics such as ethnicity and socioeconomic status Arnaboldi et al. [2012]; Guidi [2015]. Most individuals devote 40% of their social time to members of this group Dunbar and Spoors [1995]; Sutcliffe et al. [2012]. The sympathy group consists of beneficiaries, enablers, or neighbors. They are made up of 12–15 individuals, and group activities are mostly directed to a specific utility such as survival, income, successful study outcome, recreation, etc. The beneficiaries within this group are individuals with a direct interest in a specific utility. The efficiency of this group relies on the individuals with facilitation skills, known as enablers, who convene regularly and foster trust to guarantee the completion of a specific task. The affinity group, comprising approximately 50 individuals, comes next. They usually consist of friends, colleagues, or extended family members Freeman et al. [1979]. The dynamics of this group remain an open research challenge due to the difficulties associated with the manual collection of data about its members through interviews or surveys. The final circle, known as the active network, consists of 150 individuals, including the alters of the inner rings. This number is the cognitive limit on the number of individuals that we can know as persons—that is, those with whom we have a defined personal relationship Dunbar [1993]. The interaction between the ego and people in the active network occurs at least once a year. For every individual within this circle, the ego earnestly invests time and other resources to maintain the related social relationship Dunbar and Spoors [1995]; Sutcliffe et al. [2012].

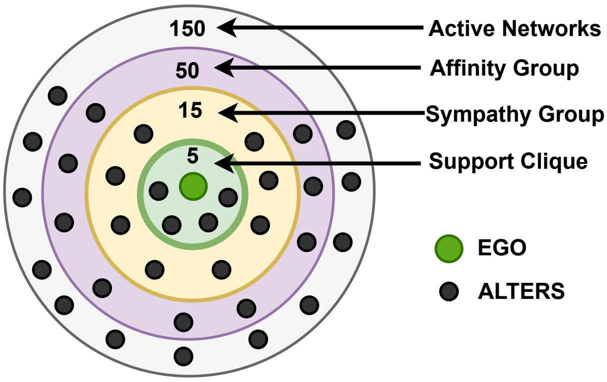


Figure 2.5: Dunbar Circles Arnaboldi et al. [2017]; Guidi [2015]

Apart from individuals within the dunbar circle, it is also possible that the ego's activity is affected by important alters that reside in a different domain, such as a workplace. Compared to other network models, the ego model's well-defined structure enhances the instrumentation for interaction modeling and facilitates the elicitation of information. We examine various metrics such as ego-alter similarity, density, effective network size, betweenness centrality, and structural hole to gain a deeper understanding of the structure and interactions within an egocentric network. We will describe these measures in more detail below.

2.3.1.2 Ego-Alter Similarity

Network measures based on ego-alter similarity make it easier to figure out how egos and alters interact with each other, how much egos can affect their alters, and how people like to interact with each other. One of the mechanisms of ego-alter similarity is homophily, which is the tendency for people to have non-negative ties Currarini et al. [2016]; Liébana-Presa et al. [2018]; McPherson et al. [2001], i.e., ties with individuals who are similar to themselves in a socially significant way Lou et al. [2013]. This insulates the ego from external influence and reinforces in-group behavior and biases. We use the E-I index to measure an ego's propensity to have ties to alters similar to them. The E-I index is the number of ties external to the groups minus the number of ties that are internal to the group divided by the total number of ties Borgatti [2011]; Crossley et al. [2015a]; DeJordy and Halgin [2008]; Hinds and McGrath [2006].

$$\mathbf{E} - \mathbf{I} = \frac{\mathbf{N}_{\text{external}} - \mathbf{N}_{\text{internal}}}{\textbf{network size}} \tag{2.5}$$

One can view the E-I index, which ranges from 1 to -1, as a gauge of a group member's degree of affiliation with its own group. A value of -1 indicates homophily, and a value of +1 indicates heterophily.

2.3.1.3 Ego-Alter and Alter-Alter Tie Attribute

The information about the nature of ties between the ego and its alters is important for network classification. So essentially, one can classify the egocentric network as either heterogeneous, homogeneous, or both. With this classification in view, insight into the presence of strong and weak ties in the ego network becomes clearer. E.g., ties to family members tend to be stronger than those with colleagues and friends. Furthermore, the lack or presence of ties between alters indicates the degree to which alters in the ego network are connected to each other and the various behaviors or states as participating, browsing, aggression, etc. that are observed in a network. Ghawi et al. [2019]; Horng and Wu [2020]; Krivitsky et al. [2019]; McCarty et al. [2007]; Passarella et al. [2012].
These tie attributes can either be binary or valued. Relationship strength depends on intimacy and time invested by both parties. Of importance is its use in identifying members of the support clique and the sympathy group in the dunbar circle, Fig. 2.5. The alters situated in these regions of the dunbar circle have strong ties with each other and with the ego. This implies that they share the same information sources and environment. The relationship between the ego and those in outer circles tends to be weak, resulting in the ego being perceived as a structural hole in relation to these changes. Structural hole theory measures the social relationship between users in social networks and, more importantly, the benefits people derive from their connections. Therefore, the ego, acting as a structural hole, regulates the flow of information to external entities that are solely associated with it. Researchers have linked structural holes to innovation, good ideas, individual performance, among other things. Burt [2004]; Goyal and Vega-Redondo [2007]; Lin et al. [2021b]; Zaheer and Soda [2009]. It operationalizes two types of social capital:

- Information: If everyone in a given network is familiar with each other, then most of the information within the network will be redundant. So within the network, there is no access to novel information Perry et al. [2018].
- Power: A node that bridges two networks is able to control the flow of information and resources between them. By acting as a bridge between two unrelated alters, an ego can manage information and resource flow without being limited by them Perry et al. [2018].

2.3.1.4 Density of an Ego Network

Density gives a measure of the overall connections between the egos and alters Perry et al. [2018]. It is the total number of ties in the network, excluding the ties involving the ego, divided by the number of pairs of alters in the ego network Perry et al. [2018]. The density measure is used to measure the strength of the social safety net, i.e., whether the network is tightly bound, loosely bound, or has structural holes in it. Given a directed or undirected tie T and a given number of nodes N, we calculate the density for undirected ties as follows:

$$\mathbf{D}_{\mathrm{ut}} = \frac{\mathbf{2T}}{\mathbf{N(N-1)}} \tag{2.6}$$

and density for directed ties is calculated as:

$$\mathbf{D}_{\mathrm{dt}} = \frac{\mathbf{T}}{\mathbf{N(N-1)}} \tag{2.7}$$

The effective size of the network where an ego is located limits the density of an ego network because there is no limit to the number of ties a node could have.

2.3.1.5 Effective Size of an Ego Network

If in a network structure, alters can be reached through different pathways, then the resources or information flowing through the network will be redundant or old Perry et al. [2018]. To evaluate redundancy, the effective size measure is employed Perry et al. [2018].As shown in Fig. 2.6, it is defined as the ego's number of alters minus the average number of ties that each alter has to other alters. It is a positive function of network size and a negative function of the number of ties among alters Perry et al. [2018].

The formula for effective size is given as follows:

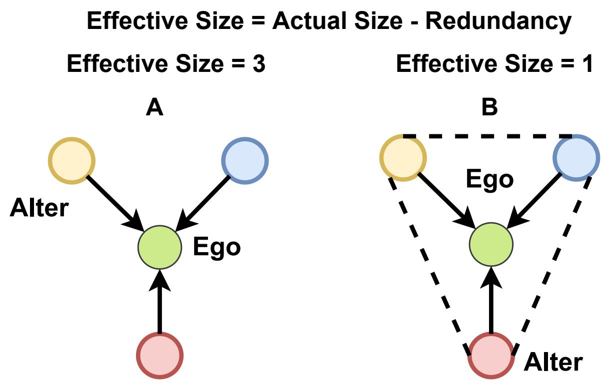


Figure 2.6: Effective Size of an Ego Network

$$\boldsymbol{ES}_e = \mathbf{n} - \frac{\mathbf{2t}}{\mathbf{n}} \tag{2.8}$$

where n is the number of alters, and t is the total number of ties to the ego network while excluding the ties to the ego Tabassum et al. [2018].

2.3.1.6 Ego Betweenness Centrality

In graph theory, betweenness centrality is a measure of centrality in a graph based on shortest paths. This reveals the structural importance of the node Buccafurri et al. [2013]; Everett and Borgatti [2005]. Therefore, in a connected graph, each pair of vertices has at least one shortest path that minimizes either the number of edges it passes through or the sum of the edge weights. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex Everett and Borgatti [2005]. A high betweenness value indicates that an ego node controls information flow between other nodes. For instance, we can consider a node with a high betweenness centrality as a suitable forwarder to enhance the efficiency of information delivery. Betweeness centrality is defined as

$$\mathbf{BC}(\upsilon) = \sum_{j \neq i} \frac{\sigma_{\mathrm{j,k}}(\upsilon)}{\sigma_{\mathrm{j,k}}} \tag{2.9}$$

where $\sigma_{\mathrm{j,k}}$ is the number of shortest paths between node j and node k and $\sigma_{\mathrm{j,k}}(\upsilon)$, the number of these paths that go through node υ. Note that the betweenness centrality of a node scales with the number of pairs of nodes as implied by the summation indices Perry et al. [2018].

2.3.2 Duocentric Networks

In this section, we will provide a brief description of duocentric networks, which are networks that facilitate diadic interactions. Given a directed network, we define a duocentric network as a sub-system consisting of a pair of nodes and their associated ties Coromina et al. [2008]; Griffin and Gonzalez [2003]. This pair of nodes are called dyads, the fundamental unit of interpersonal relations Griffin and Gonzalez [2003]; Knapp and Daly [2002]; Moreland [2010]; Yu-Hui and Fei [2010]. When a pair of egos is central to a research problem, we use the duocentric network. The main characteristic of this network is that it is bound around a pair of egos while ignoring the ties among the respective alters Kennedy et al. [2015]. When analyzing

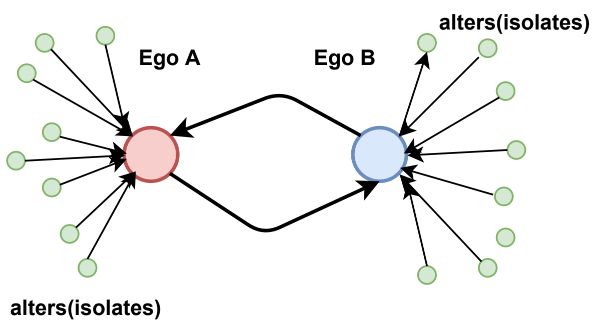


Figure 2.7: Duocentric network

a duocentric network, it's crucial to confirm if the two selected egos belong to the same class or category, meaning that a variable can accurately distinguish them from each other. Gonzalez and Griffin [1999, 2012]; Olsen and Kenny [2006]. Therefore, if

the goal of the research is to understand how male and female students perform in a programming language course, we can theoretically treat gender as a distinguishing variable. The students can further be distinguished by roles, i.e., if the respective course project is to be done by groups of students, and roles are designated within the groups. Apart from dyads being distinguishable, it is important to know if they are interchangeable. For exchangeable dyads, there are no relevant variables or roles that determine their interchangeability. One can consistently distinguish the egos in the dyads (e.g., same-sex friendships).
The following are the characteristics of a duocentric network, as shown in Fig. 2.7:

- Primary actors, Ego A and Ego B, must be central and expressed as egos.
- The ego model classifies other actors as alters.
- No relationships are captured among alters.
- We classify actors who only interact with one ego as isolates. arrow direction indicates the interaction and dependence of the item.

The interactions that occur in duocentric networks are not random. They are characterized as within-dyad dependencies. The commonalities and similarities shared by the network nodes in question both bound and influence these dependencies. The next subsection describes non-independence, a property that sheds light on the shared dependencies present in duocentric networks.

2.3.2.1 Non-independence

When dependencies exist between pairs of attributes belonging to nodes in dyadic interactions (e.g, a male and a female student that belong to the same project group in a university course), the attributes of these two individuals or nodes are then more similar to one another than other nodes, (i.e. other students in the same course in other groups). In the context of a dyad, we refer to other students and instructors as isolates within the main network. An isolate that is a structural hole influences the interaction dynamics within a duocentric network Burt [2004]. These two nodes are said to exhibit the non-independence property Kenny [1996]. This non-independence feature captures the commonalities shard by two sides of a dyad Kenny and Kashy [2014]; Kenny et al. [2020].

2.3.2.2 Dyadic Data Analysis

The primary objective of duocentric network analysis is the formulation of mathematical models that explain the non-independence property. We use two types of variables, namely exogenous and endogenous variables, for this purpose. Exogenous variables, depicted as "X" in the models, are independent variables. Only as explanatory variables do they appear, and the model does not determine their values. Endogenous variables, on the other hand, are dependent variables. Models will depict them as "Y". One or more variables in a model cause these phenomena. Also, an endogenous variable may cause another endogenous variable in a model Fox [2006]; Iacobucci [2009]; Kenny [2011].
Furthermore, a structural equation model is defined for each endogenous variable.

These structural equation models are multiple equation regression models representing assumed causal relationships among a number of variables, some of which may affect each other mutually. Fox [2006]. Using these variables, different models that produce non-independence in a duocentric network setting—the social relations model, the actor-partner model, the mutual influence model, and the common fate model—will be described in the next sub-subsections.

2.3.2.3 Actor-Partner Interdependence Model

In the actor-partner model, non-independence is hypothesized to occur as a result of preexisting attributes of each partner, which affect both his or her own interaction behavior and also the interaction behavior of his or her partner Campbell and Kashy [2002]; Kenny [1996]; Woody and Sadler [2005]. For each partner, as shown in Fig. 2.8, there exist endogenous variables and exogenous variables. The "X_i" as previously described, represent preexisting attributes or predispositions that the two actors bring to the interaction that may shape their interaction behaviors depicted by the "Y_j". Thus, in a class project consisting of dyadic groups, the neglectful behavior of a group member may result in poor performance for every group.

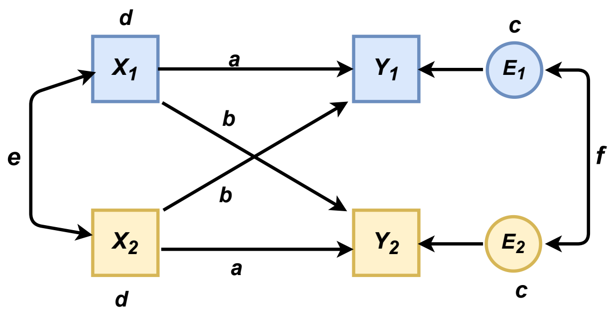


Figure 2.8: Actor Partner model Woody and Sadler [2005]

$$\mathbf{Y}_1 = \mathbf{aX}_1 + \mathbf{bX}_2 + \mathbf{E}_1 \tag{2.10}$$

$$\mathbf{Y}_2 = \mathbf{bX}_1 + \mathbf{aX}_2 + \mathbf{E}_2 \tag{2.11}$$

In Fig. 2.8, 'a' and 'b' represent the paths and are equal across the two members of the interacting pair; 'c' represents the error variances; 'd' represents the variances of the exogenous variables. The symbol 'e' signifies the covariance between the exogenous variables X1 and X2. The symbol 'f' signifies the covariance between the errors or disturbances E1 and E2.
The heart of the model is Paths a and b. Path a represents the actor effect, i.e., the effect of a person's level of X on his or her own level of Y. Path B represents the partner effect, i.e., the impact of a person's level of X on his or her interaction partner's level of Y. Woody and Sadler [2005] The structural equation model and the multilevel modeling or hierarchical linear model are two modeling approaches applicable to analyzing the actor-partner interdependence model.

2.3.2.4 Common Fate Model

In the common fate model, influences at the dyad level impact both partners in the same way, making their behaviors non-independent. Latent variables in a statistical model are random variables that are not necessarily immeasurable. We employ them to represent features of interest in a model that are either not directly measurable or not measured. We can also use them to construct estimators from non-latent variable models that are more efficient. We conceptualize these shared situational or environmental pressures as dyad-level latent variables. We can also use them to construct estimators that are more efficient than those derived from non-latent variable models. Spirtes [2001]. In Fig. 2.9 two indicators, X_1, X_2, and Y_1, Y_2 are used to measure the latent variables. They reflect the scores of dyad member A and member B (e.g., husband and wife) on the underlying latent construct Gonzalez and Griffin [2002]; Kenny [1996]; Ledermann and Kenny [2012]; Woody and Sadler [2005]. As shown in Fig. 2.9, one dyad-level latent variable, LX, influences another dyad-level latent variable, LY. The path ”a” indicates the direct influence of LX on LY.

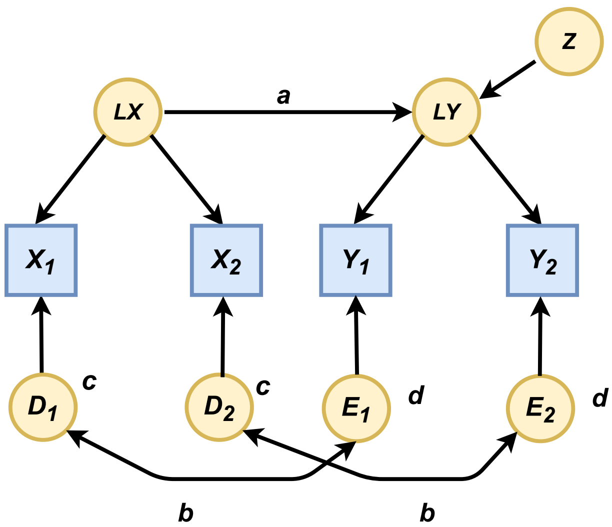


Figure 2.9: Common fate model Ledermann and Kenny [2012]; Woody and Sadler [2005]

Using LX as an example, the variances and standard error of the between-dyad latent variables LX, LY, and Z are calculated as follows:

$$\mathbf{Var(LX)} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{Var(LX_{\mathrm{bd}})} \tag{2.12}$$

$$\mathbf{SE} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{SE}_{\mathrm{Var(LX_{bd})}} \tag{2.13}$$

2.3.3 Triadic Networks

Given a directed or undirected network, a triadic network is a subnetwork consisting of any three nodes and their associated ties. These nodes take either a null or unconnected configuration, a disconnected or connected pair configuration, or an open or closed configuration, as shown in Fig. 2.10. Nodes in a triad are transitively associated to each other Faust [2010], and to determine their roles, we take the triad census, i.e., we count the number of the different triad variations it participated in.

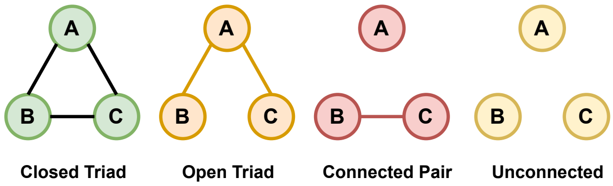


Figure 2.10: Triads Huang et al. [2015]; Tsvetovat and Kouznetsov [2011]

These nodes can form either directed or undirected relationships. For undirected relationships, the nodes can form a closed, open, and unconnected relationship. The unconnected relationship can either be completely unconnected or a connected pair, as shown in Fig. 2.10. The closed triad describes a cyclic relationship such that all nodes in the triad are connected, i.e., A|T|B (A is tied to B), B|T|C, and A|T|C. In the open triad, a single node A mediates the relationship between node B and node C. So we have A|T|C and A|T|B. Hence, information passes from A to B, then C, and back to A. Directed relationships constitute an isomorphism. An isomor-

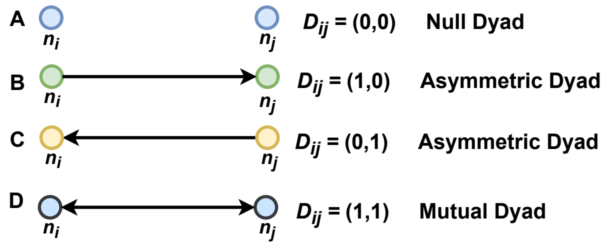


Figure 2.11: Dyadic isomorphism Uddin et al. [2013]

phism is a structure-preserving mapping between two structures of the same type that can be reversed by an inverse mapping Holland and Leinhardt [1974]; Mazur et al. [2007]; Mazur [2007], and owing that two subgraphs are isomorphic if they are identical Wasserman and Faust [1994], a dyad that is neither asymmetrical nor mutual is null as shown in the sociomatrix in Fig. 2.11. Thus we have the first Dyad variation, the null Dyad. The second isomorphism is invariant to a transformation, such as reflection; hence, it is not possible to distinguish between the two different forms, i.e., B(i $\rightarrow$ j), and C(j $\rightarrow$ i) of asymmetric dyadic relations. The mutual dyad relationship, denoted by $i \Longleftrightarrow j$ between actor i and actor j, comes into play when i $\rightarrow$ j and j $\rightarrow$ i in the dyad Moody [1998]; Uddin et al. [2013]. Thus, the mutual dyadic relation between actor i and actor j is represented by $D_{ij} = (1, 1)$ as shown in Fig. 2.11.

Thus, for a directed triad relationship, there exists $\binom{g}{3}$, distinct 3-subgraphs formed by selecting each of the possible subsets of the 3 respective nodes and their corresponding ties. This results in 16 isomorphism classes, as shown in Fig. 2.12.

The letter U stands for up, D for down, C for cyclical, and T for transitive (i.e., having two paths that lead to the same endpoint). The variation denoted with 120D has 1 mutual, 2 asymmetric, 0 null dyads, and the down orientation. In this

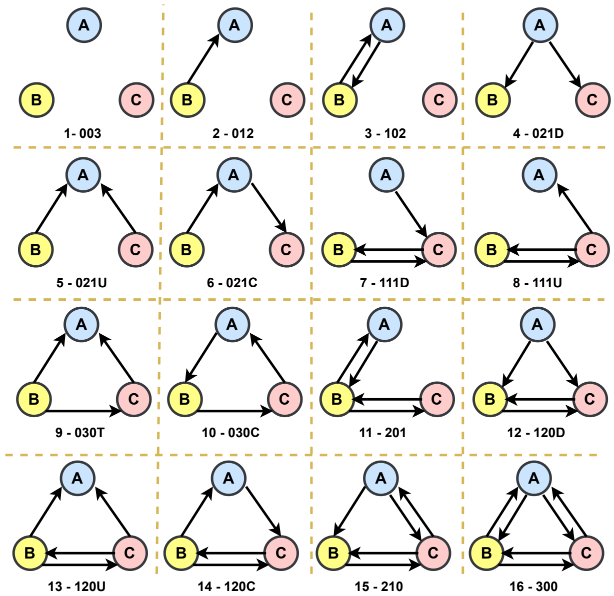


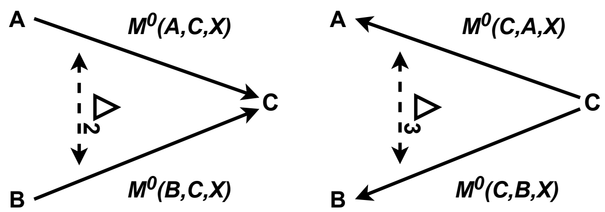
Figure 2.12: Directed triad relationship configurations Huang et al. [2015]; Tsvetovat and Kouznetsov [2011]

manner, the triads 1-3 depict an unconnected relationship, triads 4-8 and 11 depict variations of structural holes, and triads 9, 10, and 12-16 are variations of closed triads.

The relationship between nodes in directed triad relationships eventually becomes a closure. Triadic closure, which is also called transitivity or clustering, is when ties form in open triads and close over time Kossinets and Watts [2006]; Song et al. [2019]. So for two individuals with a common acquaintance, there is a high likelihood of a tie forming between them via the social influence of their common acquaintance Easley et al. [2012]; Mantzaris and Higham [2013]; Song et al. [2019]; Zhang et al. [2018]. Triadic closure not only occurs in stand-alone triads but also in triads within large groups and entire networks. Thus, as one mutual connection increases the likelihood of tie formation between two individuals, multiple mutual connections increase the probability for even more connections Louch [2000]; Song et al. [2019]. To measure the presence of triadic closure, we employ the clustering coefficient measure, which is a measure of the degree to which nodes in a graph tend to cluster together Opsahl [2013]; Yin et al. [2020]. In the next subsection, we describe mutual modeling and the triadic relations model.

2.3.3.1 Mutual Modelling

Mutual modeling is a bidirectional approach employed in both dyadic and triadic interaction modelling Dillenbourg et al. [2016]. Given a task involving three actors, A, B, and C, A builds a model of B and C, B builds a model of A and C, and C builds a model of B and A. This is represented using the notation $\mathcal{M}$(C, A, $\mathcal{X}$) which denotes "C knows that A knows $\mathcal{X}$". As the non-independence assumption is in play, C's model of what A knows includes what C knows about A. So, if A states, "C thinks I am proficient in programming," A then builds a second-level model: $\mathcal{M}$(A, C, $\mathcal{M}$(C, A, Programming - Skill)). Furthermore, $\mathcal{M}^o$(C, A, $\mathcal{X}$) represents the degree of accuracy of the model. So, for the accuracy of what A, B, and C models about each other, we have 6 models as shown in Fig. 2.13



$\mathcal{M}^o$(A, B, $\mathcal{X}$), $\mathcal{M}^o$(B, A, $\mathcal{X}$), $\mathcal{M}^o$(A, C, $\mathcal{X}$),
$\mathcal{M}^o$(C, A, $\mathcal{X}$), $\mathcal{M}^o$(C, B, $\mathcal{X}$), $\mathcal{M}^o$(B, C, $\mathcal{X}$),

Figure 2.13: Mutual modeling in a triadic interaction Dillenbourg et al. [2016]

2.3.3.2 Mutual Modelling (TRM)

The triadic relations model extends the logic of the social relations model Back and Kenny [2010]. It takes into account the characteristics of the perceiver, actor, and partner, as well as their combinations, all resulting in seven variances and 16 covariance estimates. Given a situation in which what to deduce if an actor A aggrees against partner B according to partner C, the tiadic relations model assumes that the percievers insight is comprised of eight components as shown below:

$$X_{\mathrm{ijk}} = M + a_{\mathrm{i}} + b_{\mathrm{j}} + c_{\mathrm{k}} + ab_{\mathrm{ij}} + ac_{\mathrm{ik}} + ac_{\mathrm{ik}} + bc_{\mathrm{jk}} + abc_{\mathrm{ijk}} \tag{2.14}$$

where M is the mean perception within the group, $\mathrm{a_i}$ is the groups perception of actor i's aggressiveness, $\mathrm{b_j}$ is the groups perception of partner j's victimization, $\mathrm{c_k}$ is the perception perceiver k has of aggression (among peers in general), $\mathrm{ab_{ij}}$ is the groups perception of actor i's aggressiveness toward partner j, $\mathrm{ac_{ik}}$ is the perception subject k has of actor i's aggression toward others, $\mathrm{bc_{jk}}$ is the perception subject k has of partner j's victimization by others, and $\mathrm{abc_{ijk}}$ is the specific perception k has of actor i's aggression toward partner j Card et al. [2010]. Having derived these components, the individual level variances, dyad variances, and triad level variances are calculated, and further estimations derived from them.

While there are other models employed in triadic interaction modeling, as observed in our literature survey, most of them incorporate stochastic assumptions that violate the non-independence assumption.

2.3.4 Networks of Arbitrary Size

Social capital is an efficacy derived from collaborative connections between individuals that results in the accomplishment of goals Sandefur and Laumann [1998]. For example, a group with high trustworthiness and skill for a specified task is able to accomplish much more than a comparable group with the same level of skill and no trust. As such, the structural importance of an individual or node in a network or social unit is affected by its centrality with respect to the flow of social capital. Thus, the formation of new ties, choice of partners, and evolution of the above-discussed networks are mainly driven by homophily and directed by preferential attachment De Salve et al. [2018]; Maoz [2012]. Consequently, systems that produce power-law distributions follow a pattern in the growth or evolution of dyadic and triad networks into networks of arbitrary sizes. These systems are described as scale-free. The next subsection will describe the scale-free networks.

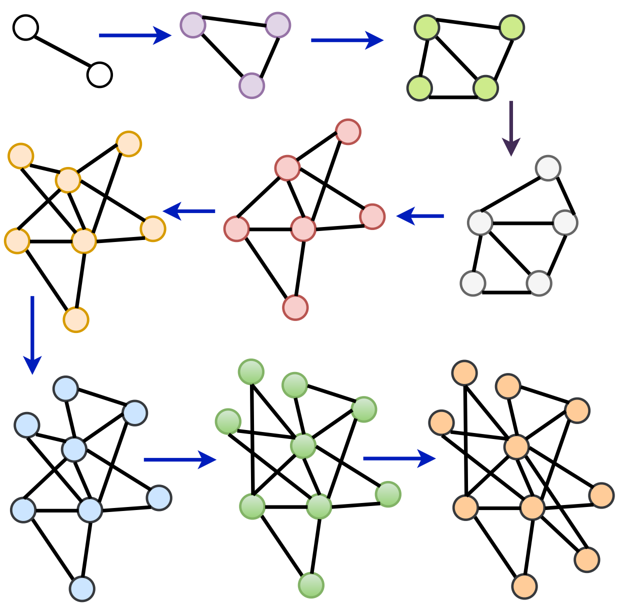


Figure 2.14: Evolution of a scale-free network as a consequence of transitive relationships

2.3.5 Scale-Free Networks

The term scale-free is a mathematical expression used to describe the power-law characteristics of a probability distribution. The most basic model capable of producing a power-law degree distribution is the Barabási-Albert (BA) model Barabási and Albert [1999]; Hauff and Nürnberger [2006]. Barabási-Albert's model connects newly created nodes to existing nodes at each time step, adhering to the "preferential attachment" principle. Thus, given a scale-free network, the probability $\boldsymbol{P}(\boldsymbol{K})$ of a node having $\boldsymbol{K}$ links follows a power law with degree exponent γ as shown in equation 2.15

$$\boldsymbol{P}(\boldsymbol{K}) \propto \boldsymbol{K}^{\gamma} \tag{2.15}$$

Furthermore, results from the study of Hein *et al* Hein et al. [2006] in which the internet was mapped show that the majority of the pages or nodes had few links, while a few pages had a large number of links. This is illustrated in Fig. 2.15.A. The logarithmic plot of the distribution of the edges is further shown in Fig. 2.15.B,

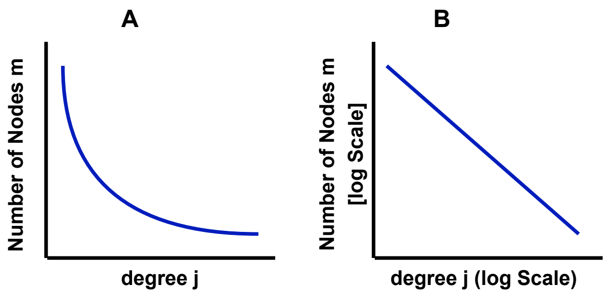


Figure 2.15: Power law distribution of node linkages Barabási and Bonabeau [2003]

which reveals the power-law characteristics of the distribution Hein et al. [2006]; Newman et al. [2006]. This power-law characteristic explains Why in a network, a large number of nodes have very few connections, and a small number of nodes, structural holes, have a very high degree Hauff and Nürnberger [2006]. In all, driven by preferential attachment, the previously discussed networks continuously grow as shown in Fig. 2.14, thus scale free.

2.4 Chapter Summary

This chapter began with a literature-driven review of the model of interest, which is the graph, followed by a description of the learning network interaction hierarchy, and finally, a discussion of the various types of networks found in learner-centered environments. The literature-driven review of the graph model provided insights into its various applications and benefits in educational settings. Additionally, the description of the learning network interaction hierarchy sheds light on the different levels of engagement and collaboration among learners within these networks. Furthermore, the exploration of different network types in learner-centered environments highlighted their unique characteristics and how they support personalized learning experiences. The ego model's well-defined structure, high degree of instrumentation for interaction modeling, and ease of eliciting information set it apart from other network models. The interactions that occur in duocentric networks are not random. They are characterized as within-dyad dependencies. The commonalities and similarities shared by the network nodes in question both bound and influence these dependencies. The next subsection describes non-independence, a property that sheds light on the shared dependencies present in duocentric networks. In directed triad relationships, nodes eventually form a closure. So for two individuals with a common acquaintance, there is a high likelihood of a tie forming between them via the social influence of their common acquaintance Easley et al. [2012]; Mantzaris and Higham [2013]; Song et al. [2019]; Zhang et al. [2018]. Triadic closure not only occurs in stand-alone triads but also in triads within large groups and entire networks. Thus, as one mutual connection increases the likelihood of tie formation between two individuals, multiple mutual connections increase the probability for even more connections Louch [2000]; Song et al. [2019]. We use the clustering coefficient measure to see if triadic closure is present. This measure shows how much nodes in a graph tend to group together Opsahl [2013]; Yin et al. [2020]. This power law explains why in a network, a large number of nodes have very few connections, and a small number of nodes, structural holes, have a very high degree Hauff and Nürnberger

[2006]. Overall, preferential attachment drives most of the growth in social networks, allowing them to scale without interference. Lastly, the hybrid pedagogy, in which both traditional and digital teaching strategies are employed to assist learners in their learning efforts, was briefly described Crawford and Jenkins [2017]. In the next chapter, we delve into the challenge, requirements of collaboration-centered learning, and a strategy for structuring teamwork in a university environment.

3. Collaboration Centered Learning

This chapter draws upon the research effort from the following papers: Obionwu et al. [2023b,e, 2022c,f, 2023f, 2024]

The challenge, structure, and requirements of the 21st-century work environment have made the acquisition of teamwork and collaborative problem-solving skills indispensable Sundstrom et al. [1990]. This is most evident in the information technology sector, where the work is often split into well-defined sub-tasks to create complex tools. Ergo, it requires a team of individuals with different backgrounds and skill sets. Usually, the basis for this skill acquisition is set during a person's studies. Due to the recent move to online learning in most institutions of higher education, curriculum administrators and developers are resorting to online environments that can stimulate task engagement, team collaboration, task reflection, and the acquisition of teamwork skills. Thus, in this chapter, we describe the collaborative learning environment, section 3.1, a strategy for recommending project partners, section **??**, and the design and development of a collaboration platform, section 3.2.

Early implementations of team-based learning showed that collaborative problem-solving within small groups was effective in stimulating active learning Gomez et al. [2010]; Michaelsen et al. [2004]. As observed in Michaelsen et al. [2004], team members assumed specific roles in an effort to efficiently solve the assigned tasks. While most team members were not effectively suited for the assigned roles, team leaders took it upon themselves to ensure their peers' learning. This challenge of fitting team members into defined roles still persists in recent traditional lecture settings Michaelsen and Sweet [2011]. In the next section, we will describe the collaborative learning environment.

3.1 Collaborative Learning Environment

Collaboration entails working with other people for an overall directed output. Generally, grouping enables team formations, within which team members synchronize to achieve a preset goal. The word collaboration is derived from the word cooperation [Jermann et al., 2001]. Renowned US educator John Dewey is accredited with promoting regular and systematic cooperation in learning environments' [Martin, 2003]. Dewey argues that learning processes are social and interactive; thus, students grow in environments where they are allowed to work in groups and interact with the curriculum. He further stressed that students should be able to participate in their learning process in this manner. The basic elements of collaboration that affect the nature of collaboration are positive interdependence, interaction, individual contribution, and interpersonal skills. Collaboration enhances a socially structured exchange of information and promotes learning among the learners. Generally, collaborative learning involves grouping students for effectiveness in the sharing of knowledge [Sita Nirmala Kumaraswamy and Chitale, 2012]. In such a process, every student is given a specific learning activity, and then they are allowed to share their content with the group members for further review and brainstorming. Consequently, students can criticize and compliment other group members' ideas, enhancing their learning process. Such a process improves communication among the group members and allows effective learning among the members.

In the case of professional environments, collaboration leads to effective task completion. For instance, companies have dedicated teams for every complex activity, like production, marketing, and finance processes [Burbank and Kauchak, 2003]. Team members are assigned a particular task; in the group, they synchronize and complete the tasks in unison. The team's efficiency is improved by assigning a team leader who manages the team's activities. The team members are expected to collaborate and usually complete their tasks efficiently, resulting in better product creation [Rius-Sorolla et al., 2021].

Collaborative learning is dynamic as it involves working in a group where everyone will have a different viewpoint and background. Synchronizing and directing the group members toward specific task completion requires a dynamic team leader. Collaboration in a school environment ultimately leads to a better learning environment. Exposing learners to such a dynamic process early in life will enhance their social and interpersonal abilities. Developing such dynamic, collaborative methodologies can revolutionize the learning process and help learning environments be more realistic and valuable to the learner [Warsah et al., 2021].

To sum up, the present-day world is enhancing collaborative activities to enhance the learning process and make it more accessible to learners through technological tools [So et al., 2010]. Digital technology tools are reshaping collaborative activities and rendering better experiences for learners. Online platforms like Zoom and Google Meet are making groundbreaking changes and reinventing how people can meet and collaborate in professional and personal interactions. In many workspaces, collaborations have gone digital, and people work more efficiently through these platforms. These online platforms have made discussions among students and teachers more interactive through mediation tools such as chat modules and other discussion facilities. These tools enable participants to engage in real-time conversations, ask questions, and share resources, fostering a more dynamic and inclusive learning environment. Additionally, the integration of multimedia features like screen sharing and virtual whiteboards further enhances the collaborative experience by facilitating visual presentations and brainstorming sessions[hong Huang and Ning, 2021].

Collaborative learning environments can be distinguished into formal and informal collaboration based on the nature of collaboration [Shane, 2005]

.

3.1.1 Formal and Informal Collaboration

1. **Formal Setting:**
   Formal learning settings include traditional classrooms, professional environments, and organizational training [Wiener, 1986]. These settings frequently have instructors or trainers who are experts in the subject matter and follow a structured curriculum. They provide a formal framework for learning, with clear objectives and assessments to measure progress and achievement. Traditional one-way knowledge transfer, for example, involves the instructor delivering lectures and students taking notes. This approach may limit interaction and engagement, as students primarily receive information passively. However, some formal learning settings have evolved to incorporate more interactive methods such as group discussions, hands-on activities, and collaborative projects. These approaches aim to enhance understanding and retention by promoting active participation and knowledge application. [Friend, 2000]. Organizational training is also a form of formal learning environment. In this training environment, collaboration is used to provide resources for the trainers to interact and develop a comprehensive understanding of the course content. Collaboration in formal settings is improved by several technological tools, which make collaboration more accessible and affordable [Bacon, 2008].

2. **Informal Setting:**
   Peer group interactions, private groups, community groups, social welfare groups, recreation groups, and sports groups are some known informal settings that induce collaboration. For instance, children learn more in a playful environment than in a traditional classroom [Allen et al., 2007]. Children's collaborations drive them to creative environments, which are mostly informal [Davies et al., 2013]. Informal collaborations promote sharing the thought process, which is important for diversifying learners' psychology. Furthermore, personal or family groups tolerate collaboration among siblings and share personal emotions and experiences to learn and excel in social engagements. Community groups work on specific tasks or ideologies that enhance community activities and turn ideologies into reality, and these collaborations are known to make such groups efficient [Pejovich, 2006]. In these settings, the role of technology in enhancing both the interaction process and informal learning, where it is necessary, is well documented. Informal learning creates better opportunities for professionals to learn and upgrade different skills, enhancing their financial and economic conditions [Manuti et al., 2015]. In the next section, we describe collaborative learning for higher education.

3.1.2 Collaborative Learning for Higher Education

The higher education system seeks to impart a skill set that enhances learners' critical thinking capacity, analyzing capabilities, and communication techniques. Collaboration methodologies can enhance higher education. Such learning methodologies can influence the learner environment and improve efficiency in learning. Higher education involves directional communication between teachers and students. The primary objective of the higher education system is to cultivate a skill set that improves learners' ability to think critically, analyze effectively, and communicate proficiently. Collaborative strategies have the potential to impact the learning environment and enhance learning efficiency, and thus, collaboration in higher education can simplify learning objectives through effective communication and efficient knowledge transfer [de Hei et al., 2020]. Most higher education also involves project-based tasks, in which projects are assigned to the team. In such scenarios, learners generally resort to peer assessments to assess who is more proficient in specific skills or knowledge areas. Peer assessments not only provide learners with valuable feedback on their own performance but also foster a sense of accountability and motivation within the team [Forbes, 2020]. In some scenarios, the tasks are complex and require more time and cognitive effort. To solve such tasks, collaboration methodologies are employed. Team members basically split the task into smaller bits and assign them to special dedicated teams to develop solutions and resolve the complex modules of the task [Kirschner et al., 2009]. The types of academic collaboration are described in the next subsection.

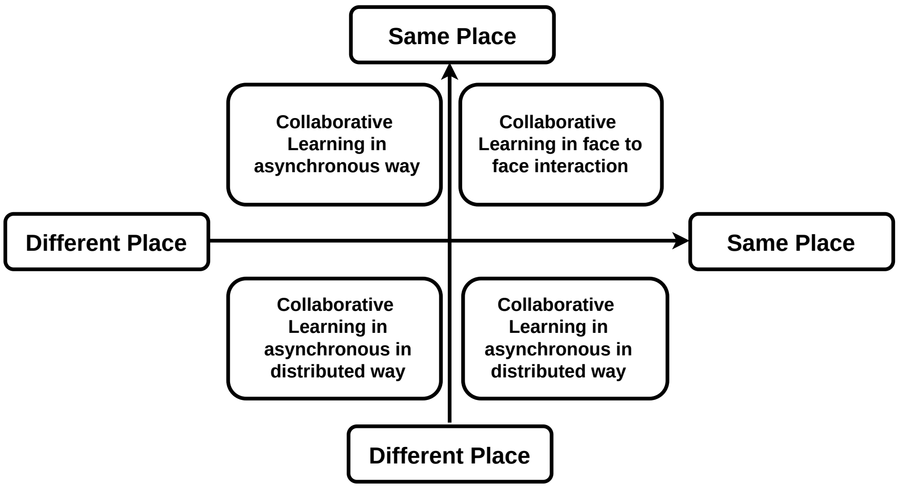


Figure 3.1: Four modes of collaborations [Baker, 2021]

3.1.2.1 Types of Academic Collaboration

Based on space and time complexity, there are four modes of collaboration in collaborative learning frameworks [Baker, 2021]. They include collaboration learning with face-to-face interactions, collaboration learning in an asynchronous manner, synchronous communication in a distributed way, and asynchronous communication in a distributed way, as shown in figure 3.1.

1. **Collaborative learning with face-to-face interaction:**

   In this type of collaboration, students are physically engaged to form groups and collaborate to achieve better results [Ellis, 2001]. This collaboration will have peer-to-peer interactions to discuss the targets of collaboration, who are expected to work in synchronization to complete the task.

2. **Collaborative learning with an asynchronous manner:**

   To pass the information to the group of learners, the learners utilize available tools to pass information to their peers [Suthers et al., 2008].

3. **Collaborative learning in asynchronous communication in a distributed way:**

   In this collaborative environment, the communication process is not continuous and real-time but packed in tools and papers for circulation purposes [Schellens and Valcke, 2005]. Such collaborations are made in informal learning environments like online communities or open and closed groups. In online collaboration, different tools, such as email, blogs, and Wikis, are used to communicate information and knowledge to the group members. People generally use notice boards, memo boards, and circulars to disseminate information to group members in physical settings.

4. **Collaborative learning in synchronous communication in a distributed way:**

   In synchronous distributed communications, the content or the information is shared among the collaborators who do not share similar physical spaces in real-time [Marjanovic, 1999]. In such conditions, online meetings, video conferences, and group calls are used. This type of communication is used in formal settings.

The knowledge of how a team member's dispositions, what they value most, their strengths, their weaknesses, and their communication style affect the overall dynamics and productivity of the team helps in assigning tasks and responsibilities in a way that aligns with each individual's skills and preferences, leading to a more harmonious and successful team environment. It is also a frequent occurrence that teams break down as a result of individual differences in personality and decision-making, which ultimately results in the failure of the assigned responsibilities and tasks. Thus, it becomes necessary to have a team member recommendation system for course projects that takes into account these hidden personality traits and dispositions. Towards the goal of conceptualizing such a system, the big five personality traits will be described in the next section.

3.1.3 Big Five Personality Traits

Personality refers to consistent patterns of thoughts, emotions, and behaviors that define individuals Snyder and Ickes [1985]. Studies have shown that specific personality traits can impact how students approach learning tasks, engage with instructional materials, and respond to different teaching approaches. Felder and Brent [2005]. The Big Five Personality Traits, also widely known as the Five-Factor Model, is a well-researched and commonly accepted model that identifies five core dimensions of human personality. Though there are multiple numbers of traits that can be used to measure personality, nonetheless, most traits can be categorized either as a facet

Figure 3.2: Big five personality traits with questionnaire Liapis et al. [2022]

of one of the Big Five or as a compound trait reflecting a blend of two or more of the various traits at all levels Colin [2015]. It has also been identified as being more closely related to academic performance Komarraju and Karau [2005]. According to the Big Five personality predictors of postsecondary academic performance, these five characteristics have a positive or negative correlation with academic performanceBooker et al. [2007]. The results show that conscientious students perform better academically than those who are not conscientious due to their motivation. Neuroticism's negative association with academic performance stems from emotional instability; however, students with a healthy mindset do not experience this negative association. Openness and extraversion produce mixed results, with extraversion influencing student performance when classroom engagement is required. Chamorro-Premuzic and Furnham [2003]. According to the findings of the study, agreeableness is not a significant predictor of a student's academic performance, implying little correlation with academics Komarraju et al. [2011]. A description of the respective traits is given below:

- **Extraversion**: One of the Big Five personality qualities is extraversion. Extraversion is associated with individuals who are outspoken, friendly, and assertive. Depending on the situation, extraversion may not have any effect on students' behavior. Extraversion refers to the degree to which a person is outgoing, assertive, and emotionally expressive. One who is high in extraversion is highly social and talkative and enjoys being around others, while one who is low in extraversion is more reserved and introspective and tends to recharge by spending time alone Carter [2009]. A student who is very good at communication is termed high in extraversion, and a student who is lacking in communication is classified as low in communicationChen and Caropreso [2004]. The Communicating and Role

Questionnaire is a form of questionnaire that measures extraversion. Students who perform better on this section are more extroverted than those who score low. Gallagher [1996].

- **Neuroticism**: Being emotionally stable is a trait known as neuroticism. It shows that emotionally healthy students outperform emotionally unstable students in the classroom Komarraju et al. [2011]. Neuroticism measures how emotionally stable or resilient an individual is with regard to negative emotions like anger, anxiety, and depression. A person high in neuroticism is low in emotional stability and can easily experience high levels of anxiety, mood swings, and emotional reactivity, while a person low in neuroticism is usually highly emotionally stable and can remain calm, composed, and emotionally secure in challenging situations Carter [2009]. Consistent pessimistic tendencies identify a highly neurotic student, while consistently high optimism characterizes a lowly neurotic student. The optimist and pessimist questionnaires are associated with this trait. Students with better scores have a tendency to be more optimistic and have a positive outlook for the future than students with lower scores Carter [2009].

- **Agreeableness**: Being agreeable is one of the attributes that shows if a person is kind and courteous. It correlates with whether students behave in a tough or sensitive manner. Despite the fact that there is some connection to academia Komarraju et al. [2011], agreeableness assesses an individual's level of warmth, compassion, affection, and consideration towards others. Individuals high in agreeableness tend to be empathetic, cooperative, kind, and avoid conflict when possible, while people low in agreeableness are more competitive and less empathetic. On the scale of agreeableness, a student who is tender in social relations is high on agreeableness, while one who is tough and usually difficult to influence socially is considered low on agreeableness. Additionally, agreeableness influences the formation and performance of a team. Students who scored lower in this part are tender, whereas those who scored better overall are tougher. Carter [2009].

- **Conscientiousness**: Conscientiousness reveals a person's level of organization and discipline. Additionally, it has been discovered that students with higher levels of conscientiousness surpass those with lower levels in terms of academic performance Komarraju et al. [2011]. Conscientiousness pertains to the extent to which a person is thoughtful, self-controlled, and goal-directed in their behavior. An individual high in conscientiousness is usually more organized, diligent, and reliable, while one who is low in conscientiousness is more impulsive, less disciplined, and poor in time management. A student who is high in conscientiousness can be known by his excellent management skills around people and resources, and the one who is low in conscientiousness usually has low management skills around people and resources Poropat [2009]. Additionally, it demonstrates the strongest correlation between all five personality characteristics Komarraju et al. [2011]. The conscientiousness questionnaire covers managing people and resources. Students who score higher on this part of the questionnaire are classified as conscientious Poropat [2009].

- **Openness**: Openness is a sign of a person's receptivity. It reveals whether someone is willing to face challenges and take risks. This element has only had a minor effect on how well students work together as a team, Komarraju et al. [2011]. Openness refers to the extent to which a person is desirous of new ideas, new knowledge, and new experiences. An individual who is high in openness tends to exhibit creativity, while one who is low in openness tends to be more inclined toward routine and resist change. A student who loves to take great risks in gaining new knowledge and experiences for higher success is classified as being highly open, and one who prefers low-risk taking and avoids change is classified as being low in openness Carter [2009]. This wider Big Five model trait encompasses both success and risk assessments. Higher scorers on this questionnaire are more inclined to take chances, and vice versa is true for lower scorers. Carter [2009].

An analysis of the relationship between collaboration and personality traits based on different studies clearly shows the effect that personality traits have on influencing collaboration among students Balakrishnan and Gan [2016]; Carro and Sanchez-Horreo [2017]. For example, studies have found that individuals with extroverted personalities tend to be more inclined toward collaborative work, as they thrive in social settings and enjoy interacting with others. On the other hand, individuals with introverted personalities may prefer independent work and require more encouragement or structured collaboration opportunities to fully engage in group projects. This knowledge paves a path to community detection. Communities are groups of individuals who share common interests, goals, or characteristics. By understanding how personality traits influence collaboration, researchers can identify and analyze the formation of communities within a group of studentsChen and Caropreso [2004]; Williams [2005]. We can utilize this information to foster more productive collaborative environments and enhance the overall dynamics of the group. In the next section, we introduce the concept of community detection.

3.1.4 Community Detection

Communities are groups of strongly connected nodes with similar properties local to their group Kanawati [2015]. We use suitable community detection algorithms to identify such communities in a network. A large collection of literature on various algorithms used over the years for different purposes is present. Some of the extensively used algorithms are Edge Betweenness Newman and Girvan [2004], Fast Greedy Clauset et al. [2004], Walktrap Pons and Latapy [2005], Label Propogation Raghavan et al. [2007], Infomap Rosvall and Bergstrom [2008], Multilevel or Louvain Blondel et al. [2008], Leiden Traag et al. [2019]. All these algorithms identify communities based on a metric named ”modularity” that was first mentioned by Newman and Girvan [2004]. Although, in recent years, new metrics such as ”surprise,” ”significance,” ”conductance,” etc. have been proposed, limited applications are seen based on them, as most of the works being carried out today in community detection are still based around modularity only Chakraborty et al. [2017]. We have attempted to apply both modularity and surprise metrics to our data for community detection purposes. These metrics fall under the category of intrinsic measures. Although there is another category of measures known as extrinsic measures, all these metrics aim to confirm the essence of the communities, albeit in distinct contexts, as outlined below.

Intrinsic Measures

These metrics focus the evaluation process on the internal community structure, specifically assessing the types of connections within it. Finding the best partition, or group of communities for a network from a huge number of possible choices by improving an intrinsic metric is usually thought of as an NP-hard problem in the field of network analysis Chakraborty et al. [2017]. Thus, a partition detection algorithm optimizes a specific metric to identify communities within a network, generating an optimal solution that represents a derived community. It must be noted that other potential solutions exist but remain unknown Chakraborty et al. [2017]. The next section provides a concise overview of the metrics employed in our study.

Modularity: There are several ways to interpret the concept of modularity. In the context of a weighted network, as expressed by Gates et al. Gates et al. [2016], modularity refers to the quantification of the relative strength of connections within a community compared to connections outside of that community. Modularity for the derived partition corresponding to a positive weighted network, $Q_{modularity}(P)$ is given by Equation 3.1 (adapted from Gates et al. [2016]).

$$Q_{modularity}(P) = \frac{1}{2w_e} \sum_{ab} \left(w_{ab} - \frac{w_a w_b}{2w_e} \right) \delta(g_a, g_b) \tag{3.1}$$

In the above equation, w_e is the strengths totaled for edges belonging to that network, w_{ab} denotes strength for an existing edge between nodes a and b, while the expected strength between them is given by the term $w_a w_b / 2w_e$ where w_a and w_b represent strengths of nodes a and b, respectively. Lastly, $\delta(g_a, g_b)$ is the "Kronecker-Delta" function, a condition whose value is 1 if nodes a and b are present in the same community; g else, its value is taken as 0. The range of $Q_{modularity}(P)$ is between -1 and 1, with 1 being the best value.

Surprise: The surprise metric for improving its efficiency and adaptability to weighted networks was extended from Aldecoa and Marin [2011] as an asymptotic surprise Traag et al. [2015]. It is a measure that calculates the partition's quality and corresponds to the level of uncertainty of finding that partition by chance through simply improving the edges present inside the communities of a random network Aldecoa and Marin [2011]. Asymptotic surprise for the derived partition corresponding to a positive weighted network, $Q_{surprise}(P)$ is given by Equation 3.2 (adapted from Traag et al. [2015]).

$$Q_{surprise}(P) = eD(r \| \langle r \rangle) \tag{3.2}$$

In the above equation, e corresponds to total edges in a network, while $r = e_{int}/e$ is the count assigned from the internally existing edges concerning communities, while $\langle r \rangle = E_{int}/E$ accounts for expected internal edges, and $D(r \| \langle r \rangle)$ is its "Kullback-Leibler divergence.".

Extrinsic Measures

These metrics evaluate the correctness of the detected communities by performing a comparison in terms of similarity with the actual communities. Therefore, the evaluation does not rely on the internal network structure, which is why it is called an extrinsic measure. To conduct a comparison, it is necessary to have access to the actual communities that align with the ground truth. Alternatively, these metrics can be used to check the similarity of the derived communities from several algorithms. For our purpose, the latter scenario is applicable. There are several metrics available for comparing two sets of communities, and we have considered adjusted mutual information in our evaluation. This metric provides the correct similarity score between the two sets, even in the absence of ground truth, regardless of the internal arrangement of community labels. Pedregosa et al. [2011].

Adjusted Mutual Information (AMI): Adjusted mutual information is a metric that checks the similarity between two clusterings or memberships in communities. As the name indicates, adjusted mutual information is a modified form of the mutual information metric. The adjusted mutual information score for given two memberships g1, g2, $AMI(g1, g2)$ is given by Equation 3.3 (adapted from Pedregosa et al. [2011]).

$$AMI(g1, g2) = \frac{MI(g1, g2) - E(MI(g1, g2))}{avg(H(g1), H(g2)) - E(MI(g1, g2))} \tag{3.3}$$

In the above equation, $MI(g1, g2)$ corresponds to their existing mutual information value, $E(MI(g1, g2))$ represents their predictable mutual information value, $H(g1)$ and $H(g2)$ terms denote calculations of entropy values for g1, g2, respectively. Adjusted mutual information (AMI) value ranges between 0 and 1 with a value of 1, meaning both the memberships are the same.

3.1.5 The Leiden Algorithm

While identifying the right community detection algorithm, the Leiden Traag et al. [2019] has been considered for our first approach due to its efficiency in several ways. According to the authors, this algorithm is an improvement over Louvain Blondel et al. [2008] as it is faster and returns strongly connected communities, and importantly, without any disconnected communities. The workings of the Leiden algorithm in the paper Traag et al. [2019] are explained as follows: It is a three-step algorithm, as portrayed in Figure 3.3, local moving shown in Figure 3.3(b), refinement shown in Figure 3.3(c), and aggregation shown in Figure 3.3(d). In the Louvain algorithm, there is no refinement step.

Local Moving

Local moving is a step in which communities form by moving the graph nodes, thereby increasing the graph's modularity. We modify this step to make the Leiden algorithm faster compared to the Louvain algorithm. The Leiden algorithm randomly selects and queues each of the nodes shown in Figure 2.14(a) one at a time. Once available in the queue, we take a node from the front and assign it to a randomly selected community if the modularity increases with this addition. If we add that node to an unknown community, we shift the nodes next to it to the rear end of

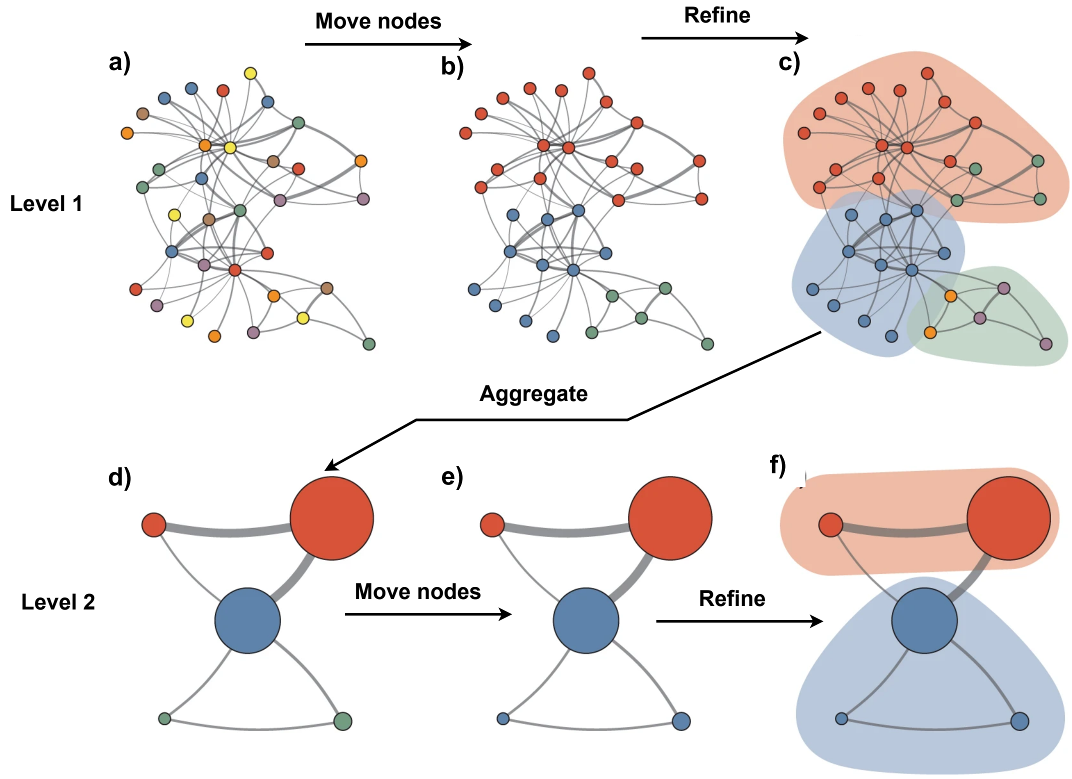


Figure 3.3: Illustration of Working of Leiden Algorithm(taken from Traag et al. [2019])

the queue. If the neighboring nodes are not already a part of the new community and are not present in the queue, this process takes place. All nodes in the queue undergo the same process. After this, the process of moving nodes is again carried out, except that only those nodes are considered for moving into new communities where their neighborhood has changed. The remaining nodes remain undisturbed. This makes the algorithm faster than Louvain, which checks all nodes for movement to new communities after the first consideration. Once this step ends, a partition or set of communities is returned by the Leiden algorithm, which represents Figure 2.14 (b), and it is used in the next step Traag et al. [2019].

Refinement

This step, as its name suggests, aims to enhance the partition obtained in the previous step. The refinement step begins with considering all the nodes from the partition as singletons or individual communities. Next, we randomly select a node and merge it with another community, but only if the modularity value increases. This merge requires two conditions: both nodes must have been strongly connected and part of the same community in the local moving partition. Considering these aspects, the step is executed for all eligible nodes, resulting in a refined version of the partition that may have some additional communities compared to the previous step, as shown in Figure 3.3 (c) Traag et al. [2019].

Aggregation

In this step, we consider the first partition from the local moving step to form the initial aggregated graph. Eventually, we form an aggregated graph by using the refined partition from the previous step. Figure 2.14(d) displays this graph, aggregating all three communities into three nodes. This figure reveals that we have two nodes, one for each of the red and green communities. The refinement step broke down the red and green communities into new communities, while the blue community remained unchanged. We carry out the local moving and refining steps for this graph, as shown in Figure 2.14 (e) and Figure 2.14 (f), respectively. These repetitions take place until an improvement in the partition's modularity value is no longer possible, and the algorithm returns the final partition Traag et al. [2019].

3.1.6 Clustering coefficients

However, we have also calculated the local clustering coefficients of the students, which we can use to produce a different set of recommendations. The process of calculating clustering coefficients involves taking the final scores of questionnaires from the transformation step and applying the normalization technique to the data. Next, we apply the non-negative matrix factorization method to the normalized data, transforming it into a single component. Similar to the community detection process, we calculate the reciprocal of the absolute differences of the non-negative matrix factorization scores for combinations of students taken two at a time. The next step generates an edge-weighted graph by using the values of absolute differences as weights for the undirected edges between the students. We again apply graph pruning to this graph to eliminate noisy edges. Finally, we calculate the clustering coefficients locally for all nodes in the graph, specifically for the students who completed the questionnaires. Its range lies between 0 and 1, where 0 means clusterability is the least and 1 means clusterability is the highest.

3.1.7 Extracting questionnaires

The MYSQL connector for Python establishes a connection to the database, allowing us to extract all the questionnaire responses through their respective IDs. These questionnaires are located in the Appendix 5.3. . We extract the six questionnaire responses into six Pandas dataframes. Each dataframe contains columns of responses given to the questions, along with a column of ID numbers unique to every student. In each questionnaire, 25 questions are present, with five options varying from strongly agree to strongly disagree, following the widely used Likert scale method for surveys. The database stores the student's selected option number and extracts it from the dataframe. When initiating the program, the administrator chooses the specific semester from which to extract data for the subsequent steps of the implementation process.

Transformation

We used Numpy and Pandas libraries to carry out the transformation of the obtained responses. We transform the values derived from the responses to maintain the correlation between the different personality traits. We keep the exact transformation criteria under wraps to guarantee that the questionnaires accurately reflect their personalities.

Nevertheless, an overview of all the steps carried out within the transformation step is mentioned in Algorithm 1. The algorithm explains that if a question receives an empty response, it deletes that particular value to avoid scoring an unanswered question. Where necessary, we replace the answered responses with new value scores and then sum up each student's scores uniquely for each questionnaire. This process results in six new dataframes, each containing the transformed scores of students from six different personality questionnaires. Finally, we merge or append all these dataframes to form a single dataframe.

Algorithm 1 Transformation of Questionnaire scores

Require: Dataframes questionnaire_responses, Dictionary{value: new_value}
Ensure: $length(questionnaire\_responses) \neq 0$

```
Initialize Dataframe all_transformed_scores
while length(questionnaire_responses) > 0 do
    Initialize Dataframe transformed_score
    while response in questionnaire_responses do
        while student_response in response do
            while value in student_response do
                if value = ∅ then
                    delete value
                else
                    replace value with new_value
                end if
            end while
            student_score ← sum(value)
            transformed_score append student_score
        end while
        all_transformed_scores append transformed_score
    end while
end while
return all_transformed_scores
```

3.1.8 Weights assignment

In the assignment of the weights part, as previously mentioned, the split of two paths for two approaches begins. We discuss all the steps involved in the implementation process for both approaches.

Normalization and NMF

Due to the variance in the nature of questionnaire scores, we apply normalization for both the first and second approaches instead of directly building edge-weighted graphs from the transformed scores in the previous step. For the implementation, existing methods from the scikit-learn library Pedregosa et al. [2011] have been used.

Also, a method from the same libraryPedregosa et al. [2011] is thought about for using the non-negative matrix factorization method on normalized data in the clustering coefficients approach as well. Incorporating the Non-Negative Matrix Factorization technique into the pipeline is an essential step in reducing the dimensions from six questionnaires to one. Thus, the calculation of weights uses the resulting dataframe for one component. The community detection approach also uses the normalized dataframe for weight calculation.

Algorithm 2 Generation of Personality Weighted Graphs

Require: Dataframe edge_weights
Ensure: $length(edge\_weights) \neq 0$

Initialize Graphs all_personality_graphs
while $length(edge\_weights) > 0$ **do**
    Initialize Dataframe personality_data, Graph personality_graph
    personality_data(source, target) = split(edge_weights[0])
    for $weight$ in $edge\_weights[1:]$ **do**
        personality_data merge weight
        personality_graph = Graph(personality_data)
    end for
    all_personality_graphs append personality_graph
end while
return $all\_personality\_graphs$

Edge weights calculation

The next step in the pipeline for both approaches has been to calculate edge weights between two students at a time by considering their scaled scores. As we intend to build undirected weighted graphs, we can assign an edge between two nodes; any of those can be taken as a source, and the other node becomes the target. With this, the sequence of steps to calculate the weights is presented in algorithm 2.

Firstly, the scaled scores are transposed for simpler processing of the next steps. Next, as illustrated in the algorithm, two students, supposedly student_a, and student_b are taken, and using their scores through the combinations method available in Python, an edge weight between them is derived from their scores. The formula for calculating the weight is taken from Puga et al. [2021] in which the absolute difference of scores is calculated first, and then its reciprocal is taken as the edge weight. We found their approach simple and efficient when assigning weight in this manner. Additionally, a modification has been included in the calculation for cases resulting in infinite edge weight. In some cases, when two students have the same scores, the formula directly produce an edge weight value of infinity, which is highest. Even though this is correct programmatically, it results in an error in the later steps. So, to avoid this error, a value higher than all the scores in the dataframe as highest_weight is used to replace the resulting infinity value wherever it occurs. The loop is executed for all six personality scores in the case of the first approach and one component score for the second approach. Thus, we obtain the dataframes with final edge weights.

Personality weighted graphs

Graphs are the final data structures that we use to detect communities and calculate clustering coefficients. In this implementation step, we generate six weighted graphs corresponding to six questionnaires that represent the different personality traits of students for the community detection approach. Similarly, for the clustering coefficients approach, we generate a single weighted graph. Algorithm 3 describes the process of generating the graphs which is identical for both approaches.

Algorithm 3 Edge-Weights Calculation

Require: Dataframe transposed_scaled_scores, highest_weight
Ensure: $length(transposed\_scaled\_scores) \neq 0$
Initialize Dataframe edge_weights
while $length(transposed\_scaled\_scores) > 0$ **do**
  while $student\_a$, $student\_b$ in $combinations(transposed\_scaled\_scores, 2)$ **do**
    edge_weights append $\{1 \ / \ |student\_a.score - student\_b.score| \ \}$
    if $edge\_weight$ in $edge\_weights = \infty$ **then**
      replace edge_weight with highest_weight
    end if
  end while
  edge_weights = transpose(edge_weights)
end while
return $edge\_weights$

The edge weights dataframe from the previous step has the first column, which has the ids of two students, and the remaining column(s) are the weights corresponding to personality traits. So, in order to build a graph, we take the first column from the dataframe and use the split method to obtain source and target nodes for the graphs. In the next step, we merge a weight column with the source and target columns. Lastly, a graph will be created representing the ties between students and the variation in the strength of ties through the weights assigned to them. This procedure is repeated for all the personality graphs. For creating the weighted graphs efficiently, networkx Hagberg et al. [2008] with extensive available documentation has been used to create graphs. However, for carrying out graph pruning and community detection, it was found that the igraph Csardi et al. [2006] is the most supported package. Hence, from these graphs, corresponding igraph graphs are created.

3.1.9 Implementation Pipeline

This section explains all the steps involved in implementing this work. Students have filled out personality questionnaires, transforming the raw responses into scores that could potentially group students.

After this step, as can be observed from Figure 3.4, there is a split into two paths that lead to the two approaches we consider for the recommendation system. For the main approach, we normalize the transformed scores. We calculate the edge weights using the normalized scores of the students. We create edge-weighted graphs based on the calculated edge weight. Later, the graphs are pruned by removing the least significant

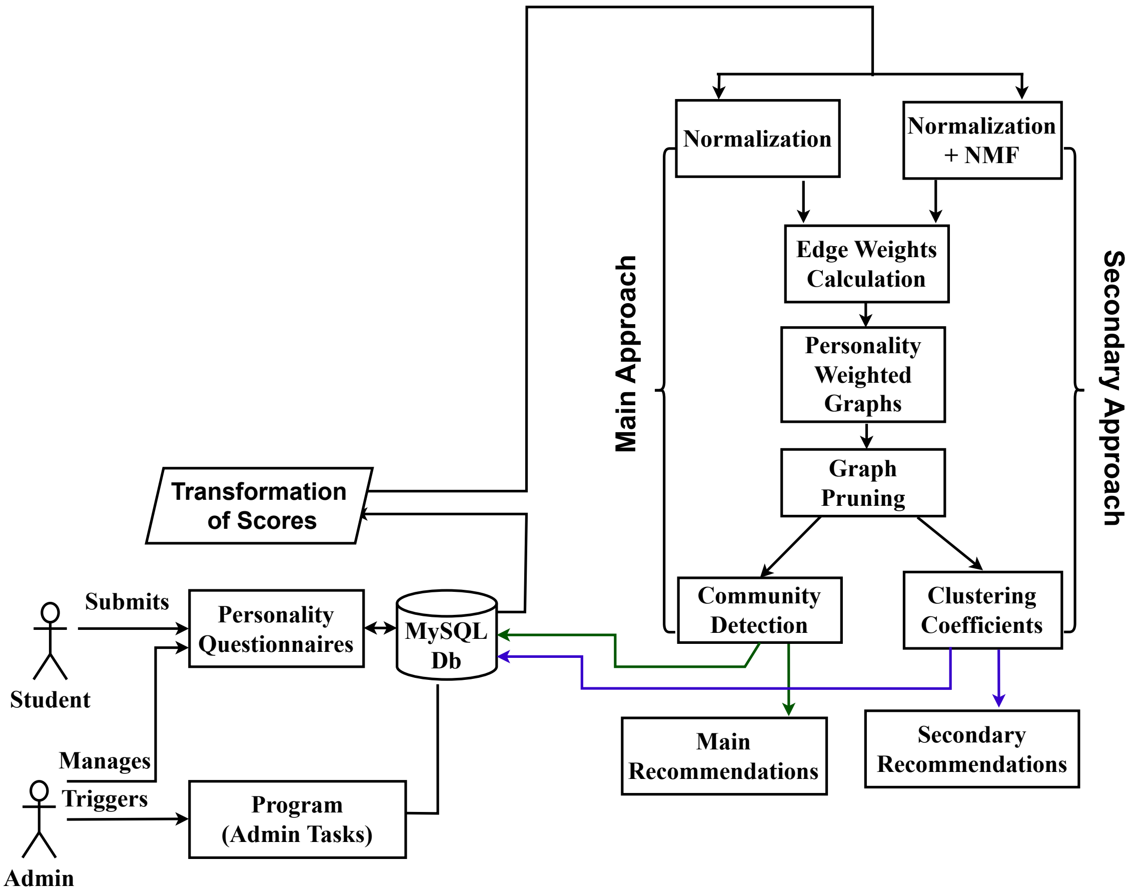


Figure 3.4: Workflow of the partner recommendation system

edges, and then, through the Leiden community detection algorithm Traag et al. [2019], students are grouped into communities. The alternate approach normalizes the transformed scores and applies non-negative matrix factorization (NMF) to reduce the dimensions of the questionnaire data. The succeeding three steps in this approach are similar to the previous approach. We calculate the clustering coefficients of students based on the pruned graph. Finally, we detect both student communities using the Leiden algorithm, store the clustering coefficients in the database, and recommend each student separately on two different pages, referred to as the main and alternate recommendations, respectively.

3.1.10 Detecting communities

Various algorithms are available for detecting communities via the igraph package. We tested some of the widely popular algorithms on the graphs to group the students before finalizing the one that best fits our purpose. In this context, we have incorporated the Leiden algorithm as our primary method for identifying communities. The algorithm's Python version package was available for installation along with well-written documentation[1] that was very beneficial for carrying out this implementation, and their efforts for maintaining the package are appreciated.

1 https://leidenalg.readthedocs.io/en/stable/intro.html

As previously mentioned, the task involves detecting communities based on all personality traits. For this purpose, the find_partition_multiplex method available in the Leiden algorithm has been adapted in this work for the purpose of generating recommendations. This method accepts a list of graphs as input and detects communities that are based on all the provided graphs. Alternatively, the find_partition method is also available and can be used to get separate communities for each graph. Furthermore, in order to have the flexibility of forming groups of different sizes, parameters such as max_comm_size and partition method were included as options for selection inside the web application. We considered having options for community sizes between 3 and 6, while the partition methods were modularity vertex partitions and surprise vertex partitions. A variety of partition methods are available in the package, but considering the nature of graphs, these two are relevant for use in this setting. We used the six personality graphs, derived and trimmed in the previous steps, for testing. We detect communities by considering both the single-layer graphs and multiplex networks methods. A detailed analogy of the results derived through the application of these methods and different parameters is presented in Section 3.1.11.

3.1.11 Evaluation and Discussion

For the recommendations of study partners to be efficient, the underlying community detection algorithm should create communities of decent quality. To carry out the evaluation, the questionnaire data filled by 60 students were considered for the study. This chapter discusses the evaluation process carried out in two phases: one is comparing the community detection results without applying the graph pruning technique to the data, and another is applying it.

In general, the Leiden algorithm's Traag et al. [2019] vertex partition creates the communities for each phase's unique questionnaire graphs. The first thing that is done is a comparison of its results to those of other algorithms using modularity scores. This step aids in selecting and substantiating the most appropriate and effective community detection algorithm, from which we can recommend communities. Next, we look at the modularity scores of the communities made by the Leiden algorithm through modularity vertex partition and surprise vertex partition for graphs of different sizes. We do this for each individual graph. As with the last case, we also compare the communities made by the Leiden algorithm using modularity vertex partition and surprise vertex partition for different sizes and types of multiplex networks or combined questionnaire graphs by looking at their improvement scores. Lastly, a similarity check only occurs in the second phase with graph-pruned data, using adjusted mutual information between the communities generated by different algorithms and vertex partition methods. Note that every community detection algorithm aims to maximize the modularity metric, a widely used metric that ranges from -1 to 1, with 1 being the highest. On the other hand, adjusted mutual information ranges between 0 and 1, where 1 means the communities generated with different algorithms or methods are identical or the same.

3.1.12 Without Graph Pruning Technique

The first phase of the evaluation, which does not apply the graph pruning technique, involves considering the questionnaire graphs with all of their existing weighted edges.

In this regard, the graphs consisted of 60 nodes, 1770 edges, average and max degree values of 59, and a density of 1, as shown in Figure 3.5. Even though such densely connected graphs make it difficult to identify strong community structures, it will be a useful step to examine the way algorithms perform on both noisy graphs and noiseless graphs.

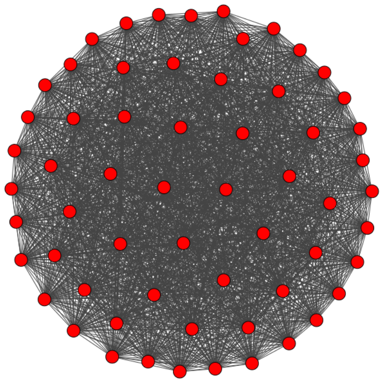


Figure 3.5: Without Graph Pruning Technique

3.1.13 Comparison of community detection algorithms

For graphs containing the above-mentioned features, a modularity scores comparison amongst six algorithms is carried out and shown in Table 3.1.

Questionnaire	**Fast Greedy**	**Infomap**	**Label Propagation**	**Leiden**	**Multilevel**	**Walktrap**
Extrovert or Introvert	**0.336**	0	0.327	**0.336**	**0.336**	0.328
Optimist or Pessimist	0.372	0.371	0.358	**0.373**	**0.373**	0.366
Tough-minded or Tender-minded	**0.299**	0	0	**0.299**	0.298	0.291
Managing People and Resources	**0.322**	0	0.274	**0.322**	**0.322**	**0.322**
Communicating and Role	**0.280**	0	0.259	**0.280**	**0.280**	0.261
Success and Risk	0.310	0	0.314	**0.320**	**0.320**	0.316

Table 3.1: Comparison of community detection algorithms for Modularity scores on individual questionnaire graphs before applying graph pruning technique

The table clearly shows that the modularity vertex partition of the Leiden algorithm led to the highest modularity score across all six questionnaire graphs. In addition to the Leiden algorithm, the Multilevel and Fast Greedy algorithms also achieved the highest score on 5 and 4 questionnaire graphs, respectively. Apart from these algorithms, the Walktrap algorithm equaled the high modularity score for one graph. On the contrary, the label propagation algorithm scored 0 for modularity for one of the graphs, and the infomap algorithm attained 0 for five questionnaire graphs. Such results with null modularity values indicate that the graphs contain noisy edges, leading to the failure of widely used algorithms like Infomap to detect strongly connected communities. In addition to the comparison of modularity scores, from Table 3.2 total communities detected by them indicate the same pattern that algorithms had not been able to detect communities properly.

Questionnaire	Fast Greedy	Infomap	Label Propagation	Leiden	Multilevel	Walktrap
Extrovert or Introvert	4	1	4	4	4	3
Optimist or Pessimist	3	3	3	3	3	3
Tough-minded or Tender-minded	3	1	1	3	3	4
Managing People and Resources	3	1	2	3	3	3
Communicating and Role	3	1	6	3	3	6
Success and Risk	4	1	4	4	4	4

Table 3.2: Comparison of the number of communities created by community detection algorithms on individual questionnaire graphs before applying graph pruning technique

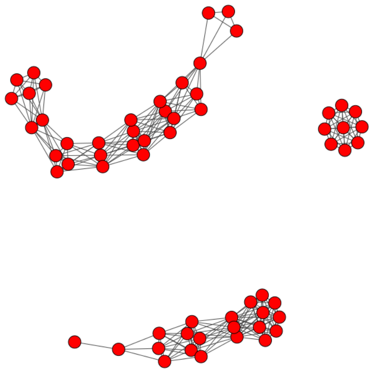


Figure 3.6: Communicating and Role questionnaire graph

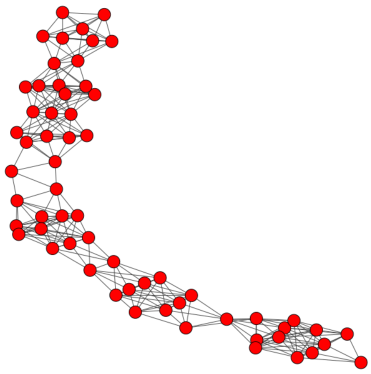


Figure 3.7: Optimist or Pessimist questionnaire graph

It is especially noteworthy that the Infomap algorithm grouped all the nodes into one community for five graphs. Likewise, the label propagation algorithm also grouped all the nodes into one community per graph. However, most other algorithms were only able to identify three or four communities for each graph. Moreover, Leiden and multilevel algorithms detected the same number of communities for all the graphs. The results show that the Leiden algorithm performed slightly superior to the others on noisy graphs for the modularity metric and prove that it is the right choice for generating recommendations through its detected communities.

However, these results clearly demonstrate that, based on noisy graphs, forming communities for the purpose of recommending study partners is not a beneficial approach. In the interest of elevating the efficacy of Leiden and recommending relevant study partners, it is necessary to remove noisy edges from the graphs. The next section of this chapter discusses the second phase of evaluation and the results after applying the graph pruning technique to remove noisy edges from the graphs.

3.1.14 Graph Pruning

Graph pruning refers to a technique that is intended to discover the important regions overshadowed by edges that are considered noisy or less significant to the graph and need to be removed Dianati [2016].

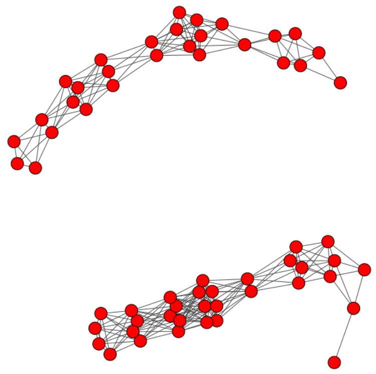


Figure 3.8: Tough and Tender questionnaire graph

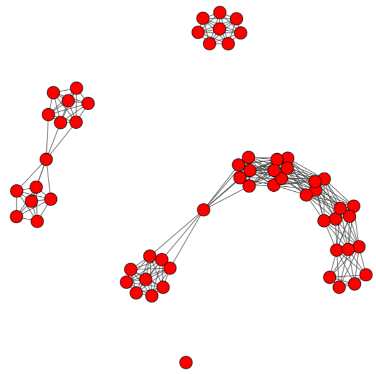


Figure 3.9: Success and Risk questionnaire graph

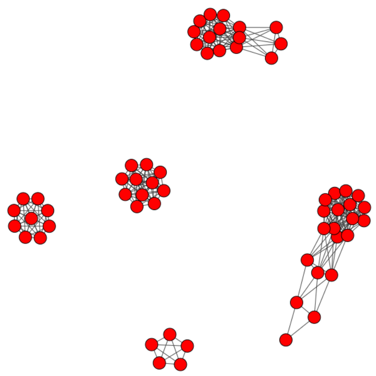


Figure 3.10: People and resource management questionnaire graph

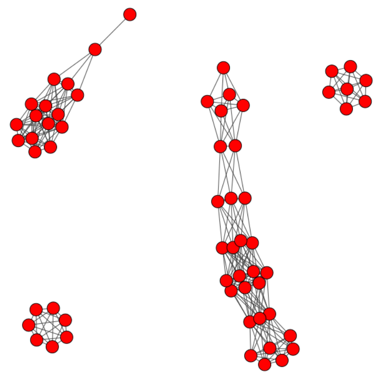


Figure 3.11: Extrovert and Introvert questionnaire graph

In our case, the graphs derived from personality questionnaires become complete, weighted, and highly dense, which in turn reduces the possibility of identifying optimal communities or clusters through algorithms. To handle this problem, the application of graph pruning is a necessity. In this regard, a graph pruning technique through a filter named Maximum Likelihood Filter (MLF) proposed by Dianati [2016] is preferred for its efficiency and better time complexity and has been used in both our approaches.

According to Dianati [2016], MLF works by using a null model graph to figure out a p-value for each edge using a marginal probability distribution. Once this is done, only edges with a p-value less than the chosen threshold value will be kept. The author further stated that the null model graph is created while ensuring that the original graph's total weight and node strength of all the nodes are present in the null model graph. Also, it is mentioned that the filter will return a graph with significant edges, as whatever edges are retained have the least possibility of being formed randomly during the marginal distribution calculation Dianati [2016].

3.1.15 With Graph Pruning Technique

Technically, this step retains the most significant edges from all six questionnaire graphs and drops the least significant edges, thereby improving the cohesiveness of the detected communities. Figure 3.6,3.7,3.8,3.9,3.10 and 3.11 show the questionnaire graph after applying the graph prunning technique.

The graph pruning technique through a maximum likelihood filter with different edge retention percentages has been applied separately to all the noisy graphs, and after analyzing the results, it was found that when 15% significant edges were retained, the algorithms showed the best possible modularity and adjusted mutual information scores. Hence, all the results discussed in this section are applicable to our 15% significant edge retained graphs.

Questionnaire	**Nodes**	**Edges**	**Average Degree**	**Max Degree**	**Density**
Extrovert or Introvert	60	274	9.133	15	0.154
Optimist or Pessimist	60	269	8.966	13	0.151
Tough-minded or Tender-minded	60	266	8.866	15	0.150
Managing People and Resources	60	293	9.766	16	0.165
Communicating and Role	60	266	8.866	15	0.150
Success and Risk	60	266	8.866	14	0.150

Table 3.3: Properties of individual questionnaire graphs after applying graph pruning technique

Table 3.3 shows the properties of all six questionnaire graphs after the application of graph pruning to retain 15% significant edges. Consequently, in comparison to the previously stated properties of graphs in Section 3.1.12, the edges are reduced thereby the degrees are also significantly lowered, and overall the graphs are less densely connected.

3.1.16 Comparison of community detection algorithms

From Table 3.4, it is clearly observable that the application of graph pruning improved the modularity scores of all the algorithms. The Leiden algorithm again achieved the

Questionnaire	Fast Greedy	Infomap	Label Propagation	Leiden	Multilevel	Walktrap
Extrovert or Introvert	0.730	0.730	0.666	**0.733**	**0.733**	0.655
Optimist or Pessimist	0.692	0.691	0.663	**0.694**	0.680	0.688
Tough-minded or Tender-minded	0.572	**0.621**	0.592	**0.621**	0.572	0.545
Managing People and Resources	**0.752**	0.749	0.749	**0.752**	**0.752**	0.747
Communicating and Role	0.688	**0.705**	0.670	**0.705**	0.693	0.683
Success and Risk	0.707	0.702	0.691	**0.710**	0.705	0.465

Table 3.4: Comparison of community detection algorithms for Modularity scores on individual questionnaire graphs after applying graph pruning technique

highest modularity score for all six questionnaire graphs. The multilevel algorithm achieved the highest modularity scores for two graphs. A drastic difference is visible in the performance of the Infomap algorithm. This algorithm previously showed modularity scores of 0 for most of the graphs, as seen in Table 4.5. In contrast, after the graph pruning, this algorithm achieved decent non-zero modularity scores. Specifically, the Infomap algorithm on the pruned graphs, along with the Leiden algorithm, leveled the modularity scores for 2 out of 6 graphs.

Questionnaire	Fast Greedy	Infomap	Label Propagation	Leiden	Multilevel	Walktrap
Extrovert or Introvert	6	7	9	6	6	4
Optimist or Pessimist	5	6	7	5	5	6
Tough-minded or Tender-minded	4	6	8	6	4	9
Managing People and Resources	6	7	7	6	6	6
Communicating and Role	6	7	7	7	6	6
Success and Risk	7	8	8	7	7	5

Table 3.5: Comparison of communities created via community detection algorithms with individual questionnaire graphs after applying graph pruning technique

Analogous to the changes in the modularity scores, the detected communities count as well displayed positive changes, as presented in Table 3.5. It is evident from this table that there is an increase in the detected communities for all the algorithms. It seems that the graph pruning has removed all the least significant edges present previously which made the algorithms form weakly connected and lesser communities with low modularities. Consequently, the communities now have strong internal ties, and weak external ties, and thereby the changes in the community structure led to more communities.

Next, an attempt to understand the similarities between the communities generated by the algorithms is discussed.

3.1.17 Leiden algorithm's AMI score similarity check with other algorithms

If an algorithm delivered good modularity value, it would not be an ideal algorithm to prefer if it detects communities that are completely different in comparison to other algorithms. Hence, in this subsection, similarities between the communities generated by the Leiden algorithm and communities generated by other algorithms are checked through adjusted mutual information score. Similarity checks are carried out on a one-to-one basis meaning that at a time communities of the Leiden algorithm are compared with communities of only 1 of the other 5 algorithms. For each questionnaire graph, the Leiden algorithm's communities are compared with the communities of other algorithms, and their adjusted mutual information scores are reported in Table 3.6.

Questionnaire	Fast Greedy	Infomap	Label Propagation	Multilevel	Walktrap
Extrovert or Introvert	0.844	0.947	0.745	**1**	0.569
Optimist or Pessimist	**0.958**	0.878	0.818	0.887	0.851
Tough-minded or Tender-minded	0.684	**0.907**	0.733	0.684	0.569
Managing People and Resources	**1**	0.932	0.898	0.966	0.898
Communicating and Role	0.785	**0.829**	0.719	0.725	0.651
Success and Risk	**0.958**	0.858	0.894	0.803	0.434

Table 3.6: Adjusted Mutual Information score similarity of communities formed by Leiden algorithm with other community detection algorithms on individual questionnaire graphs after applying graph pruning technique

Table 3.6 shows that the communities found by all the algorithms are similar. Some of the scores for adjusted mutual information are equal to 1, which means that those algorithms found communities that were exactly the same as the communities found by the Leiden algorithm. In addition, we achieved a few scores that were close to 1. We can clearly discern a pattern in the algorithms, which consistently yield similar results across all the different comparisons conducted. In other words, the Multilevel and the Infomap algorithms previously achieved high modularity scores for some of the graphs, along with the Leiden algorithm. Likewise, even in this similarity check, the Multilevel and Infomap algorithms got high adjusted mutual information scores for two graphs. In contrast to the modularity score pattern, the Fast Greedy algorithm achieved high adjusted mutual information scores for 3 graphs. This subsection's analysis reveals that the Leiden algorithm, along with other similar algorithms, accurately detects communities. Similar to the first phase, the next subsection analyzes the modularity scores of the Leiden algorithm for various community sizes.

3.1.18 Comparison of modularity scores of the Leiden algorithm for different community sizes

The results achieved so far after the application of the graph pruning technique on the questionnaire graphs demonstrated significant improvements in communities detected by all the algorithms.

Table 3.7 shows the pre-graph pruning phase, where negative modularity scores were obtained for all the graphs, irrespective of the size. In contrast, Table 3.8 shows the modularity scores of the Leiden algorithm through modularity vertex partition for different community sizes improved significantly after graph prunning. Also, we see from the table that the algorithm managed to achieve positive modularity scores for all the specified community sizes between 3 and 6. For every graph, when a community size of 6 is specified, the algorithm gives a high modularity score compared

Questionnaire	Size 3	Size 4	Size 5	Size 6
Extrovert or Introvert	-0.015	-0.015	-0.015	-0.015
Optimist or Pessimist	-0.015	-0.015	-0.015	-0.015
Tough-minded or Tender-minded	-0.015	-0.015	-0.015	-0.015
Managing People and Resources	-0.015	-0.015	-0.015	-0.015
Communicating and Role	-0.016	-0.016	-0.016	-0.016
Success and Risk	-0.015	-0.015	-0.015	-0.015

Table 3.7: Comparison of Modularity scores of Leiden algorithm through Modularity Vertex Partition for different community sizes on individual questionnaire graphs before applying graph pruning technique

Questionnaire	Size 3	Size 4	Size 5	Size 6
Extrovert or Introvert	0.150	0.226	0.286	**0.354**
Optimist or Pessimist	0.162	0.243	0.302	**0.353**
Tough-minded or Tender-minded	0.153	0.222	0.312	**0.377**
Managing People and Resources	0.138	0.209	0.276	**0.333**
Communicating and Role	0.161	0.234	0.293	**0.368**
Success and Risk	0.150	0.232	0.305	**0.381**

Table 3.8: Comparison of Modularity scores of Leiden algorithm through Modularity Vertex Partition for different community sizes on individual questionnaire graphs after applying graph pruning technique

to the other three sizes. We can directly say by observing the table that when a high community size is specified, the modularity scores will possibly be higher than the sizes lower than it. Also, an indirect inference is that when community sizes are not specified, the algorithm tries to freely group more nodes into each community based on the strength of their cohesion. Therefore, Table 3.4 about the Leiden algorithm without a specified community size has better modularity values than the Leiden algorithm for all the specified community sizes in Table 3.8.

However, for the purpose of recommending study partners to a student for team formation, a tradeoff is to be made between the modularity score and communities of desirable sizes. In this regard, obtaining communities of desirable sizes is prioritized over achieving the highest possible modularity scores. It is also to be highlighted that obtaining negative modularity values is not desirable. Another aspect of the comparison in the Leiden algorithm besides the community sizes is the type of vertex partition method. Both modularity vertex partition and surprise vertex partition are compared in this phase again, as mentioned previously, to decide the best method to use in our pipeline.

Results of the Leiden algorithm's surprise vertex partition method are mentioned in Table 3.9. Similar to the first phase where the modularity vertex partition method recorded a pattern of negative modularity scores, the surprise vertex partition method in this phase achieved positive modularity scores that were comparable to the modularity vertex partition method's scores. Moreover, the surprise vertex partition method also achieved a high modularity score for all the graphs when a community size of 6 is specified. Both methods performed equally well in terms of modularity scores. Analysis of the communities formed by both methods revealed

Questionnaire	Size 3	Size 4	Size 5	Size 6
Extrovert or Introvert	0.143	0.221	0.295	**0.367**
Optimist or Pessimist	0.155	0.245	0.311	**0.374**
Tough-minded or Tender-minded	0.141	0.238	0.291	**0.350**
Managing People and Resources	0.137	0.205	0.288	**0.333**
Communicating and Role	0.161	0.217	0.303	**0.370**
Success and Risk	0.152	0.225	0.305	**0.367**

Table 3.9: Comparison of Modularity scores of Leiden algorithm through Surprise Vertex Partition for different community sizes on individual questionnaire graphs after applying graph pruning technique

that the surprise vertex partition method formed more singleton communities than the modularity vertex partition method. In other words, the surprise vertex partition method did not group certain nodes into any communities, instead leaving each node as an individual community. Additionally, the surprise vertex partition method created numerous communities that were smaller than the specified community size. Such deviations are not desirable, as we want all or a maximum of students to be grouped into communities of specified sizes, avoiding students being singled out or being part of small groups. As a result, the modularity vertex partition method is a slightly more preferable choice to select for our community detection.

3.1.19 Leiden algorithm's AMI score similarity check for different community sizes

An analysis of similarities between communities detected by both the modularity vertex partition and surprise vertex partition methods is performed through adjusted mutual information scores, as in the case of a similarity check between communities detected by the Leiden algorithm and other algorithms. In Table 3.10 and Table 3.11 adjusted mutual information scores for communities detected through the Leiden algorithm's modularity vertex partition and surprise vertex partition methods are listed for individual graphs and multiplex networks, respectively. From the tables, it is to be observed that a direct correlation does not exist between community sizes and adjusted mutual information scores. However, we can deduce that when community sizes are high, the communities identified by both partition methods tend to exhibit greater similarity, as evidenced by their elevated adjusted mutual information scores. It is to be pointed out from Table 3.10 that the highest value of adjusted mutual information for individual graphs is 0.918, while for multiplex networks it is only 0.401, as mentioned in Table 3.11. It indicates that the communities detected through both methods of the Leiden algorithm are dissimilar to a large extent.

These results differ from those obtained for the adjusted mutual information score similarity check between the Leiden algorithm through modularity vertex partition and other algorithms, which were conducted without a specified community size. In that case, some of the similarity comparisons yielded adjusted mutual information scores as high as 1. Perhaps the detection of communities through the surprise vertex partition, which relies on the asymptotic surprise metric, leads to significant differences in the community structure compared to the modularity vertex partition, which utilizes the modularity metric as its basis. We must reiterate that the surprise

Questionnaire	Size 3	Size 4	Size 5	Size 6
Extrovert or Introvert	0.530	**0.561**	0.552	0.554
Optimist or Pessimist	0.272	0.256	0.393	**0.464**
Tough-minded or Tender-minded	0.274	0.373	0.457	**0.551**
Managing People and Resources	0.716	0.586	0.784	**0.918**
Communicating and Role	0.631	0.633	**0.777**	0.752
Success and Risk	0.361	0.489	**0.655**	0.590

Table 3.10: Adjusted Mutual Information score similarity of communities formed by Leiden algorithm's Modularity Vertex Partition with Surprise Vertex Partition on individual questionnaire graphs after applying graph pruning technique

Membership	AMI
Size 3	0.204
Size 4	0.311
Size 5	**0.401**
Size 6	0.265

Table 3.11: Adjusted Mutual Information score similarity of communities formed by Leiden algorithm's Modularity Vertex Partition with Surprise Vertex Partition on combined questionnaire graphs (Multiplex Network) after applying graph pruning technique

vertex partition method's high occurrence of singleton communities could be another potential reason for the differences. In this regard, we prefer to consider detecting communities through the modularity vertex partition method as the best applicable method for our work.

After developing a strategy that provides a thorough and precise method for identifying distinct groups within a network, we proceed to explain how we enable online collaboration between instructors and students on the team platform. The platform is an integral part of the SQLValidator and affords us the capability to administer team-oriented exercises and tests to assess students' SQL programming skills. One of the objectives of the teams' platform is to facilitate high-level discussions, which are similar in quality to discussions that take place in traditional collaboration settings. The next section describes this platform.

3.2 Teams Platform

Early implementations of team-based learning showed that collaborative problem-solving within small groups was effective in stimulating active learning Gomez et al. [2010]; Michaelsen et al. [2004]. As observed in Michaelsen et al. [2004], team members assumed specific roles in an effort to efficiently solve the assigned tasks. Despite the ineffectiveness of most team members in their assigned roles, team leaders assumed responsibility for their peers' learning. This challenge of fitting team members into defined roles still persists in recent traditional lecture settings Michaelsen and Sweet [2011]. However, the structure in which an action occurs to a greater extent facilitates the eventual behavior or how individuals respond, so we devised the *Teams subsystem* of SQLValidator to facilitate task reflection and the acquisition of collaborative problem-solving skills. Our initial results show that, when students collaborate, their solutions to project tasks are on average better than those without visible collaboration within a team. Furthermore, our investigation of different degrees of instructional design shows that an explicit demand for team self-organization helps in completing collaborative tasks more efficiently. We will discuss these in the upcoming sections.

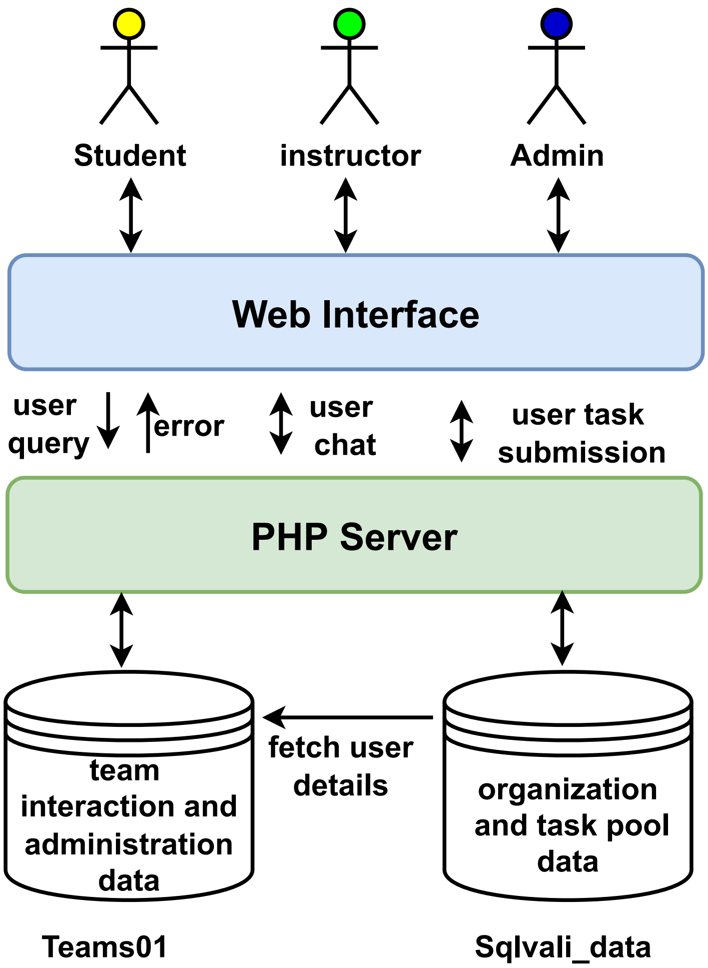


Figure 3.12: Teams Architecture

There are three main features when implementing a web-based application; these are centralization, replication, and distribution. The Teams' system uses a centralized client-server architecture. The general architecture of the application has been depicted in Fig: 3.12. As depicted, user interactions via a web interface by way of posting chats, creating submissions in the group wiki, and executing queries in the editor, etc. are mediated by a PHP server. The relational database management system is tasked with storing and managing all the data resulting from student and instructor interaction. To achieve this objective, the Teams' platform interacts with two main databases:

- `db2_data` contains all relevant data to maintain the organization of the platform itself, such as user management and task definitions.

- `db1_teams01` contains all standard tables and data used to support project task submissions, chat management, user query evaluations, and admin management.

Thus records of all user interaction are stored for analytical purposes. In the net section, we give a description of the team environment.

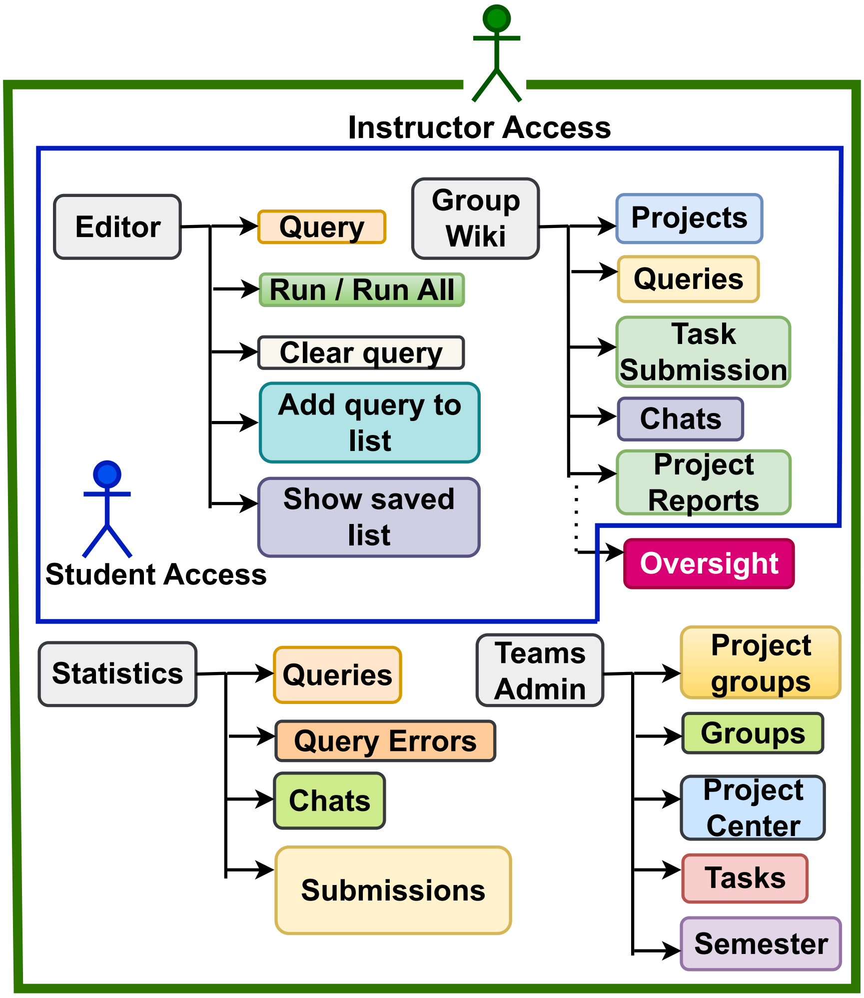


Figure 3.13: Teams Overview

3.2.1 Student Access

As communication is often an integral feature of collaboration tools, we made the chat persistent in the group wiki and code editor pages. Students have the option of updating and deleting their chat posts. To keep teams focused on current milestones, we structured the chat in pages. Only the latest 10 chat posts are visible. History buttons are available to provide access to previous chat posts. Our strategy aims to instill a culture of self-assessment and reflection, and, thus, we allow teams to improve their solutions and resubmit again. This feature is accessible in the group wiki/task submissions. Unlike the chat system, where team members can edit and delete their chat posts, team members can only create, read, update, but not delete task submissions. Since many tasks are based on the Structured Query Language SQL, our platform includes a query editor. The editor, apart from executing queries,

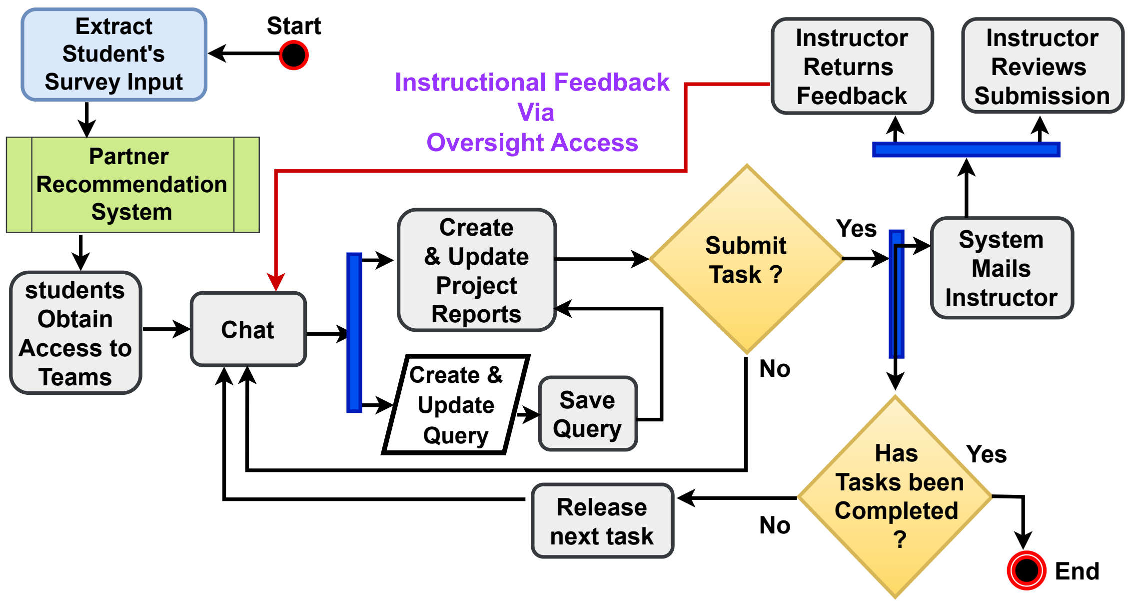


Figure 3.14: Teams Workflow

allows collaborators to store previously used queries. Thus, if in the course of the milestones it is required to alter the solutions, and hence the queries, the team can access all their previous queries from the group wiki. It also allows selected execution of related queries. The group wiki gives them access to the project tasks, saved queries, and submission pages.

3.2.2 Instructor Access

The instructor, apart from having access to administrative activities, can grant itself membership in any team where his feedback is required. This is facilitated via the oversight access shown in Fig. 3.14. Thus, the instructor can perform CRUD operations on chat posts and task submissions. However, all the queries executed in the instructor profile are not transferred to the team profile. In general, the oversight feature facilitates the integration of instructional feedback, which is typical for traditional team project interactions. The teams overview diagram further (Fig. 3.13) shows the other activities specific to administrators in Our Teams' platform.

3.2.3 Teams Workflow

Given that individual students have completed the personality survey, teams are generated via the partner recommendation system, which is described in Section **??**, the administrator loads several tasks into each team profile and initializes the teams. The student members gain access, introduce themselves, and immediately start interacting with the tasks in the group wiki. The interaction will result in chat and project report commits, and they agree that the answer to a respective question, the project report, is updated again, after which a submission is made. Once the first task is solved and submitted, the next task is activated. This process continues until the last task is activated. Once a submission event is registered, the system mails the respective instructor, and the review process starts. Once the review is done, the instructor, via the oversight link, gives a response in the teams chat. Depending on the response, the entry in the project report will either be updated or left as the final response to the respective question. This process will continue until the final task. A description of the task is shown in Section 3.2.7.1

3.2.4 Survey insights

Being that the team projects were developed to stimulate the cultivation of collaborative skills Obionwu et al. [2023f], having systems and structures generate collaborative behaviors alone is insufficient Erbguth et al. [2022]. Collaboration and team engagement as a feature can be utilized to help learners coordinate and communicate effectively to achieve a common goal. Thus, to cultivate community learning and enhance collaboration, we designed tasks to incorporate communication and not force them on the students. We further sought to gain insight into our students's psychological affinity for collaborative engagements and behavioral dispositions toward collaborative learning. To achieve these goals, we partly adapted the ”Students' Readiness for CSCL” questionnaire Xiong et al. [2015]. Eight items from this questionnaire were selected from the ”Motivation for collaborative learning” evaluation, and ten items were selected from the ”Prospective behaviors for collaborative learning” questionnaire. In the 2023 winter semester, we had 140+ enrollments in our teams' platform. Although we decided not to enforce survey participation, 95 students from those enrolled in the semester course participated in the surveys. Furthermore, we allowed the possibility of skipping sections of the questionnaire. In the next subsection, we give a description of the course participants based on the survey results.

3.2.5 Limitations

Although, through the evaluation of the Leiden algorithm on the questionnaires, it was shown that this approach to detecting communities is comparatively better than the other algorithms, it is not a perfect detection. This is because optimizing the modularity metric is an NP-Hard problem, which makes it impossible to find out the correctness of the communities. Multiple intrinsic metrics can possibly be applied in the process of detecting communities, and the results can be analyzed to approximately estimate their correctness. Another limitation in both approaches is the trade-off between attaining high modularity and the level of graph pruning. The level of pruning is decided manually only through multiple trials on the questionnaire graphs and the resulting modularity scores. The second limitation can be tackled to an extent as mentioned in the future scope.

3.2.6 Participants

The pilot study was conducted in the context of the 2021 database concept summer semester's course. To nurture a collaborative environment, students were required to form teams consisting of 3 individuals, as triads are typically more stable and engaging than other social network structures Yoon et al. [2013], and well attuned to our task structure, as will be discussed in the next subsection. To increase trust among team members, team formation took place at the beginning of the course. The students were advised to form teams of three for the theoretical part of the exercises, where all lecture material has been practiced before starting the collaborative team project. This not only fostered easy acquaintance but also willingness to deal with the team process. Furthermore, the collaborative tasks are extracted from the concepts described in the lectures and theoretical exercises; thus, the teams are expected to have acquired all the skills and information needed to engage with the collaborative tasks.

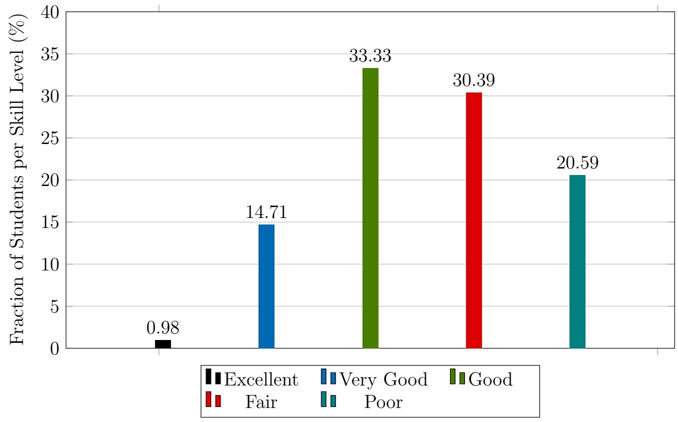


Figure 3.15: Student's Initial Practical Programming Knowledge

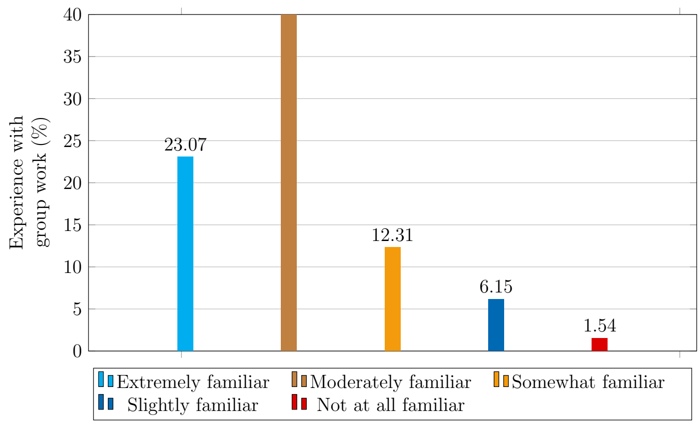


Figure 3.16: Student's group work experience

Four exercise instructors oversaw the weekly exercise meetings and helped facilitate teamwork. To estimate the participants perceptions and experiences with respect to teamwork, and collaboration, we conducted surveys. The result of our inquiry into their self-perceived practical programming knowledge is shown in Fig. 3.15. The results suggest that: about 1% had extensive experience with general programming 15% were proficient, 33% had above-average experience, while 51% had rather limited programming skills. Overall, a considerable number of the students' population were beginners, and hence we taught them the fundamentals of using SQL. Furthermore, Fig. 3.16 shows their team work experience. The results indicate that: about 80% had worked in team projects or tasks prior to enrolling in our course, while 20% had

Item	Motivation for collaborative learning	No.	Mean	SD
Mot.1	I like to work with other students in group activities.	65	2.8	1.21
Mot.2	Comparing with doing individual assignments, it is more effective to learn by doing group work.	65	2.85	1.21
Mot.3	I will need teamwork skills in my future job.	65	3.2	0.89
Mot.4	Working in groups allows me to tackle more complex topics than working individually.	65	3.05	1.08
Mot.5	There are many opportunities for discussion and sharing ideas by working in groups.	65	3.08	1.00
Mot.6	I believe I can do well in the group work.	65	3.15	0.87
Mot.7	I believe I can support group-mates.	65	3.2	0.90
Mot.8	I believe I can play an important role in the accomplishment of the group task.	65	3.02	0.89

Table 3.12: Motivation for collaborative learning Questionnaire

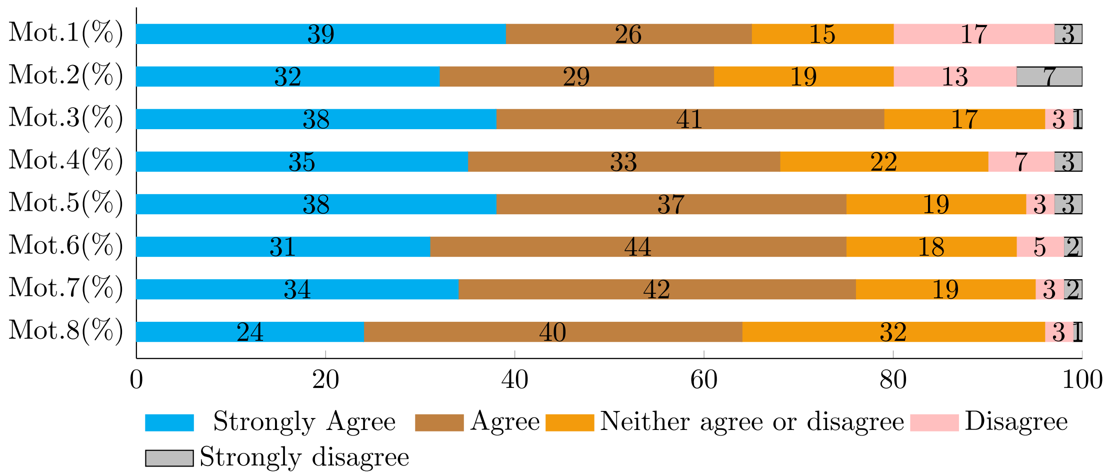


Figure 3.17: Motivation for collaborative learning feedback

limited team work experience and thus needed guidance on how to work in team projects.

3.2.7 Collaboration and Team interaction questioner description

In the first survey among the participants of the course, we aimed to elicit our participants' self-evaluation and experiences on team interaction, and collaboration. A total of 65 of the participants responded to the voluntary survey at the beginning of the course. Most of the users were between the age of 27-31, and have previously not used our collaboration platform.

We show descriptive statistics like the mean, the standard deviation for the quantitative questions rated on a 5-point Likert scale (in numeric representation: 0 = Strongly Disagree to 4 = Strongly Agree) in Table 3.12, which contains questions and responses about their motivation for teamwork, and in Table 3.13, we show their self evaluation of their collaboration behavior. The vast majority of quantitative replies were agree (3) and Strongly Agree (4) on the scale. Therefore, the standard deviations are fairly small for a vast majority of the questions. There were no questions that were answered mostly negative, but there are several questions with mixed replies. In general, questions were prepared in such a way that not only perceptions about current team collaboration, and interaction events are elicited, but also their previous collaboration and teamwork experiences, behavior, and opinions.

We observe from Table 3.12 and the corresponding plot, Fig. 3.17, that more than 65% of the participants liked working in groups (item Mot.1), and 61% agreed that it was more effective to work in groups (item Mot.2). 79% of participants in item Mot.3 held a notion that teamwork skill was important for their future job, while in item Mot.4, 68% agreed that complex tasks can easily be tackled by sharing the workload with group members. Furthermore, in item Mot.5, 75% agreed that working in groups provided opportunities for discussion and sharing of ideas, and in item Mot.6, 75% can perform well while in group work. 76% believed that they can support their group mates in item Mot.7 and in item Mot.8, 64% believed they can play an important role in the accomplishment of the assigned group task.

Item	Prospective behaviors for collaborative learning	No.	Mean	SD
Beh.1	I like to share my ideas with others.	60	3.02	0.98
Beh.2	I am open to new ideas.	60	3.27	0.84
Beh.3	I am tolerant of different ideas.	60	3.25	0.88
Beh.4	I am able to express what I think in an appropriate way, not harming other group members.	60	3.18	0.81
Beh.5	I always participate in an appropriate way.	60	3.13	0.83
Beh.6	I am able to provide feedback on overall team's performance.	60	2.9	1.00
Beh.7	I am able to provide feedback on individual team member's performance.	60	2.8	0.95
Beh.8	I am able to monitor my group's progress.	60	2.87	0.98
Beh.9	I am able to implement an appropriate conflict resolution strategy.	60	2.77	0.95
Beh.10	I am able to recognize the source of conflict confronting my group.	60	2.87	1.02

Table 3.13: Prospective behaviors for collaborative learning questionnaire

Considering Table 3.13 and corresponding feedback, Fig. 3.18, 60 students participated in this survey group of question as our survey questions are not obligatory, out of which 77% of the participants indicated that they liked to share ideas (item Beh.1), and in item Beh.2, 87% indicated that they were open to new ideas. In item Beh.3, 84% are tolerant of different ideas. 82% indicated that they can express their thoughts appropriately (item Beh.4) and 75% further indicated in item Beh.5 that they always participated appropriately during group work. In item Beh.6, 73% of the participants indicated that they were able to provide feedback on individual team member's performance, while 70% in item Beh.7 indicated that they were able to provide feedback on individual team member's performance as well as monitor their group's progress in item 8. Furthermore, in item 9, 63% indicated that they were able to

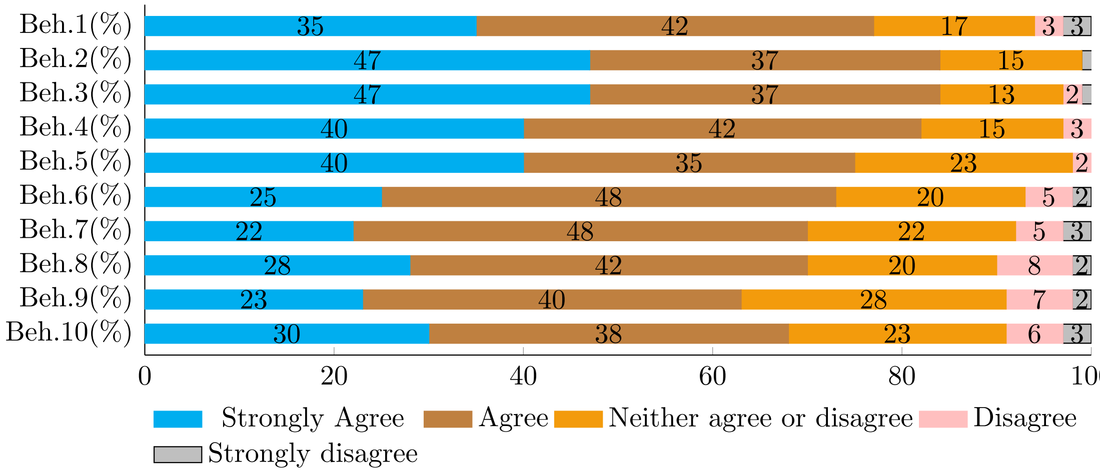


Figure 3.18: Prospective behaviors for collaborative learning feedback

implement an appropriate conflict resolution strategy and in item 10, 68% indicated that they were able to recognize the source of conflict confronting their group.

In general, the standard deviations from the mean were modest, as most of the participants indicated that they either agreed or strongly agreed with the perceptions on collaboration and teamwork that were queried about in the survey. So, in general, all participants have a positive attitude and motivation towards the expected teamwork. This is reinforced by Fig. 3.16 which showed that about 80% of the participants already experienced group work. Thus, around 20% of our participants have not experienced working in teams. Ergo, our project was a guide for this group of participants on the basics of teamwork, and collaboration.

3.2.7.1 Task description

Our collaborative tasks are based on the Structured Query Language SQL, a standard for performing CRUD operations on a database. Thus, to create a collaborative SQL project with reasonable level of complexity, we employed the concept of roles. These roles are known to affect how team members collaborate Lyons [1971]; Oke et al. [2016]; Ruch et al. [2018]; Senior [1997]. Furthermore, regulating group learning is important for learning processes and outcomes. Teams have to plan, monitor and evaluate, respectively, reflect on their teamwork - a challenging task, especially for novices in teamwork. A Collaboration Script that guide the planning, monitoring, and reflection activities can support teams Näykki et al. [2017]. Based on this, we created two conditions, structured, and unstructured projects, as we discuss in Section 3.2.8. The general task description is described in the next section.

Collaborative problem-solving facilitates not only peer knowledge transfer but also several beneficial skills, such as communication skills, teamwork, and respect for others. It also stimulates independent responsibility for learning and sharing information with teammates Hung et al. [2008]; Parker [2006]. Table 3.14 shows a summary of our task description. In general, it is team-centered, and instructors do not dictate or enforce any collaboration pattern. The overall scene that we present in the part "Introduction and Objective" is that students should follow the whole design cycle within a data management project – from use case modeling over schema design to schema definition and data analysis. To facilitate collaboration within this scenario, we define the three roles: (1) stakeholder, who is responsible for defining a complex use case and interesting analyses, (2) administrator, who should create the schema and execute the ETL process, and (3) the developer who implements the analyses. To this end, students individually and collaboratively assume responsibility for solving different aspects of the project milestones. The tasks also encourage team strategy reflection. Thus, teams have the option of re-evaluating, and resubmitting a previously submitted solution. The goal here is to induce learning strategies adjustment considerations and stimulation of self-reflection skills.

Collaborative Task Sections	
Introduction and Objective	**Motivates and stresses the importance of teamwork. Describes the expectancy of each milestone.**
Specification of Roles	**Explain the different roles to be assumed by participants of the team.**
Teams, Role Formation, & Selection	**Students form triad social units and choose either a stakeholder, an administrator, or a developer role.**
Planning and Task Sequence	**Explain the steps that teams should go through to achieve the objective.**
Description of Tasks without reflection script	**A total of six tasks from database modeling to data definition and querying.**
Description of tasks with reflection script	**In addition to the tasks with reflection script, it contains another first task, which addresses project planning and an additional last task that inquires team reflection.**
Reflection and Extension	**Encourage teams with reflection script-based tasks to think about what has been learned and how to apply that learning to different contexts.**

Table 3.14: Summary of our task description.

3.2.8 Project type

We created two project types, groups working on tasks with reflection script and teams working on tasks without reflection script groups, in order to assess the impact of instructional guidance on the extent of collaboration. The tasks with reflection script, shown in Table 4.24, had a general description of the task, which was assigned to one of the roles (responsibilities change from task to task). Furthermore, the last instruction always asked for a critical discussion inside the group.

In contrast to teams with reflection scripted tasks, teams with tasks that require reflection, Table 3.16, were required to plan their team work before the first task submission. We further described the planning process and possible discussion points. In the preceding tasks, we also described steps to take and last steps within their tasks required the teams to reflection on what they have done. With these explicit instructions, we aimed at encouraging students to collaborate and especially to regulate their teamwork more systematically.

Sample tasks without reflection script	
Task	**Description**
Task 1 ER modeling	**The stakeholder(s) designs a use case for which the data management should be done. This use case is described in a natural language formulation and an ER model is designed for it. The created and described ER-diagram should contain at least 3 entities and two relations. Please upload both in the SQLValidator and discuss whether the solution needs adjustment.**

Table 3.15: Sample tasks without reflection script

3.2.9 Analysis of the Collaborative Project

Having described the platform, task groups, and their respective tasks, we now provide a preliminary analysis of the collaborative activity. This preliminary analysis is done on the remaining 28 teams that solved the collaborative task toward the end of the summer semester 2021 (please note that some students dropped within the course and, thus, the reduced number of teams).

To structure our analyses, we derive two indicators from the teams.

First, since the SQLValidator Teams allows us to see who solved the tasks, we differentiate between the number of students submitting within the project team as an indication for collaboration. Thus, the label ”3 submitters” (13 teams) implies that each of the team members submitted at least one task, while ”2 submitters” (8 teams) and ”1 submitter” (7 teams) implies that only one or two team members did all the submissions. Hence, many teams distributed the tasks among themselves, which is a positive sign for the overall collaborative setup. Still, when taking a more in-depth look into the data, only 6 teams strictly followed the role distribution. This is a common problem that also Näykki et al. [2017] identified, as their instructions were also often disobeyed. As a result, we need to implement incentives that motivate collaboration among the students.

The second indicator is the project type (cf. Section 3.2.8) as it should have an impact on the teams' teamwork. In the charts, ”Str.” stands for groups with tasks that require reflection (13 groups) and explicit collaboration instructions, and ”Unst.” designates teams with tasks that require no reflection (15 groups) with only recommendations for collaborative practices. Notably, teams were shuffled in random into one of both project types without them knowing what task description they got.

In the following, we first analyze the skills and motivation of the teams in forms of the Moodle submission, their messaging behavior, as well as their final project grading.

Sample tasks with reflection script	
Task	**Description**
Task 0 Project Planning	**(a) Meet online in the Teams Chat. Briefly discuss the task. Are there any problems of understanding? Clarify any questions about the task. (b) Then discuss the concrete implementation: make a time plan and distribute the roles (Consideration: Do you already have experience with a certain role or do you want to strengthen your skills in a certain role?) Please also store the role distribution in the SQLValidator! (c) Also, briefly discuss what you find important about teamwork. What do you expect from your team members? As a team, write down three key points that the team members want to adhere to.**
Task 1 ER modeling	**(a) The stakeholder designs a use case for which the data management should be done. This use case is described in a natural language formulation and an ER model is designed for it. The created and described ER-diagram should contain at least 3 elements and two relations. Please upload both in the SQLValidator. (b) The two team members provide feedback on the stakeholderś solution (assessment and suggestions for improvement). Through this review process, all team members intensively deal with each task. (c) Discuss (stakeholders) the feedback with the team members and discuss how to proceed. Revise the original solution and upload the final result to SQLValidator.**

Table 3.16: Sample tasks with reflection script

3.2.10 Analysis of Team Skill and Motivation

Fig. 3.19 shows the group scores obtained from the theoretical part of the exercises that preceded the team's project. These scores range from 61 (the minimum criterion for exam qualification) to 100 and usually represent the motivation of the students and their understanding of the exercises because these points come from graded team submissions of theoretical exercise tasks.

This analysis yields two insights. First, we can see that student teams with a higher Moodle score (i.e., motivation) also tend to follow the rules of collaboration more

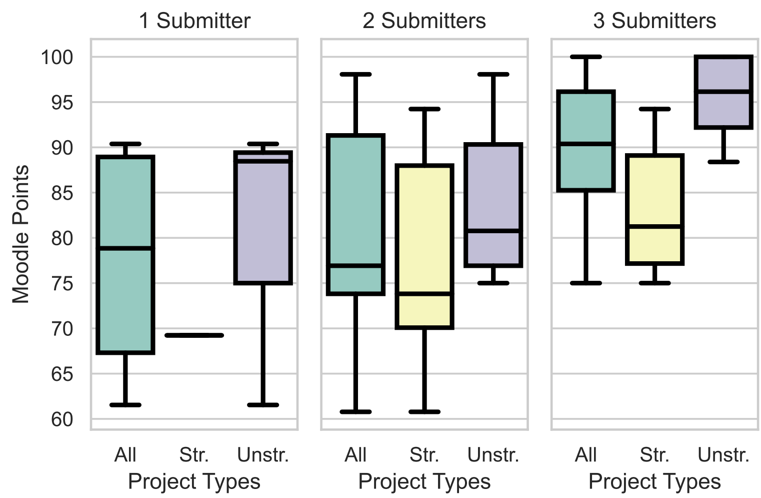


Figure 3.19: Scores in the Theoretical Exercise Submitted to Moodle (thus, Moodle Points)

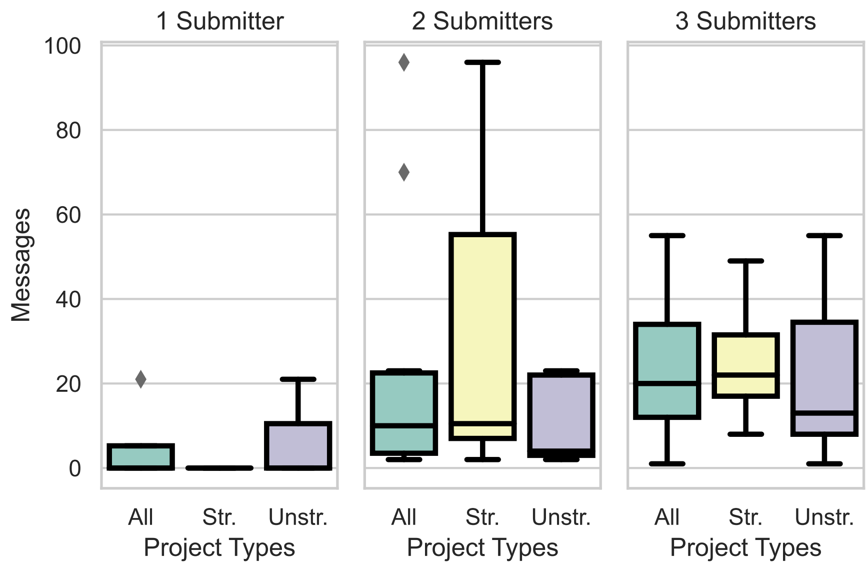


Figure 3.20: Team Messages

strictly, as we can see from the higher medians of Moodle points for the teams with two or three submitters. Secondly, our random assignment of tasks resulted in a slight bias towards team tasks that necessitated reflection. Teams with tasks that did not require reflection, on average, performed better, which could potentially impact the final score in the collaborative project.

3.2.11 Analysis of Chat Behavior

In Fig. 3.20, we show the messages sent through the integrated chat system. A positive result of this analysis is that when collaboration happens (i.e., two or three people submit tasks), teams working on tasks with a reflection requirement make more use of the integrated chats than teams working on tasks with no reflection

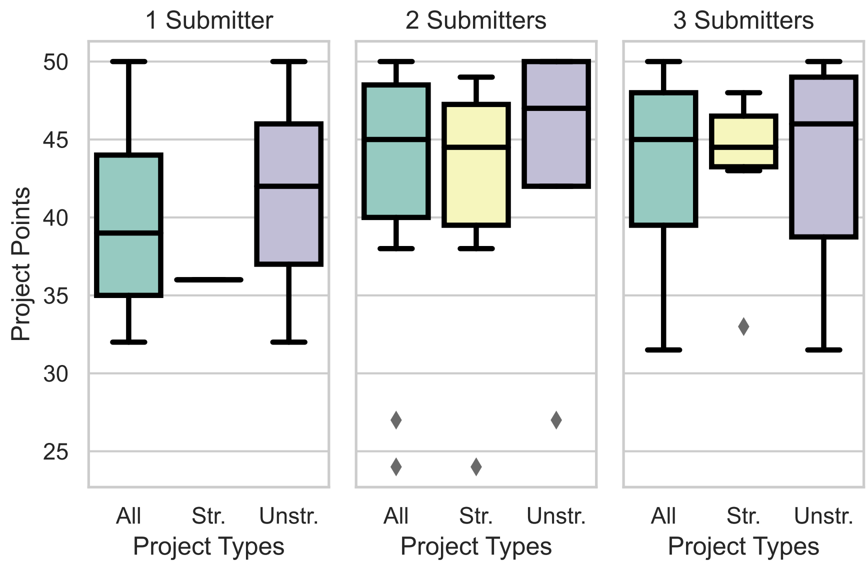


Figure 3.21: Teams Project Scores

requirement. This is a positive sign that our extra instructions for collaboration are fruitful. However, we should approach this result cautiously, as teams are not compelled to utilize the integrated chat for communication, potentially concealing much of their actual communication from us and exhibiting a distinct pattern.

3.2.12 Analysis of Project Results

At the end of the collaborative project, we graded the submitted tasks of the teams. The maximum amount of possible points is 50, with some teams having achieved this, as visible in Fig. 3.21. The score distribution leads to two insights. First, comparing the median scores of all groups, teams with more submitters also got better scores. Hence, collaboration really helped students reach better results on database-related tasks. However, our second insight reveals that teams completing tasks with a reflection requirement did not score as high as teams completing tasks without a reflection requirement, which contradicts our initial goal. However, this could be explained by the results from Fig. 3.19, where teams working on tasks with no reflection requirement had more Moodle points. This suggests that they possess superior skills and motivation for the course, which will consequently lead to improved outcomes in the collaborative project.

3.3 Chapter Summary

Collaboration across different disciplines is increasingly important for advancing scientific knowledge and translating research into practical applications. Highly integrated and engaged collaborative research teams exhibit several characteristics that enable them to successfully create and sustain their work over time. We evaluated teams that excelled in their tasks, as well as those that struggled due to conflicts, to identify crucial criteria for team success and effectiveness. The scientific objective is undeniably the focal point of the collaborative endeavor. We must establish supporting features to prevent the team from derailing. One of the most crucial aspects is trust. Without trust, the team's cohesion may deteriorate over time. Establishing a common vision, strategically selecting team members, fostering constructive disagreements, managing conflicts, and defining work roles such as stakeholder, administrator, and developer are key factors that team members should consider. In our case, this may involve a potential rotation of tasks. Self-awareness and proficient communication abilities are essential for the successful leadership and administration of scientific teams. Successful teams effectively do many activities, but there is no universal recipe for execution as each team has unique strengths and weaknesses. Individuals with excellent collaboration skills and awareness of the essential components needed to advance the core scientific work form effective scientific collaborations.

In the area of the development of a recommender system that helps students find suitable study partners for group assignments or projects. Students usually face challenges finding study partners, collaborating with randomly chosen partners, or both. Hence, we have considered developing a recommender system that takes students' personality traits through underlying personality questionnaires and suggests study partners. We have devised two approaches for grouping similar students based on their questionnaire responses. Both of these recommendations fall under the collaborative filtering method, which groups students based on personality similarities in their answers. In the first approach, we evaluate various algorithms designed to detect communities based on their modularity scores and their preference for forming teams of a specific size. The results confirmed that Leiden is the appropriate community detection algorithm for grouping students. As previously mentioned, we applied preprocessing techniques to our data during the implementation process, which included transforming raw questionnaire responses into scores, scaling these scores, and pruning the weighted graphs. These steps were necessary to improve the performance of the methods. Particularly, pruning the graph using the Maximum Likelihood Filter significantly enhanced the performance of the Leiden algorithm. We evaluated our results by comparing the modularity scores for the detected communities or groups, and observed better modularity values after pruning. Likewise, graph pruning showed a significant difference in the results for the clustering coefficients approach. In terms of reliability, the Leiden algorithm uses the multiplex partition method to form groups of desired sizes based on the responses from all six personality questionnaires. Unlike the other algorithms, the Leiden algorithm enables us to recommend study partners with several matching personality traits.

4. Automatic Instructional Feedback

This chapter draws upon the research effort from the following papers: Obionwu et al. [2022b, 2023a,b,d, 2022g, 2023g] and current efforts.

We have proposed and developed several intervention strategies and systems to facilitate the understanding of the SQL language and its skill acquisition. However, the majority of these interventions are primarily designed around a series of assessments, often neglecting the engagement patterns of the participants. Consequently, a student's exercise task engagements lead to a significant number of easily preventable errors. However, as evident in the body of literature, structured learning engagements can potentially increase an individual's awareness of the medium of instruction and the received instructions. Thus, in this chapter, we describe the challenges and potential benefits of incorporating structured learning engagements into SQL skill acquisition interventions. We will outline our efforts to understand students' engagement with the SQL language study area and how we plan to foster the acquisition of structured learning engagements. Additionally, we will examine the impact of these interventions on reducing avoidable errors and improving overall skill acquisition outcomes. The rest of the chapter is organized into three sections. Section 4.1 describes and evaluates students learning engagement. Section 4.2 describes keyword detection-based intervention strategies, and Section 4.3 describes intervention strategies that are based on retrieval-augmented generation.

4.1 Retrospective Student Engagement Analysis

In today's world, databases support the majority of electronic interactions. A database is an organized collection of data that consists of tables, queries, views, reports, and other objects. To be able to access and manage a relational database, we mostly use the *Structured Query Language (SQL)* which is the de facto standard for accessing and manipulating data [Saake, 2018]. SQL encompasses a variety of statement categories, loosely classified as sublanguages: data query language (DQL), data definition language (DDL), data control language (DCL), and data manipulation language (DML). [Saake, 2018].

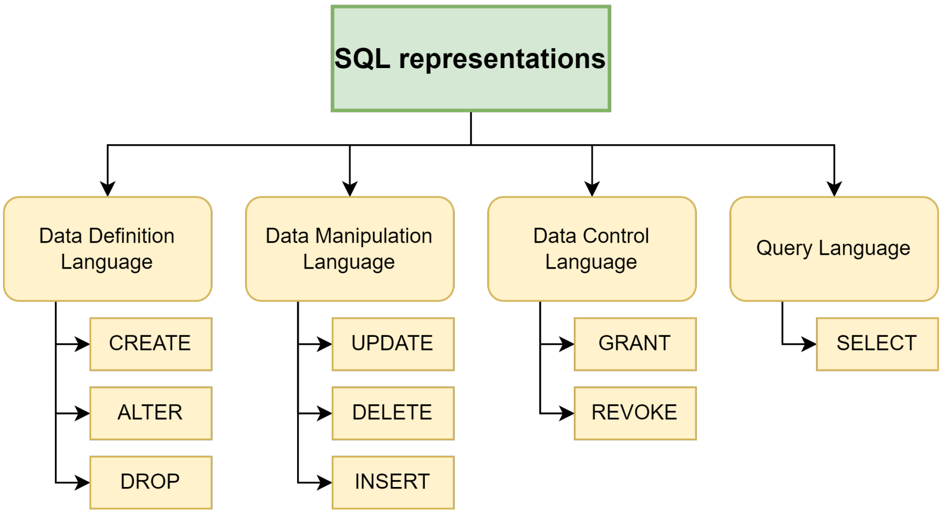


Figure 4.1: SQL language representations with a selection of important commands

The **Data Definition Language** (DDL) commands are essential for database administrators, as they allow them to define the organization and structure of the database. These commands include creating tables, defining constraints, and modifying the schema of existing objects. By using DDL, administrators can ensure data integrity and maintain a well-structured database system. DDL commands include statements such as **CREATE**, **ALTER**, and **DROP**, which allow users to create new tables, modify existing ones, or delete them entirely. These commands are essential for managing the overall structure and organization of a database, ensuring that it aligns with the desired schema and meets the specific requirements of an application or system. An example is shown in Listing 4.1. The **D**ata **M**anipulation **L**anguage (DML) includes commands such as **INSERT**, **UPDATE**, and **DELETE**, which are used to manipulate data stored in a database. These commands allow users to add new records, modify existing ones, or remove unwanted data from the database. DML statements are essential for maintaining the accuracy and integrity of the data within a database system. They provide a means for users to interact with the data and perform necessary operations according to their requirements. An example is shown in Listing 4.1. The **D**ata **C**ontrol **L**anguage (DCL) is a distinct category inside SQL Commands, specifically focused on managing and controlling data access and permissions. Some examples of DCL instructions are **GRANT**, which is a command that authorizes designated users to carry out specific actions, and **REVOKE**, a command used to eliminate a user's ability to access a database object. An example is shown in Listing 4.1.The **Data Query Language** (DQL) statements are utilized to execute queries on the data contained within schema objects. DQL commands serve the goal of retrieving the schema relationship depending on the provided query. While commonly associated with DML, it is more accurate to classify the SQL SELECT statement as an instance of DQL. The Data Manipulation Language classifies the SELECT statement when it incorporates data manipulators like FROM or WHERE. An example is shown in Listing 4.1. Having introduced the

Listing 4.1: Example DDL,DML,DQL,DCL Query

```
--DDL
CREATE TABLE Staff
(
   Staff_Position int primary key,
   FirstName varchar(25) NOT NULL,
   LastName varchar(25) NOT NULL,
   Department varchar(10) NOT NULL
   Academic_Degree varchar(15) NOT NULL
);

--DML
INSERT INTO Staff
   values('CEO','Victor','Christ','AGI-Dept.','Professor');

--DQL
SELECT * FROM Staff where LastName = 'Christ';

--DCL
GRANT ALL ON Staff TO 'Victor'@'localhost';
```

Now given an exercise task for the creation of the table MADE_OF, a student may attempt the task by submitting a query (cf. Listing 1 in the SQLValidator Obionwu et al. [2021a]. The system will validate the student's submission and provide feedback, depending on whether the query has some form of error or is correct.

Listing 1 Sample Student Query with Error

```
CREATE TABLE MADE_OF (
amount decimal,
wname varchar(20),
gname varchar(20),
foreign key(wname)references WINE(name),
foreign key(gname)references GRAPE(name),
primary key(wname))
```

To this end, the student will receive two tables as feedback, as shown in Table 4.1 and Table 4.2. In this scenario, the differences between the two tables indicate that the submission was incorrect. We will return the yellow color to the student as feedback if the primary key appears to contain an error or is missing. For this submission, *"8"* will be recorded in the error code attribute in the logs for the submission. Fig. 4.2 depicts their hierarchy, and Table 4.3 describes the error classes. In general, the codes designate a missing table feature, a syntax error, or a successful execution. There are currently five classes: syntax, table, foreign keys, constraints, and schema. The syntax error dominates every other error. For instance, if a primary error (such as a missing size description for the decimal type) coexists with a syntax error, as

Table 4.1: User Generated Solution Table

Schema				Constraints	
column_name	data_type	is_nullable	column_default	column_name	constraint_type
wname	varchar	NO	”	wname	PRIMARY KEY
gname	varchar	NO	”	gname	PRIMARY KEY
amount	decimal	YES	NULL	wname	FOREIGN KEY

Table 4.2: System Generated Solution Table

Schema				Constraints	
column_name	data_type	is_nullable	column_default	column_name	constraint_type
wname	varchar	NO	”	wname	PRIMARY KEY
gname	varchar	NO	”	gname	PRIMARY KEY
amount	decimal	YES	NULL	wname	FOREIGN KEY
				gname	FOREIGN KEY

previously mentioned, the display of the primary error will halt until the correction of the syntax error. Once the student corrects all errors, a color code indicates positive feedback. A syntax error shows a red color in the SQLValidator, while a semantic error is shown in a yellow color, and a green color indicates a successful submission.

A section of the logged data is shown in figure 4.4. We utilize these error classes to gain insights into students' learning engagement. The SQLValidator admin dashboard, our tool for analyzing student engagements and evaluating the viability of our intervention strategies, is shown in fig. 4.3. The home menu shows us an overview of the demographics, respective semesters, courses of study, and an overview of their interaction activities. It further enables us to analyze the surveys, evaluate group and individual skill acquisition, etc. We will describe our walk-through tutorial strategy in the next subsection, which guides novice students to become proficient in the use of structured query language.

4.1.1 Trials and Errors

It is important for students to be aware of syntax errors and also practice proper syntax in their query writing. From the evaluations we have done, we observed that syntax errors are the most common type of error that students make in their writing, as shown in fig. 4.5. To assist students in understanding the rules of syntax, we reviewed the literature for strategies that can effectively mitigate these errors.

We also noticed that the students lacked comprehension of system feedback and its ability to detect and rectify syntax errors in their queries. To address this issue, we designed an interactive exercise tutorial walk-through that guides students through the process of solving representative tasks. This tutorial walk-through, shown in Fig. 4.6, simulates a syntax error and representative schema-based errors. Once a student has successfully answered five correct query-based questions, we will disable the tutorial walk-through for their profile. By limiting access to the tutorial after a certain number of correct responses, we encourage students to independently apply their knowledge and skills to query writing. Should this criteria remain unmet, the system will display an overview of the available sample tasks. Once a task is selected, the user is prompted to read through the associated text. For this walk-through, task 1 is described. The Task 1 tutorial is intended to show the user the SQLValidator's behavior in the event of a syntax error, which is why it is imperative that the predefined query be used. Thus, in the fourth step, the user's attention is then drawn to the input text area where a predefined SQL query is to be inputted. To make certain the expected query is entered, an optional autofill function that automatically inserts the expected query into the input field is provided. Given that the intended query is entered, the corresponding feedback from SQLValidator in the event of a syntax error is displayed in step eight. If not, the system prompts the user to input the anticipated query. Steps nine and ten return to the user a well-described explanation of the feedback. The next activity corrects the error-prone SQL query and repeats the submission step, leading to the successful completion of the task. The activity will conclude, and an offer will be made for the user to proceed to the next tutorial walk-through. These second and third walkthroughs simulate different forms of scheme-related errors.

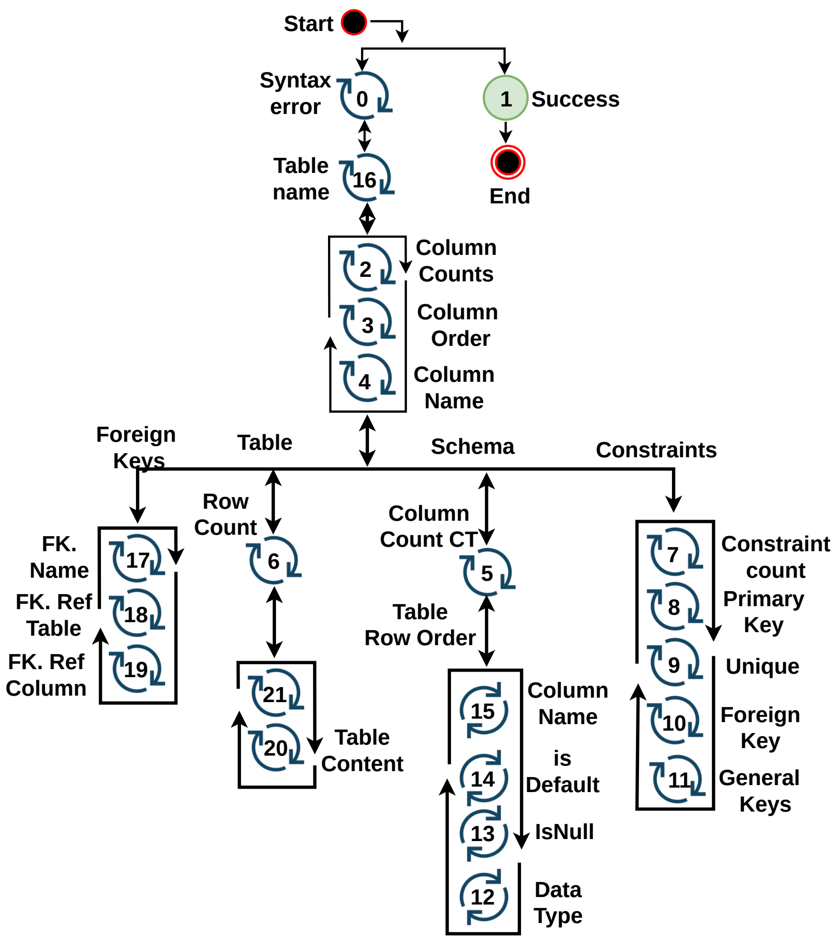


Figure 4.2: Error Classes and Codes

Table 4.3: Error Codes Table

Error Class	**Error Code**	**Error Name**	**Description**
Syntax Error	0	Syntax Error	Missing semi-column at the end of the statment
Table	2	Column Counts	Missing to intitialize one or more columns
	3	Column Order	Starting a table with "Name" column and then the "ID" column
	4	Column Name	Name a column as "Name" instead of "First Name"
	6	Row Count	Giving a higher number of rows than the number in the table
	20	Table Content	There is a "plz" column in the student's courses table
	21	Table RowOrder	The Rows show up not in the order of "ID" column
Constraint	7	Constraint Count	Missing one or more constraints
	8	Primary Key	Missing to intitialize PK
	9	Unique	Missing to intitialize unique column
	10	Foreign Key	Missing to initialize FK
	11	General Keys	Missing to intitialize unique key
Foreign Keys	17	FK Name	Writing the FK-Name wrongly
	18	FKRef Table	Writing the FK-Name of reference table wrongly
	19	FKRef Column	Writing the FK-Name of reference column wrongly
Schema	5	Column Count CT	Entering 4 values to a table that contains 3 columns
	12	Data Type	Intialize "name" column with integer value
	13	IsNull	Missing to intialize that a value can be null
	14	IS Default	Missing to intialize that a value is deafult value for a column
	15	Column Name	Intialize "name" column instead of "firstname"
	16	Table Name	Name a student table instead of Employee

Figures 4.7, 4.8, and 4.9 show a semester-wide snapshot of the count of syntax errors from the summer semester of 2020, where students engaged mostly using the trial and error method, the summer semester of 2021, when we integrated the tutorial system, and the current 2022 summer semester. The visualization shows a progressive decrease in the count of syntax errors committed by the student during their online exercise engagement. In total, 120 students engaged with the SQLValidator during the summer semester 2020 database concepts course and accumulated a total of 10,863 syntax errors. The integration of the tutorial system into the summer 2021 database concepts course resulted in the accumulation of 4537 syntax errors among the 94 students. The availability of the tutorial walk-through and other latent factors led to the adoption of a structured form of engagement. In the just-concluding 2022 summer semester, 69 students engaged with the SQLValidator and accumulated a total of 1,361 syntax errors. While we have implemented collaborative learning and a recommendation systemObionwu. et al. [2022] by now, which has in no doubt

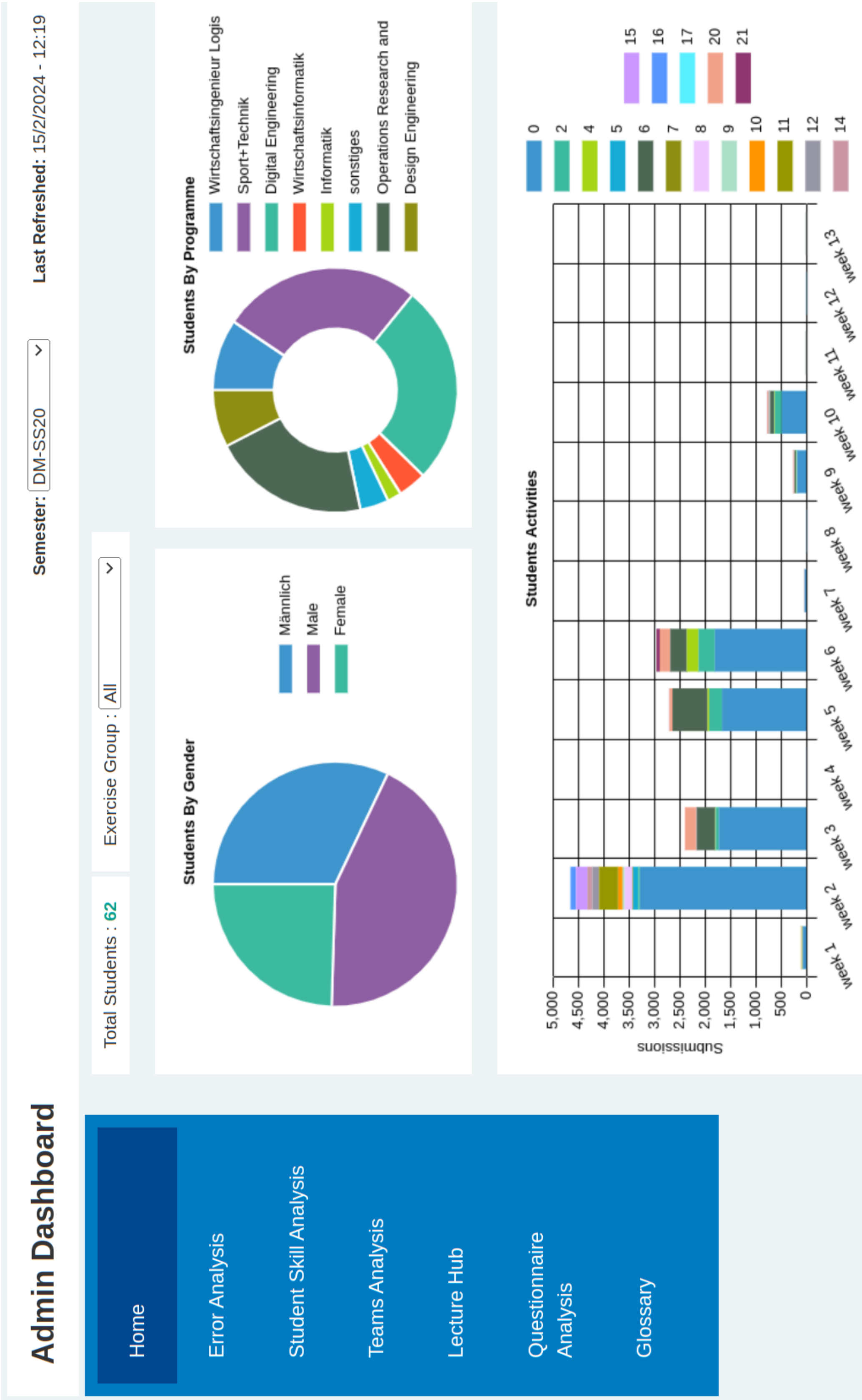


Figure 4.3: Admin Dashboard

ID	User ID	User Query	Error Classes	Task id	Time Stamp
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, customer.cid from customer inner join on orders.cid=17	0	787	2021-06-02 21:47:39
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, customer.cid from customer inner join orders on orders.cid=17	6	787	2021-06-02 21:47:54
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, customer.cid from customer inner join orders on customer.cid=17	6	787	2021-06-02 21:48:17
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, orders.cid from customer inner join orders on customer.cid=17	6	787	2021-06-02 21:50:10
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, orders.cid from customer inner join orders on orders.cid=17	6	787	2021-06-02 21:50:27
xxxx	xxxx	select orders.oid, orders.Did, orders.Date, orders.cid from orders inner join customer on orders.cid=17	6	787	2021-06-02 21:50:45
xxxx	xxxx	select * from orders join customer on customer.cid=17	2	787	2021-06-02 21:52:32
xxxx	xxxx	select * from orders left join customer on customer.cid=17	2	787	2021-06-02 21:56:01
xxxx	xxxx	select * from orders leftjoin customer on customer.cid=17	0	787	2021-06-02 21:56:25
xxxx	xxxx	select * from orders left join customer on customer.cid=17	2	787	2021-06-02 21:56:29
xxxx	xxxx	select * from orders Left join customer on customer.cid=17	2	787	2021-06-02 21:56:35
xxxx	xxxx	select * from Orders where customer.cid=17	0	787	2021-06-02 21:57:14
xxxx	xxxx	select Orders.ord, orders.did, orders.date, customer.cid from Orders where customer.cid=17	0	787	2021-06-02 21:58:05
xxxx	xxxx	select Orders.oid, orders.did, orders.date, customer.cid from Orders where customer.cid=17	0	787	2021-06-02 21:58:13
xxxx	xxxx	select * from Orders where customer.cid=17	0	787	2021-06-02 21:58:30
xxxx	xxxx	select * from Orders where order.cid=17	0	787	2021-06-02 21:58:54
xxxx	xxxx	select * from Orders where orders.cid=17	0	787	2021-06-02 21:59:04
xxxx	xxxx	select Orders.oid, orders.did, orders.date, customer.cid from customer where customer.cid=17	0	787	2021-06-02 21:59:58
xxxx	xxxx	select orders.oid, orders.did, orders.date, customer.cid from customer where customer.cid=17	0	787	2021-06-02 22:00:08
xxxx	xxxx	select orders.oid, orders.did, orders.date, customer.cid from customer, orders where customer.cid=17	6	787	2021-06-02 22:00:27
xxxx	xxxx	select orders.oid, orders.did, orders.date, customer.cid from customer, orders where. orders.cid=17	6	787	2021-06-02 22:08:22
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from customer, orders where. orders.cid=17	6	787	2021-06-02 22:09:25
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from customer, orders where orders.cid=17	6	787	2021-06-02 22:10:53
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from customer, orders where orders.cid=customer.cid	6	787	2021-06-02 22:11:31
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid, customer.name from customer, orders where orders.cid=customer.cid	2	787	2021-06-02 22:11:51
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid, customer.name from customer, orders where orders.cid=customer.cid and orders.cid=17	2	787	2021-06-02 22:12:12
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from customer, orders where orders.cid=customer.cid and orders.cid=17	0	787	2021-06-02 22:12:26
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from orders where orders.cid=customer.cid and orders.cid=17	0	787	2021-06-02 22:12:37
xxxx	xxxx	select orders.oid, orders.did, orders.date, orders.cid from customer, orders where orders.cid=customer.cid and orders.cid=17	0	787	2021-06-02 22:12:49
xxxx	xxxx	Select * from Orders INNER JOIN Customer ON Customer.cid=Orders.Cid;	2	787	2021-06-03 15:09:38
xxxx	xxxx	Select * from Orders INNER JOIN Customer ON Customer.cid=Orders.Cid and customer.cid=17;	2	787	2021-06-03 15:10:59
xxxx	xxxx	Select * from Orders INNER JOIN Customer ON Customer.cid=Orders.Cid and customer.Cid=17;	2	787	2021-06-03 15:11:11
xxxx	xxxx	Select * from Orders INNER JOIN Customer ON Customer.cid=Orders.Cid and Customer.Cid=17;	2	787	2021-06-03 15:11:27

Figure 4.4: Log Data

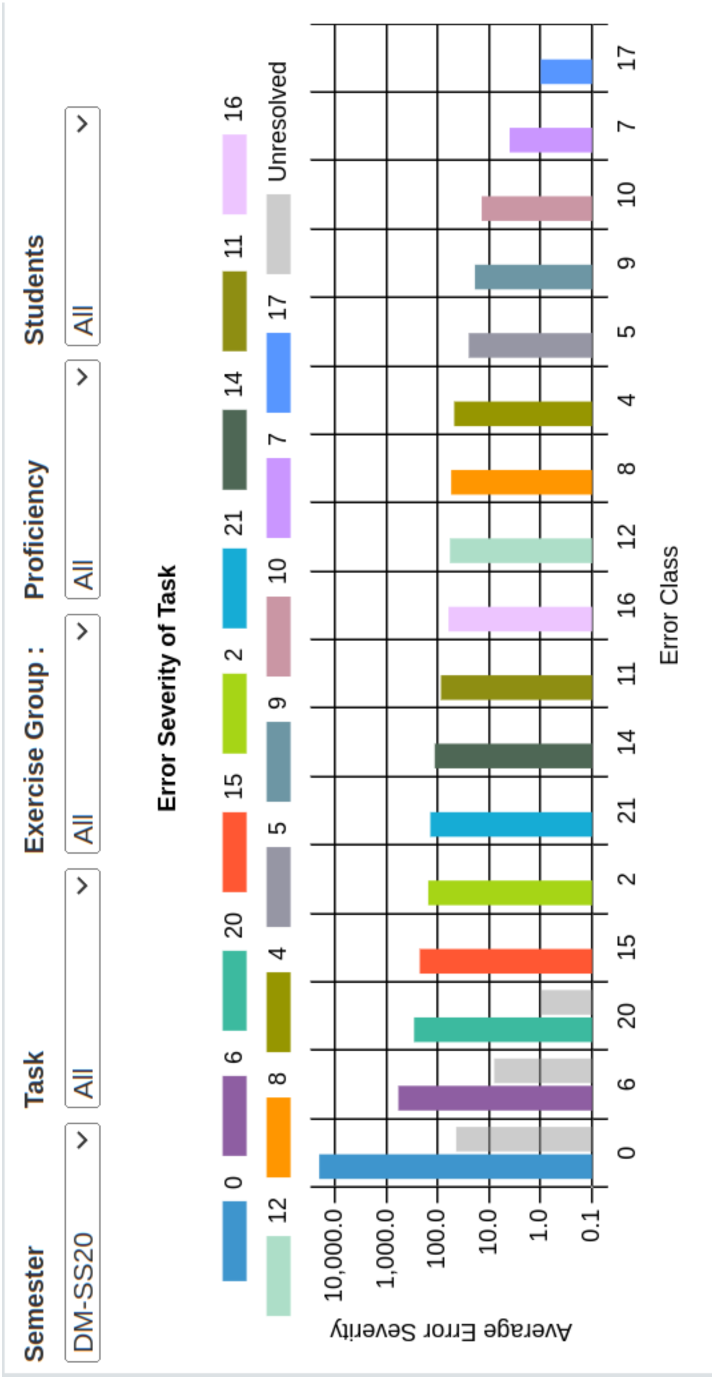


Figure 4.5: Average Error Severity

contributed to the low count of syntax errors, the viability of the SQLValidator tutorial walk-through is evident, as recorded by similar tutorial walk-through implementations in the literature Alomari et al. [2020]Lin et al. [2021a]da Silva and da Silva Aranha [2015]. While this strategy is effective for addressing syntax errors, schema-based errors are indicators of a lack of knowledge. Therefore, the next strategy, the slide recommendation strategy, aims to guide the students to specific sections of the slide that discuss the relevant exercise tasks. We discuss this strategy in the next section.

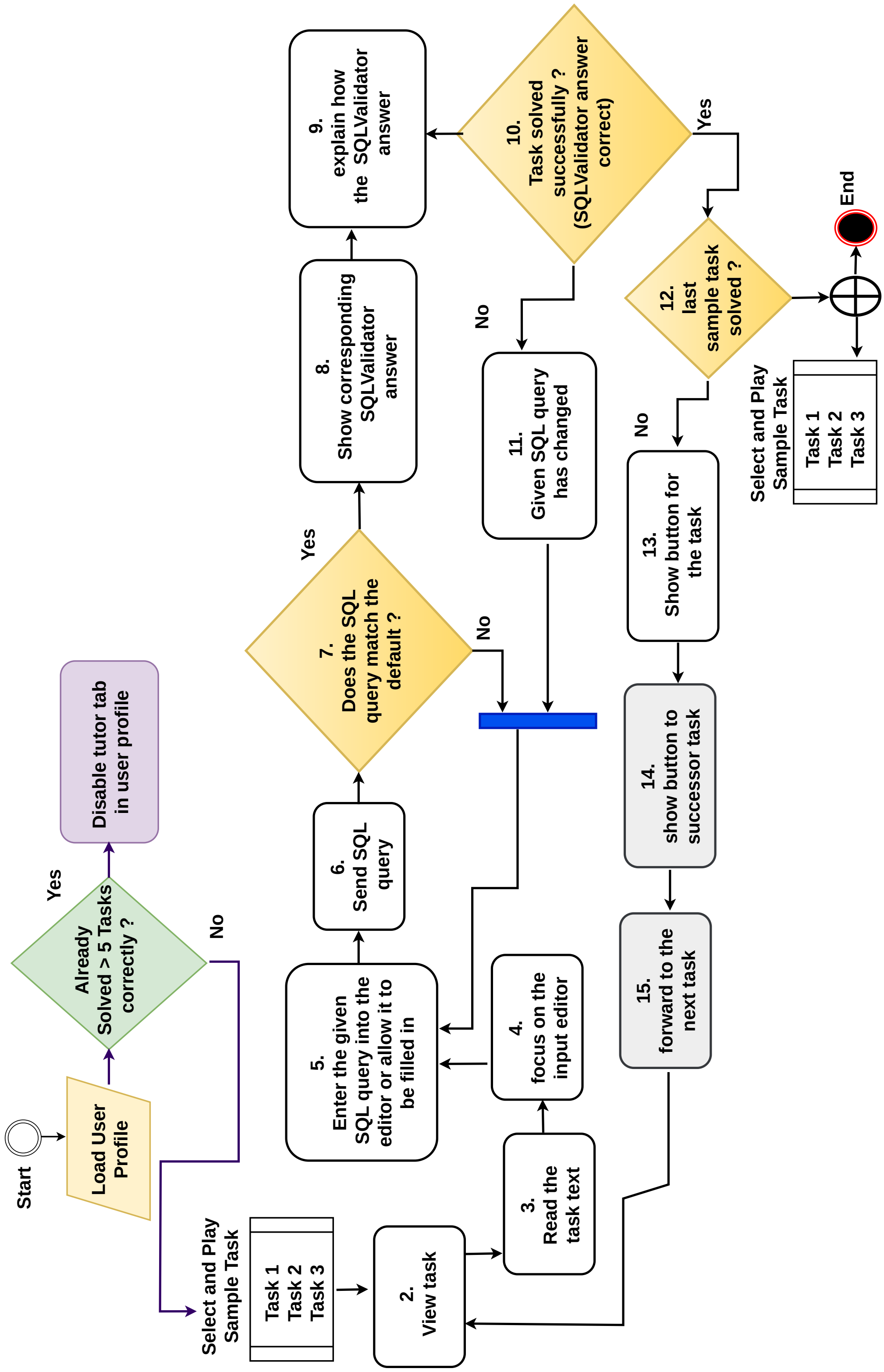


Figure 4.6: *SQLValidator Tutorial Overview*

4.2 Slide recommendation System

Errors will always occur during programming exercises, and Fig. 4.2 shows the different types of errors that students are likely to encounter while working on

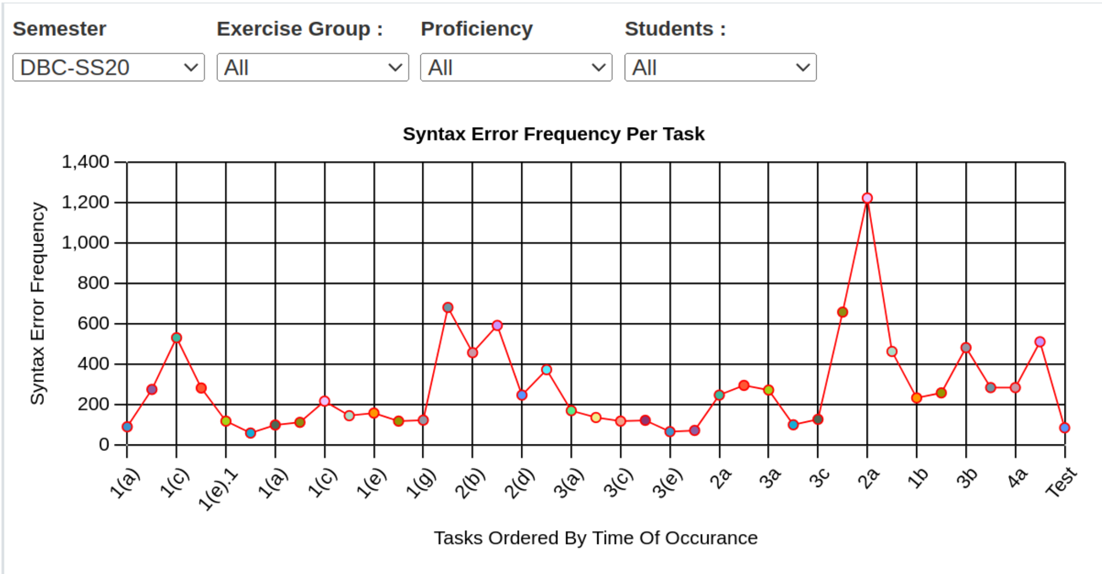


Figure 4.7: *2020 Syntax Error Distribution*

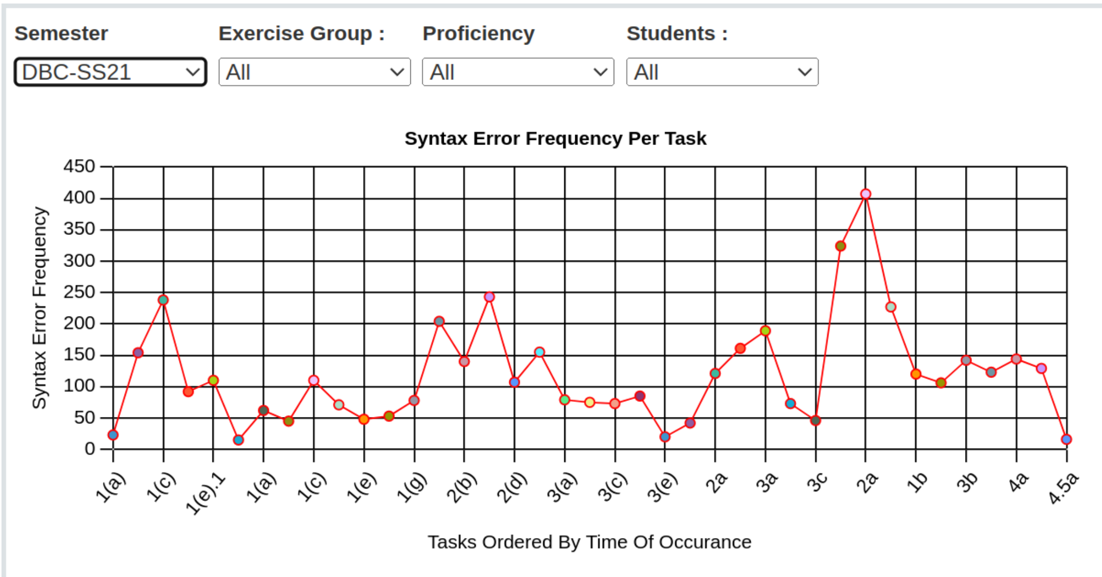


Figure 4.8: *2021 Syntax Error Distribution*

structured query language exercise tasks. A description of the errors is given in table 4.3. It is estimated that between half and 90% of university students are prone to syntax and other forms of errors Ahmed et al. [2022]; Geng et al. [2023]; Saenz and De Russis [2022]. To address this challenge, we devised a recommendation strategy that leveraged the relationship between lecture slides and respective exercises to create instructional feedback. Instructional feedback involves providing students with information about their learning in order to assist them in their study activities Smith and Lipnevich [2018]. Students typically receive feedback in various forms, including written comments and verbal discussions. It can come through traditional avenues or digitally and is essential for promoting growth and improvement in both academic and professional settings. In the next subsection, we describe the traditional forms of instructional feedback.

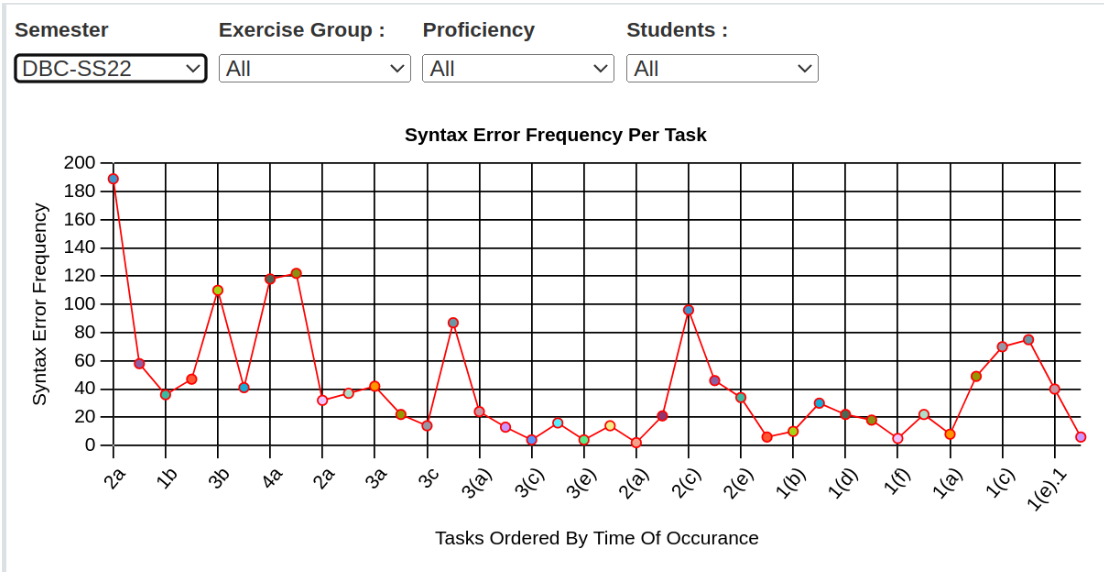


Figure 4.9: *2022 Syntax Error Distribution*

4.2.1 Traditional Forms of Instructional Feedback

Teachers have historically used established techniques known as conventional instructional feedback methods to assess, direct, and assist students' learning. These strategies typically involve direct interaction between teachers and students and has been an integral part of educational practices for a long time. These strategies include:

1. Verbal Feedback: This entails direct interaction between the teacher and student, typically occurring during class discussions, one-on-one interactions, or teacher-student conferences. Verbal feedback allows for immediate clarification and personalized guidance, fostering a deeper understanding of the material. Additionally, it provides opportunities for students to ask questions and engage in dialogue with their instructors. Kluger and DeNisi [1996]

   - In-Class Discussions: Teachers interact with students by discussing their work, providing prompt verbal comments to help explain ideas, correct misunderstandings, and promote analytical thinking. This real-time feedback can enhance student comprehension and encourage active participation in the learning process. Additionally, it allows for a more dynamic exchange of ideas and fosters a collaborative learning environment.

     Topping [1998]
   - One-on-One Conferences: One-on-one meetings between teachers and students can provide personalized feedback tailored to the individual student's needs and learning style. These meetings allow for a deeper understanding of the student's progress and challenges, fostering a stronger teacher-student relationship. Van der Kleij et al. [2015]

2. Written Comments: Instructors offer written feedback on assignments, essays, tests, or projects. This feedback is valuable for students to understand their

Summary of papers for instructional feedback		
Author	Type of Feedback	Result
Chalmers et al. [2018]	face-face Vs. written feedback	• face-face feedback was beneficial and positive in keeping up student motivation • pedagogic value - face-to-face feedback gained greater appreciation for the rigorous nature of the markers' decision-making • feedback as dialogue helped build better student-teacher rapport
Cordova et al. [2021]	traditional vs. conceptual feedback	• students who received conceptual feedback had a higher level of effectiveness in applying their new knowledge. • students who received conceptual feedback had better code coverage and higher programming grades
Denton et al. [2008]	traditional vs. computer-assisted formative feedback	• 40 students out of 169 found electronic feedback more valuable compared to the handwritten comments • digital feedback was easier to read than handwritten feedback
Johnson et al. [1999]	online learning Vs. face-face learning	• online learning lacked strong social interactions • face-face learning showed better teacher-student interaction • face-face learning allowed detailed analysis of the class and helped the teacher vary the type of feedback • self assessment was more *"comfortable"* in online learning
Kluger and DeNisi [1996]	feedback intervention	• effectiveness of feedback intervention decreases as the focus changes from the task to the self.

Table 4.4: Summary of papers for instructional feedback

strengths and areas for improvement. It also helps instructors communicate specific points and suggestions for future work. The remarks can be thorough and precise, emphasizing strengths, identifying areas for work, and proposing tactics for enhancement. Hattie and Timperley [2007]

3. Quantitative Feedback: Assigning grades or marks on assignments, tests, or assessments is a form of feedback. This type of feedback provides students with a clear indication of their performance and progress in a course. It can also motivate students to strive for improvement and success in their academic endeavors. Though often considered summative, grades can also serve as feedback if accompanied by explanations or comments detailing the reasons behind the grade. Black and Wiliam [2010]

4. Rubrics: An explicit set of criteria and expectations used for assessing a particular type of work or performance. Rubrics offer transparency by breaking down what constitutes exemplary work into different categories and levels. This enables students to comprehend the precise expectations and the evaluation process. Rubrics can also help instructors provide consistent and fair feedback to all students based on predetermined criteria.

   Andrade [2005]

Table 4.4 displays a summary of research publications investigating several traditional feedback methods in education and their use in diverse educational settings. The research articles in the table offer theoretical explanations and insights on the effectiveness and contextual significance of traditional feedback compared to other educational feedback forms. In the next section, we describe the digital and hybrid forms of instructional feedback.

4.2.2 Digital Instructional Feedback

Digital instructional feedback involves using technology-based methods to provide guidance, evaluation, and support in educational settings. This feedback technique uses digital tools and platforms to enhance the teaching and learning process, offering a dynamic and often personalized way for teachers and students to communicate and collaborate. It allows for immediate feedback, promotes student engagement, and can help track progress more efficiently compared to traditional methods. Digital instructional feedback includes many formats, such as online quizzes, interactive exercises, and virtual simulations Yarbro et al. [2016]. Educators can use these tools to provide immediate feedback and track students' advancements in real time, allowing for targeted interventions and personalized education. Furthermore, digital instructional feedback promotes active engagement and introspection among learners, fostering autonomy and self-guided learning Grant and Basye [2014]. Digital instructional feedback can enhance the accessibility and inclusivity of education by offering accommodations for learners with diverse needs, such as visual impairments or learning disabilities. Additionally, the use of digital instructional feedback can help bridge the gap between traditional classroom settings and remote learning environments, providing flexibility for students to access educational resources from anywhere Grant and Basye [2014]. This scenario is referred to as hybrid pedagogy. By combining face-to-face instruction with online resources and tools, hybrid pedagogy allows for flexibility and personalized learning experiences.

4.2.3 Hybrid instructional recommendation

Hybrid instructional feedback combines conventional face-to-face teaching with online or digital components. This technique tries to leverage the advantages of both in-person and digital interactions to create a flexible and dynamic learning environment. Johnson et al. [1999]. Researchers have found that hybrid instructional feedback enhances student engagement and motivation by providing a more personalized learning experience. This approach also allows educators to provide timely feedback and support to students, leading to better academic outcomes overall. By incorporating digital components, learners may easily access resources and engage in interactive activities at their desired pace, thus enhancing their understanding and retention of knowledge. Singh et al. [2022]. Hybrid instructional feedback promotes collaboration and communication among students through the use of online discussion forums and group projects. This allows for a variety of perspectives and encourages active participation, fostering a deeper understanding and analytical thinking skills. Engaging in introspection and self-improvement enhances students' academic performance and equips them with useful skills applicable in other aspects of their lives Alarifi [2023]. Moreover, the integration of personalized feedback and a growth-oriented mentality creates a positive learning environment that motivates students to strive continuously for outstanding academic achievement. In the next section, we describe some of the concepts we employed to derive the instructional feedback we provided to students.

4.2.4 SQL exercise analysis

Knowing which SQL topics feature in the exercises is necessary to support students in solving them. Our strategy for topic extraction is similar to the topic extraction for the lecture slides. Essentially, we analyze the appearances of SQL keywords in the exercise solutions, and then, by computing tf, idf, and tf*idf values, we estimate which keywords best describe the exercise. We collated a list of relevant keywords from a pool of SQL keywords containing 58 elements. The chosen list of keywords is displayed in Table 4.5

Relevant SQL keywords					
select	distinct	where	and	or	not
null	update	delete	min	max	count
avg	sum	like	in	between	as
join	union	group	having	exists	any
all	case	create	<	<=	>
>=	round	=	drop	alter	constraint
unique	primary	foreign	check	default	view
concat	substring	select distinct	natural join	left join	right join
full join	primary key	foreign key	create view	create table	group by
order by	insert into	insert	order		

Table 4.5: Selected SQL keywords for the keyword list

We note here that our exercise task administration environment has an inbuilt set of solutions for each exercise task. The SQL exercise analysis workflow is shown in Fig. 4.10. The respective activities are briefly described in the following sections.

4.2.4.1 Querying stored solutions

A database stores the solution to each SQL exercise, along with other exercise-related information like an ID and locale preferences. We obtain the exercise solutions and their IDs by querying the database, which is our only objective. We store the result as an associative array, where the key is the exercise ID and the index value contains the solution in string format. Unlike the lecture slides, preprocessing is not necessary because the solutions come already formatted. Next, we recognize the keyword. We perform this operation similarly to how we handle the lecture slides.

The keyword recognition activity block in Fig. 4.10 shows a sample list of recognized keywords.

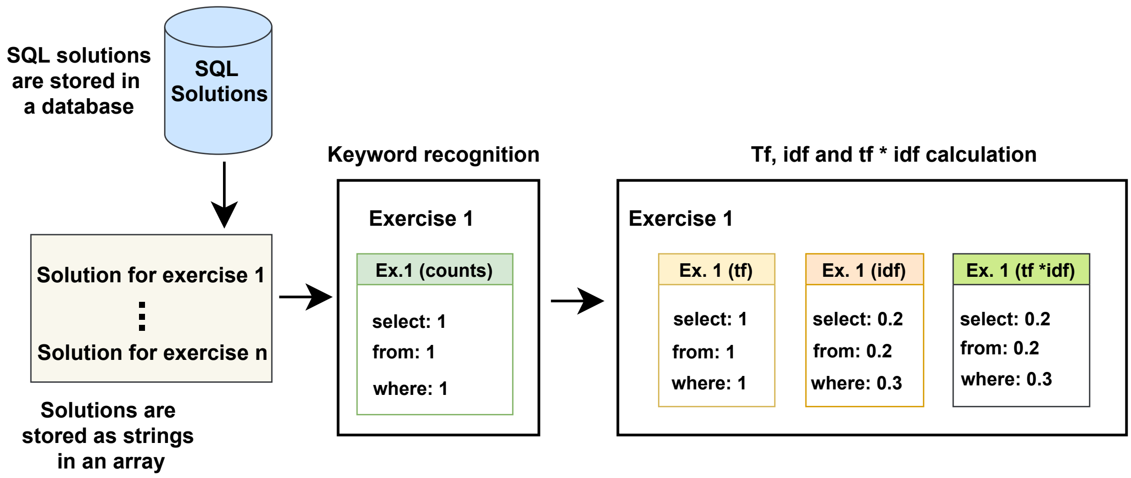


Figure 4.10: Concept of SQL exercise analysis

4.2.4.2 Keyword analysis

In the next step, we calculate the tf, idf, and tf*idf values for each keyword on a page. Considering the exercise displayed in Figure 4.10, the keyword that has the highest IDF weight is the WHERE keyword, with the select and FROM keywords having a lower IDF value. The tf*idf computation shows that the topic of the page is about the usage of the WHERE keyword.

4.2.5 Concept of comparing slides and exercises

In the previous sections 4.2.4, we described the process of converting both slides and SQL exercises into a format that allows us to compare them. The comparison is done by computing the cosine similarity between the lecture slides and SQL exercises Obionwu et al. [2022a]. A depiction of this process for hypothetical Exercise 1 is shown in Figure 4.11. It consists of the following steps: merging of previous analysis, computation of cosine similarity, and mapping of exercises to pages with the highest cosine similarity.

4.2.5.1 Merging of previous analysis

The results of the slides and exercise analysis are merged into a list during the first step of the comparison process. For each exercise, we now have access to the tf*idf values from every page with respect to this exercise. Figure 4.11 shows the mapping between Exercise 1, and each page alongside the result of their keyword analysis. The three keywords SELECT, WHERE, and FROM were identified in Exercise 1 with their tf*idf values of 0.2, 0.2, and 0.3. Page 1 contains the same keywords and tf*idf values as Exercise 1, plus the keyword " $>$ " with a tf*idf value of 0.3. The second page features the SELECT and FROM keywords with tf*idf values of 0.2 and 0.5. The last page contains the ALTER and TABLE keywords with values of 1 and 0.5.

4.2.5.2 Computation of cosine similarity

The cosine similarity calculates the angle between two word vectors. In our use case, the word vectors consist of keywords recognized from a query or lecture slide.

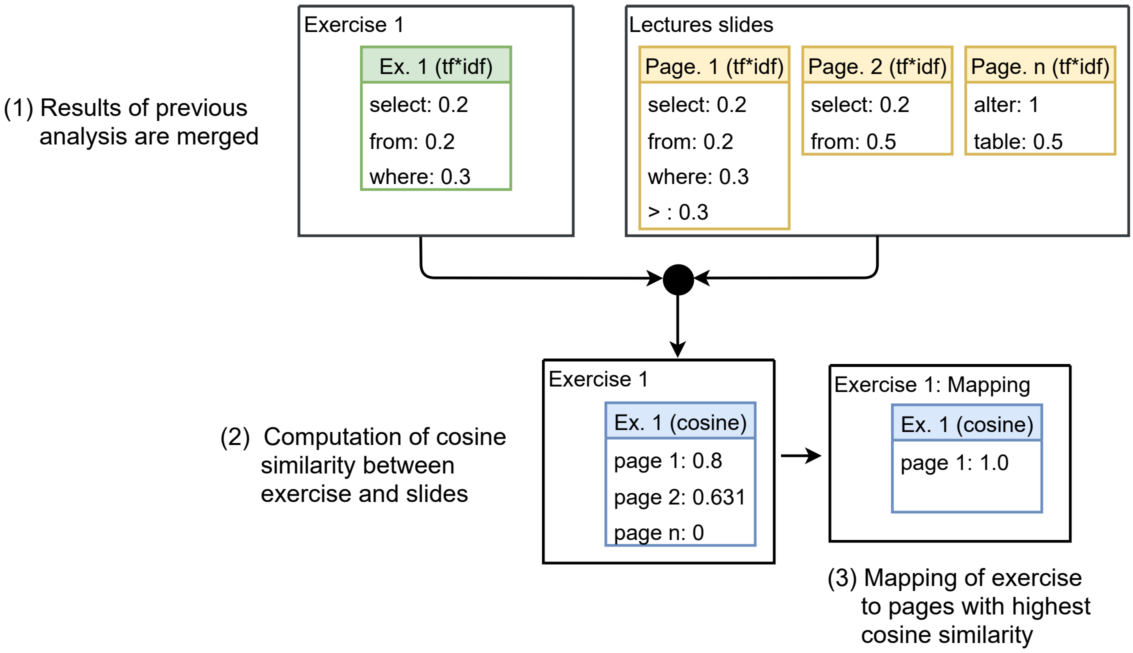


Figure 4.11: Concept of combining the analysis results for the lecture slides and SQL exercises together Obionwu et al. [2022a]

$$cosine(a,b) = \frac{a \cdot b}{\|a\| \cdot \|b\|} = \frac{\sum\limits_{n=1}^{N} a_i, b_i}{\sqrt{\sum\limits_{i=1}^{N} a_i^2}\sqrt{\sum\limits_{i=1}^{N} b_i^2}}$$

Figure 4.12: Formula for cosine similarity by Sidorov et al. [2014]

The calculation of the angle is shown in Figure 4.12. We divide the dot product of two vectors, a and b, by their respective lengths. A low angle between those vectors means that their content is similar, while a high angle expresses dissimilar content. An advantage of using the cosine angle as a similarity metric is that the length of the vectors is not relevant. In our use case, we employed this method.

As a result, the cosine similarity for each page of the lecture is calculated with regard to Exercise 1. The result shows that Page 1 has the highest cosine similarity of 0.8, followed by Page 2 with 0.631, and Page n with 0. The page with the best cosine value, in this case, Page 1, is then selected to be mapped to Exercise 1, and hence recommended to students having a problem with Exercise 1. A further feature of our system is the clustering of keywords, which we will discuss next.

4.2.5.3 Keyword Clustering

A cluster is a group of objects differentiated based on their similarity and dissimilarity Diday and Simon [1976]; Jain et al. [1999]. Thus, keywords are grouped such that members of a group are more similar to other keywords in the same group and dissimilar to the members in other groups. Our system has a feature that allows the creation of clusters consisting of an arbitrary number of keywords Habibi and Popescu-Belis [2015]. To the end, after tf*idf values for each keyword are calculated, members of the cluster are scanned for the highest tf*idf value Ramos et al. [2003]; Wu et al. [2008]. We apply this value to each member of the cluster. The reasoning for clustering keywords is the presupposition that the occurrence of specific keywords leads to a certain SQL topic. For example, clustering the keywords $<$ and $>$ employed as a range selection increases their tf*idf value compared to other keywords in the page that are not part of the cluster. As a result, the clustered keywords exert a greater influence on the cosine calculation. Thus, it is more likely that the resulting recommendation will point to a page about range selection. In the next section, we will describe the recommendation workflow.

4.2.6 Recommendation Workflow

The recommendation of a lecture slide necessitates the execution of several processes and decisions, and this is our primary strategy for integrating instructional feedback into the SQLValidator. In this subsection, we give a detailed description of how it works. The workflow is shown in figure 4.13. The description of each process uses an index number to describe the corresponding action.

The recommendation process in the frontend starts once the student submits a solution, and an error condition occurs. The system will generate a recommendation based on the specific task. The backend generates recommendations by analyzing the content of the exercises and slides and identifying the characteristics of SQL keywords. We will establish a mapping between the lectures and slides using the keywords. This mapping enriches the feedback students receive by recommending the slide section most similar to a specific exercise. To carry out this process, the administrator will first navigate to the slide recommendation tab, A1, and initiate the process, A2. Select one of three recommendation states: ground truth, prediction system, or no recommendation in A3. Selecting the option of no recommendation will prevent A4 from providing any recommendations to the student. This serves as a way of loosely coupling the recommendation system with the SQLValidator task management system, should there be a decision not to use the recommendation. We further utilize this option for A/B testing, ensuring that one group of students does not receive recommendations while another receives suggestions. Ultimately, we compare the performance of both groups to determine whether the recommendation was beneficial for the students.

The next option is to choose whether to use the ground truth or a predicted recommendation. Once we select the ground truth, we indicate the maximum number of recommended slides (A5). We store this number in the database, enabling future attempts at a question to query the database and determine the number of recommendations to send to students. In A6, the student will receive an optimal. recommendation chosen by an expert. The path to manual recommendation requires a lot of effort, as lots of time needs to be spent choosing which slide is suitable for which exercise, and in cases where the order of slides changes, the process will need to be repeated. A more attractive option will be to use the prediction system, A7. When choosing this option, we also specify the number of recommendations we will

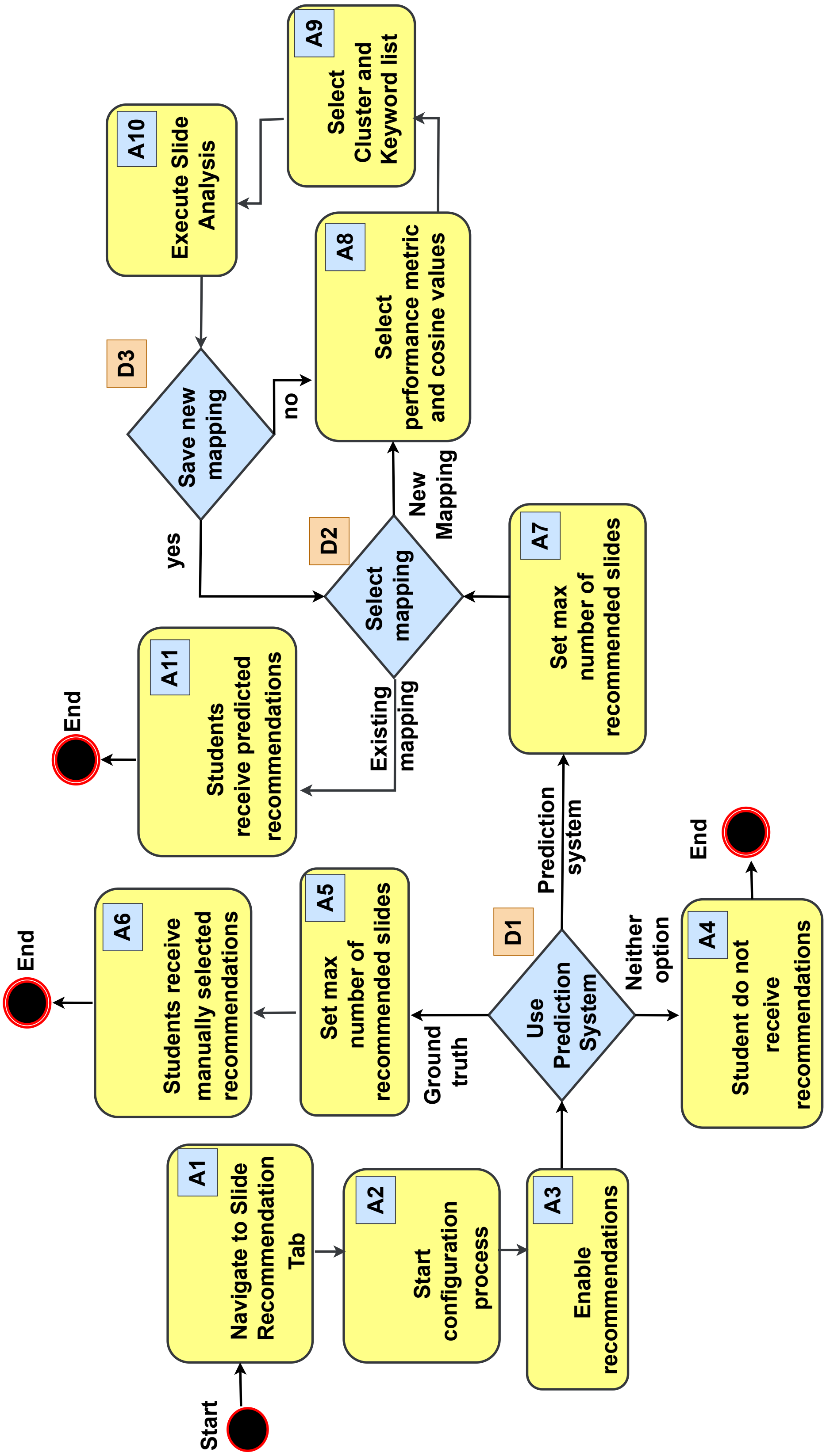


Figure 4.13: Recommendation Workflow

provide to the student. To ensure the best recommendation, we choose slides that have a high cosine similarity.

The next option involves selecting a map. Each exercise's recommendations, meta data such as performance values, and the specific semester of creation comprise a mapping. In A8, the administrator can choose to use an existing mapping or create a new one. To prevent incorrect recommendations, a new mapping requires the specification of a set of configurations, including the minimum cosine value, cosine cutoff, and join detection, among others. The minimum cosine value ensures that recommended slides achieve a minimum similarity of 0.5. A9 selects both the cluster and the keyword list. We create these lists from a pool of SQL keywords. This pool exceeds the current number of keywords currently used in the slide, so future slide modifications will not lead to the modification of the list. However, we must manually update the list if an unknown keyword appears. The next action is the execution of the slide analysis, A10. After computing the similarity between lecture slides and exercises, we generate a visualization for each recommendation. After checking the recommendations, the evaluation settings, which consist of the performance settings of accuracy, precision, recall, specificity, f-measure, and f-beta, are also checked. The administrator now has the option to store the mapping in the database if they choose to make it persistent. However, the database will not store the mapping if it fails to achieve appropriate performance. Once a mapping becomes active, it generates recommendations for each exercise, providing feedback on slide location. In A11, students who encounter errors during their exercise engagements will receive a recommendation that directs them to the lecture hub.

In the past sections, we have introduced the background and conceptual framework for our recommendation system. In the next section, we will show a sample process for mapping lecture slides to exercise tasks.

4.2.7 Implementation

Keywords are a vital part of our strategy, as they are used every time we search for occurrences of certain SQL keywords. We implemented a tool to create customary keyword lists that contain specific SQL keywords selected by the administrator. For instance, our implementation includes SQL keywords like CONCAT, which the lecture does not currently mention. Since this could also change in the future, our implementation already supports a wide variety of keywords. We now show the process of mapping slides to exercises by walking through a recommendation for Exercise E that is about the update operation:

Update wine set vintage = vintage +1
Where color = 'red';

The keyword analysis for Exercise E is displayed in Table 4.6 with WHERE and UPDATE recognized once and the equality sign twice. The idf values reveal that the UPDATE keyword is the rarest of these three, with an idf of 1.82 compared to 0.301 for the WHERE keyword and 0.519 for the equality sign. A tf*idf value of 0.91 is assigned to the UPDATE statement, which means that it is the most important SQL keyword for the Exercise E.

Our system identifies the 34th page of the second chapter as the most similar slide in the lecture. The keyword analysis for this slide is shown in Table 4.7 with the

keywords	count	tf	idf	tf-idf
Where	1	0.5	0.301	0.151
Update	1	0.5	1.82	0.91
=	2	1	0.519	0.519

Table 4.6: Keyword analysis for exercise E Obionwu et al. [2022a]

Chp: 2, Page: 34, Cosine: 0.984				
keywords	count	tf	idf	tf-idf
where	1	0.5	0.507	0.254
update	2	1	1.109	1.109
in	1	0.5	0.273	0.137
=	2	1	0.556	0.556
as	1	0.5	0.316	0.158

Table 4.7: Keyword analysis of the recommended page to exercise E Obionwu et al. [2022a]

keywords WHERE, IN, AS recognized once and UPDATE and equality sign twice. The cosine similarity between Exercise E and the recommended page with 0.984 is close to one, which resembles a high similarity.

4.2.8 Baseline evaluation

The baseline evaluation of our system derives a mapping between slides and SQL exercises by purely computing the cosine similarity without using any additional parameters. We only need to select the preferred method for calculating the IDf values. The baseline approach selects only the slide with the highest cosine similarity for recommendation. If multiple slides share the best cosine value, then they are recommended. The confusion matrix of the baseline approach using the idf$_{sub}$ computation is shown in Table 4.8. In our implementation, we used 180 slides and 66 SQL exercises, which led to 11.880 entries in the confusion matrix. Out of the 70 entries predicted as positive, 38 were positive. The column predicts 11.810 entries as negative, of which 178 are counted as false negatives and 11632 as actual negatives. The table cell of true negative entries is of interest to us since it contains a little more than 98% of all entries. This imbalance in instance distribution is expected because there are 180 possible recommendations for each exercise, but usually only a few slides for each exercise are labeled as recommendable. Suppose there is an exercise for which we selected three pages as appropriate. Even if our implementation incorrectly recommended a random slide for this exercise, it would still classify 176 slides as not recommendable, adding 176 entries to the true negative cell. The baseline approach reinforces this effect by restricting the recommendation to only the pages with the highest cosine value, thereby further reducing the number of recommended slides. Furthermore, for the baseline evaluation, the idf value of a keyword can also be calculated by considering, for all available pages, the number of occurrences of the keyword. Thus, we will refer to the sub-chapter-wise idf calculation as idf$_{sub}$, and the collection-wise idf values will be referred to as idf$_{col}$. The result for the baseline approach in combination with the idf$_{col}$ calculation is shown in Table 4.9. The variation in the idf calculation is barely showing in the classification since the idf$_{col}$ method predicts 66 instances as positive compared to 70 positive predictions in the idf$_{sub}$ computation. The difference in negative predictions is also negligible, with 11.810 negative predictions in Table 4.8 and 11.814 in Table 4.9.

	pred. pos.	pred neg.
actual pos.	38 (TP)	178 (FN)
acutal neg.	32 (FP)	11632 (TN)
total	70	11.810

Table 4.8: Confusion matrix of baseline approach with idf$_{sub}$ Obionwu et al. [2022a]

	pred. pos.	pred neg.
actual pos.	38 (TP)	178 (FN)
acutal neg.	28 (FP)	11636 (TN)
toal	66	11.814

Table 4.9: Confusion matrix of baseline approach with idf$_{col}$ Obionwu et al. [2022a]

Table 4.10 shows the performance metrics with respect to the confusion matrices from Table 4.8 and 4.9. The accuracy of both idf calculations is rather high, with 0.982 using idf$_{sub}$ and 0.983 using idf$_{col}$. This is mostly due to the previously described fact that most of the pages are correctly classified as true negatives. The precision value of idf$_{sub}$ is slightly lower than the precision of the idf$_{col}$ method with 0.576. That means slightly more than half of our baseline's recommendations are correct. Each of the remaining metrics recall, F-measure, and $F_{\beta=0.5}$ are rather similar for both idf computations. The recall value for both methods is 0.176, which implies that around 17% of the slides classified as recommendable are selected by our system. The F-measure, which is influenced equally by the precision and recall metrics, reaches 0.266 with the idf$_{sub}$ and 0.27 with the idf$_{col}$.

The most important metric in our use case is $F_{\beta=0.5}$ which equates to 0.383 for the idf$_{sub}$ calculation and 0.396 for the idf$_{col}$.

The performance metrics of the collection-wise idf approach are slightly better than those of the subchapter-wise idf. Therefore, we will focus on the collection-wise idf calculation technique in the next section because the peak performance will be achieved by using idf$_{col}$. Hence, when referring to the baseline approach, we mean the baseline approach using the collection-wise idf from now on. In the following sections, we will try to improve the recommendation performance by utilizing our already introduced optimizations of join detection, clustering of keywords, and minimal cosine values.

4.2.9 Detecting joins

Using pattern matching in combination with a list of relevant SQL keywords is insufficient for the detection of joins that are formulated with the WHERE keyword.

Metric	value_{sub}	value_{col}
Accuracy	0.982	0.983
Precision	0.543	0.576
Recall	0.176	0.176
F-Measure	0.266	0.27
$F_{\beta=0.5}$	0.383	0.396

Table 4.10: Performance metrics for baseline approach Obionwu et al. [2022a]

Hence, we used a different method for join detection described in Section 4.2.9. Figure 4.11 shows the confusion matrix of our join detection alongside the rate of change compared to the baseline approach using collection-wise idf values.

Applying join detection yields a positive effect on the classification results. The number of true positive predictions increased by 15.8% while the number of false-positive predictions decreased by 21.4%. The join detection also has a beneficial effect on the false negative and true negative predictions, although they profited percentage-wise significantly less compared to the positive predictions. This is because the number of instances in the column of negative predictions is higher than the number of instances in the second column, and thus the false negative and true negative table cells are less affected percentage-wise. The performance metrics of the baseline approach with and without the join detection are shown in Figure 4.12. Each of the performance metrics increased with the activated join detection. The accuracy value increased almost negligibly from 0.983 to 0.984. The recall and F-measure improved more, with 0.176 to 0.204 and 0.27 to 0.312, respectively. Especially noteworthy is the increase in the precision value from 0.576 to 0.667 due to enabling join detection. In the next section, we show the effect of keyword clustering.

idf_{col}	pred. pos.	pred neg.
True pos.	44 $\uparrow$ 15.8%	172 $\downarrow$ 3.4%
True neg.	22 $\downarrow$ 21.4%	11642 $\uparrow$ 0.052%
total	66 ± 0	11.814 ± 0

Table 4.11: Results of activated join detection compared with baseline approach Obionwu et al. [2022a]

Metric	$\neg$(join detection)	join detection
Accuracy	0.983	0.984
Precision	0.576	0.667
Recall	0.176	0.204
F-Measure	0.27	0.312
$\text{F}_{\beta=0.5}$	0.396	0.459

Table 4.12: Performance comparison with and without join detection Obionwu et al. [2022a]

4.2.10 Clustering keywords

In Section 4.2.5.3, we introduced keyword clustering. In this section, we show how keyword clustering improves our slide recommendations. One strategy to find viable cluster candidates is to take a look at exercises for which there is no correct recommendation. These exercises are then analyzed based on the respective keywords they have in common with their desired recommendations. These keywords that are featured in both the exercise and its desired recommendations are then selected to be part of the cluster. This approach could not be successfully applied in our use case since clustering these shared keywords had a negative effect on other recommendations and thus decreased performance.

In our strategy, we analyzed our data set and identified specific keywords that needed to be clustered. This cluster consists of the keywords <,>=, and SELECT. This process of choosing suitable keywords is manual. Table 4.13 depicts the confusion matrix for this clustering approach. The clustering leads to 4.6% more true positive predictions, while the false positive recommendations were lowered by 9.1%. The performance metrics are displayed in Table 4.14 alongside the comparison to the former best approach without cluster usage but with join detection. The application of the cluster causes the accuracy to increase from 0.983 to 0.984. More notably, the precision rises from 0.667 to 0.697. The recall value increases slightly, from 0.204 to 0.213. The improvement of both the recall and precision values causes the F_β value to increase from 0.459 to 0.479. Especially, the improved precision and F_β metrics imply that the clustering of keywords enables our system to recommend useful slides to the students.

	pred. pos.	pred neg.
True pos.	46 $\uparrow$ 4.6%	170 $\downarrow$ 1.2%
True neg.	20 $\downarrow$ 9.1%	11644 $\uparrow$ 0.02%
total	66 ± 0	11.814 ± 0

Table 4.13: Confusion matrix of cluster application Obionwu et al. [2022a]

Metric	¬cluster	cluster
Accuracy	0.984	0.984
Precision	0.667	0.697
Recall	0.204	0.213
F-Measure	0.312	0.326
$F_{\beta=0.5}$	0.459	0.479

Table 4.14: Performance comparison with and without clustering Obionwu et al. [2022a]

The improved performance due to the clustering is attributable to two more mappings between exercises and slides that are now done correctly. One of the exercises for which the prototype found the correct recommendation will be referred to as task E and is shown below:

The keyword analysis for task E yields the results shown in Table 4.15 with the recognized keywords SELECT, GROUP BY, GROUP, AS, MIN, and ALL. The

```
SELECT job, MIN (ALL salary) AS min_salary
FROM employee
GROUP BY job;
```

MIN and ALL keywords have the highest tf*idf value with 1.217 assigned to it and therefore they are the most important keywords for this exercise.

task G				
keywords	count	tf	idf	tf*idf
select	1	1	0.087	0.087
group by	1	1	1.121	1.121
group	1	1	1.121	1.121
as	1	1	0.405	0.405
min	1	1	1.217	1.217
all	1	1	1.217	1.217

Table 4.15: Keyword analysis for task E from the SQLValidator Obionwu et al. [2022a]

The recommendation before clustering is incorrect since the recommended page is not helpful to the students. The chosen page is the twenty-fourth page of the ninth chapter, "Views and Access Control." Page 24 contains information about the problems with aggregation views, although task E does not feature any information about views. Hence, the recommendation on page 24 is not useful for students who are challenged by Exercise E. Table 4.16 displays the keyword analysis for page 24. The keywords

WHERE and HAVING were recognized once, and the keywords SELECT, GROUP BY, <, MIN, and GROUP twice. The highest tf*idf values are reached by the keywords MIN at 1.556 and < at 1.352.

Chapter: 9, Page: 24, Cosine: 0.682				
keywords	count	tf	idf	tf*idf
select	2	1	0.347	0.347
where	1	0.5	0.484	0.242
group by	2	1	1.109	1.109
group	2	1	1.051	1.051
having	1	0.5	1.301	0.651
<	2	1	1.352	1.352
min	2	1	1.556	1.556

Table 4.16: Keyword analysis of the incorrectly referred page 24 from chapter nine Obionwu et al. [2022a]

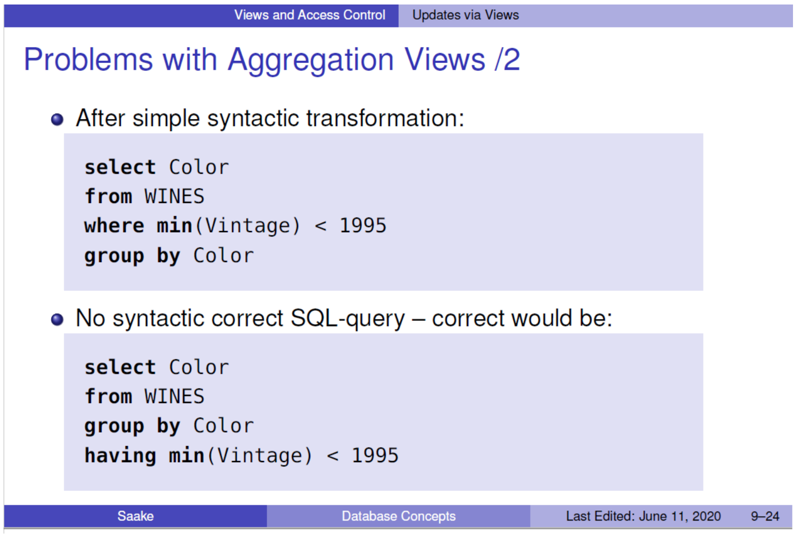


Figure 4.14: Incorrect recommendation of page 24 from chapter nine to task E before clustering Obionwu et al. [2022a]

The recommendation to task E should contain information as to how the GROUP BY keyword can be used to aggregate data. Instead of recommending page 24 of the ninth chapter, the independent labelers chose page 61 of the sixth chapter, displayed in Figure 4.15 as a good fit for task E since it visualizes the process of using the GROUP BY clause.

The keyword analysis for our desired recommendation is shown in Table 4.17. Page 61 contains the three SQL keywords AND, GROUP BY, and GROUP once, with the GROUP BY keyword reaching a tf*idf of 1.109 and the GROUP clause following at 1.051. The comparison between Table 4.16 and Table 4.17 shows that the cosine similarity of page 61 with 0.625 is lower than the cosine value of 0.682 from the current recommendation. In order to change the recommendation from page 24 to page 61, we need to influence the cosine similarity between task E and the slides by creating a suitable cluster. At first, we tried to create clusters of keywords that are both contained in task E and our desired recommendation of page 61. Unfortunately, every cluster that used this approach led to a performance decrease, so a different strategy was necessary in order to establish a correct mapping for exercise E. Instead of increasing the cosine similarity of our desired recommendation, we can also decrease the cosine similarity of the current recommendation by utilizing a cluster. Page 24 contains the $<$ and SELECT keywords, with the SELECT clause also shared with task E. We have a cluster in use that contains the SELECT and $<$ statements and thus changes the tf*idf values of page 24. The performance evaluation of page 24 is shown in 4.18 with the SELECT keyword having a tf*idf value of 1.352 instead of 0.347. The increased tf*idf value causes the similarity between task E and page 24 to shrink, and thus the new cosine value equals 0.625. The cosine similarity of page 61 does not change with respect to exercise E because our desired recommendation does not share any keywords with the cluster. The unchanged similarity of 0.64 is sufficient in order to be chosen for recommendation in task E, since the former cosine value of page 24 decreased. Our cluster contains three keywords in total, with the $>=$ clause not being mentioned yet.

In our research, we observed a performance decrease when using a cluster that only contains the SELECT and < keywords. We believe that using the cluster without the >=yields a side effect to the other exercises, which is why we chose to include >= in our cluster.

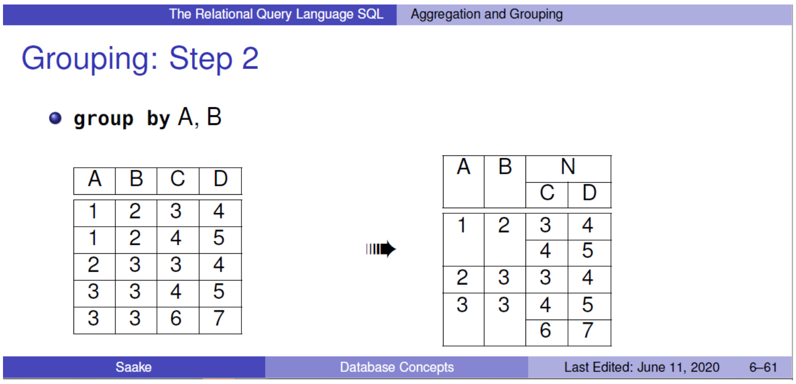


Figure 4.15: Page 61 of the sixth chapter which should be chosen for recommendation Obionwu et al. [2022a]

Chapter: 6, Page: 61, Cosine: 0.64				
keywords	count	tf	idf	tf*idf
GROUP BY	1	1	1.109	1.109
GROUP	1	1	1.051	1.051
AND	2	1	0.499	0.499

Table 4.17: Keyword analysis for page 61 from chapter six Obionwu et al. [2022a]

Chapter: 9, Page: 24, Cosine: 0.625				
keywords	count	tf	idf	tf*idf
select	2	1	0.347	1.352
where	1	0.5	0.484	0.242
group by	2	1	1.109	1.109
group	2	1	1.051	1.051
having	1	0.5	1.301	0.651
<	2	1	1.352	1.352
min	2	1	1.556	1.556

Table 4.18: Keyword analysis for page 24 from chapter nine after clustering Obionwu et al. [2022a]

4.2.11 Minimal cosine value

In the course of optimizing our recommendation, we have observed that there are some exercises for which the best slide recommendation has a rather low cosine similarity. These cosine values range from $[0-1]$ with zero meaning no similarity and 1.0 being almost identical. Thus, a recommendation with a cosine value of 0.1 is most likely not very helpful to the students. In this type of scenario, where the best recommendation has a low cosine value, we resolved not to make any recommendations as they would not be helpful to students. To enforce this rule, we included an option to set a minimal cosine similarity that has to be reached for any page in order to be recommended Agrawal and Phatak [2013]; Muflikhah and Baharudin [2009]; Strehl et al. [2000]. Table 4.19 shows results of a further extension of the baseline approach with a minimal cosine value at 0.2, 0.4, and 0.6. The second column of the table contains the performance for the baseline approach using idf_{col} with the extensions introduced in Sections 4.2.9 and 4.2.10. The baseline approach uses a minimal cosine value of zero because there is no threshold implemented that restricts pages with low cosine values from being recommended. The third column displays the performance values for a minimal cosine value of 0.2. A comparison between the baseline approach and a minimal cosine value of 0.2 shows that the minimal cosine value has a small positive effect on performance. The precision, F-measure, and $F_{\beta=0.5}$ slightly increase while the recall value remains unchanged. This implies that the minimal cosine value filtered out slides that were incorrectly recommended. Using a minimal cosine value of 0.4 increases the performance further with a precision value of 0.767 and an $F_{\beta=0.5}$ of 0.505. The recall value is still unchanged, which means that by setting the minimal cosine value to 0.4, there are just incorrect recommendations being filtered out. The fifth column displays the performance of setting the cosine value to 0.6. There is a slight decrease in all metrics compared to the previous column. This is because increasing the cosine value eventually leads to correct recommendations being filtered out. Since our focus is on maximizing the F_β and precision values, we set the minimal cosine value to 0.4.

Metric	baseline	cos-min$_{0.2}$	cos-min$_{0.4}$	cos-min$_{0.6}$
Accuracy	0.984	0.984	0.985	0.984
Precision	0.697	0.708	0.767	0.763
Recall	0.213	0.213	0.213	0.208
F-Measure	0.326	0.327	0.333	0.327
$F_{\beta=0.5}$	0.479	0.483	0.505	0.498

Table 4.19: Performance metrics for evaluated minimal cosine values Obionwu et al. [2022a]

4.2.12 Discussion

The assessment demonstrated that a logical correlation between course slides and SQL tasks is attainable. Our system offers valuable feedback for various activities and rarely suggests slides that are not intended to be suggested. Augmenting the baseline approach with supplementary techniques resulted in a substantial enhancement of the system's performance, particularly in terms of precision. Therefore, the likelihood of the system causing confusion among students by suggesting irrelevant slides has decreased. The performance review indicates that we successfully reached our objectives by offering a solution to assist pupils struggling with comprehending the structured query language programming. Nevertheless, there are still issues regarding the referral procedure.

4.2.12.1 Usage of keywords in the English language

We faced an issue during our research where certain SQL terms like AND, IN, or AS are commonly utilized in the English language outside of the SQL context. Counting the frequency of keywords in SQL becomes problematic due to the teaching slides being in English. Our recommendation process faces a significant challenge when a keyword is infrequently found in the corpus, resulting in a high idf weight that heavily influences the page's subject determination. If a page has numerous SQL keywords, the impact of one keyword being erroneously identified can be lessened by the tf*idf values of the other keywords. A page with few keywords may cause non-SQL related terms like "IN" or "AND" to confuse the recommendation process for that page. Figure 4.16 shows page 40 of the ninth chapter, where the IN keyword is present but not utilized in any SQL context. The keyword analysis of this page is presented in Table 4.20. The terms JOIN and IN are identified, and their tf*idf value of 0.699 is displayed in the fifth column. If the IN keyword is mistakenly identified on page 40 of the ninth chapter, it could lead to tying this page to an activity that pertains to understanding the proper usage of the IN keyword, even though the page itself does not contain any information concerning the IN keyword.

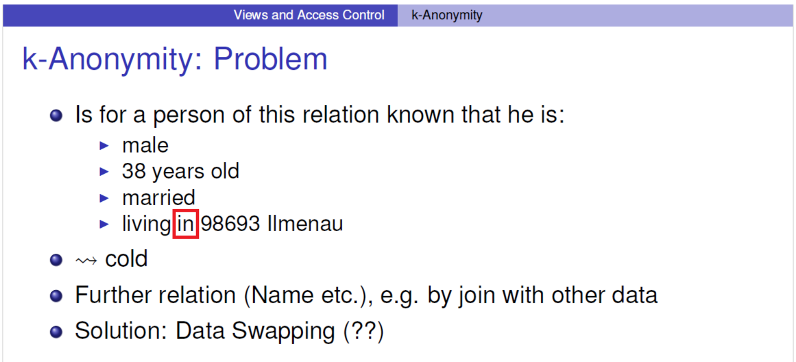


Figure 4.16: Slide 40 from chapter nine that contains a SQL keyword not used in a SQL context

Chapter: 9, Page: 40				
keywords	count	tf	idf	tf*idf
in	1	1	0.699	0.699
join	1	1	0.699	0.699

Table 4.20: Tf*idf analysis for page 40 from chapter nine which features a SQL keyword in a non SQL context Obionwu et al. [2022a]

Figure 4.16 displays page 40 of the ninth chapter, which contains the IN keyword, although it is not used in any SQL context. Table 4.20 presents the keyword analysis of this page. The keywords JOIN and IN are recognized, and the tf*idf value for both is shown in the fifth column with 0.699.

By incorrectly recognizing the IN keyword on page 40 of the ninth chapter, it might result in linking this page to an exercise that requires the students to understand the correct usage of the IN keyword, although the page does not hold any information about the IN keyword.

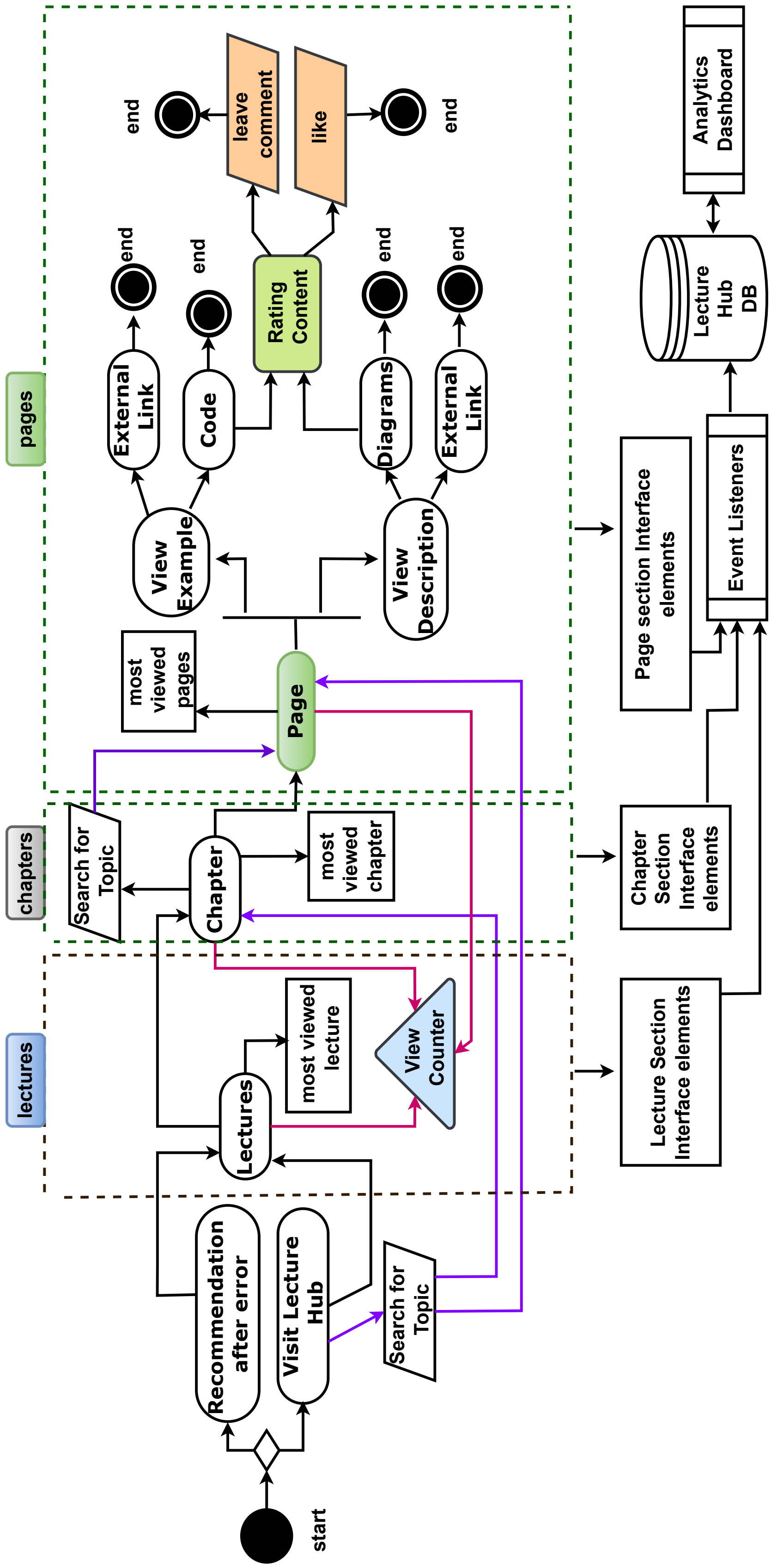


Figure 4.17: Lecture Hub Obionwu et al. [2022a]

4.2.12.2 Tracking Learning Engagements

From the activity analysis we derived from Moodle, the instructional feedback system is insufficient because recommendations that point to a lecture slide are an end in themselves. It is important to know how students interact with lecture materials and the recommendations they receive. This information can help improve the effectiveness of the feedback system and enhance student learning outcomes. Additionally, understanding student engagement with course materials can provide insights into areas for improvement in teaching methods. To this end, we developed the lecture hub, shown in figure 4.17. The lecture hub gives us the ability to track student activity in real time. It has a lecture page, and each lecture has several chapters, and each chapter has multiple pages. Click, hover, scroll, etc., and other event listeners are employed to track student activity on this platform. By analyzing the data collected from the lecture hub, educators can identify which parts of the course material are most engaging for students and which may need further clarification or reinforcement. This data-driven approach allows for targeted interventions to enhance student comprehension and retention of the material. Thus, if a student encounters an error, receives a recommendation, and clicks the link that takes them to the lecture hub at time "t1," we will be able to track their interaction until they return for a retrial at time "tn". Thus, we now have the capacity to track the provenance of a student's learning interaction. The lecture hub also has a rating system that allows us to receive feedback from students on their experiences, and all the lecture hub activities are displayed in the SQLValidator dashboard.

In general, the recommendation, while it behaves according to design, is not human-like. In a traditional instructional feedback setting, the instructor, in most cases, already expects that the student has already studied the lecture material before requesting assistance. Thus, pointing back to the lecture slides, while it helps reinforce the importance of the knowledge described on the recommended page, does not, in most cases, render the expected assistance. To truly provide effective feedback, the recommendation should offer more personalized and tailored guidance based on the specific needs and understanding of the student. This can help bridge the gap between simply pointing out information and actually helping the student grasp and apply the concepts effectively. Thus, it becomes important to understand the students intent. Understanding the student's intent can guide the feedback towards addressing their individual learning goals and challenges. By personalizing the recommendations, educators can better support students in achieving a deeper understanding of the material. We also want a system or an agent that can act as an instructor. This agent must be capable of understanding the lecture material and providing tailored feedback to each student based on their specific needs and learning styles. This personalized approach can enhance the overall learning experience and improve student outcomes in the long run. In the next section, the conversational agent is introduced.

4.3 The Conversational Agent

Our instructional feedback strategy was centered on recommending suitable lecture slides for the respective exercise tasks. For this, we developed a system that employed keyword analysis and cosine similarity. However, in an effective traditional learning scenario with small class sizes, it is possible to tailor feedback to each student's specific needs, address their individual learning gaps, and enhance their understanding of programming concepts where they are deficient De Lorenzis et al. [2023]. To replicate this personalized form of instructional feedback and also make it automatic for the large population of students using our learning management system, we currently employ generative pre-trained transformers and associated language models to create an intelligent conversational agent. Conversational agents have attracted considerable attention in recent years, mostly due to their capacity to participate in natural language exchanges with humans. They offer feedback through comments, recommendations, or responses given to users during their interactions. The quality of feedback greatly affects the user's experience. Thus, a conversational agent's response should demonstrate coherence, relevance, and meaningfulness while delivering factual information, answering a user's query, or giving directions. This agent can adapt to the unique needs of each student, providing personalized support and guidance throughout their learning journey. Also, replying with clarity facilitates interactions with reduced ambiguity, ultimately resulting in more fruitful talks. Giving human-like feedback fosters trust by facilitating interactions that simulate real discussions, a crucial aspect particularly in fields like healthcare, customer support, and education Babu and Akshara [2024]. To ensure that the presented responses are factual, we employ knowledge graphs, word embeddings, and semantic search. Therefore, we focus our current approach on retrieving information and plan to enhance the retrieved data in the future. Our system employs a Bidirectional Encoder Representations from Transformers (BERT) model in decoding the intent of the student's query Devlin et al. [2018]. Based on the decoded intent, the system then retrieves relevant information from vector embeddings created by the knowledge graph. This ensures that students receive reliable and comprehensive answers to their questions. Additionally, we adapt and update the knowledge graph regularly, ensuring that the information provided remains up-to-date and relevant.In the next section, we discuss dialogue systems, as they are central to developing a human-like conversational agent.

4.3.1 Dialogue System

Dialogue systems are computational systems that engage in natural language interactions with humans. These systems are designed to understand and respond to user input in a way that simulates human conversation. Dialogue systems are commonly used in customer service, virtual assistants, and other applications where interaction with users is necessary Arora et al. [2013]; Deriu et al. [2021]; Ma et al. [2021]. Conventional dialogue systems mostly rely on rule-based and non-neural machine learning approaches. Rule-based systems are simple to construct and can provide natural responses, which is why they were widely used early on in commercial products. However, the communication flows in these systems are predetermined, limiting the applications to specific scenarios Arora et al. [2013]; Ma et al. [2021]. They mainly consist of an input decoder, natural language understanding, a dialogue manager, a domain-specific component, a response generator, and an output renderer, as shown in fig. 4.18.

The input decoder component is responsible for identifying and interpreting the input. It transforms the input into plain text. This component is exclusive to text-based

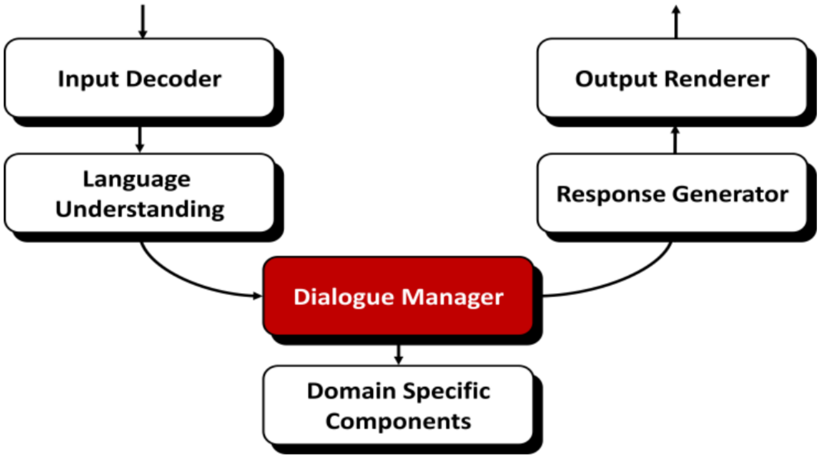


Figure 4.18: Components of dialogue system Arora et al. [2013]

dialogue systems and transcribs spoken words, or user utterances, into a sequence of text. Speech recognition is required for this function; thus, these systems employ automatic speech recognition (ASR) Recognition—ASR [2013], computer speech recognition Schroeder [2004], or simply speech-to-text (STT) Nagdewani and Jain [2020]. In addition to speech, the conversation system can accept inputs such as gestures and handwriting. The natural language understanding block, as the name implies, attempts to comprehend what the user is trying to communicate. The system transforms the word sequence into a semantic representation for the dialogue manager's usage. This component involves the utilization of morphology, syntax, and semantics. Morphology is the examination of the arrangement and substance of word forms Lieber [2021]. Once the keywords are identified and their meaning determined, they are passed to the dialogue manager. The dialogue manager handles every part of the conversation. The system analyzes the user's text to understand its meaning within context and generates a semantic representation for the system's answer. It consists of the following components: dialogue model, user model, knowledge base, discourse manager, reference resolver, and grounding module. With these components, it carries out several tasks, which include:

- Preserves the record of conversation
- Address improperly formatted and unidentified text.
- Access the information saved in files or a database.
- Determines the optimal user reaction and oversees initiative and system feedback.
- Conversation Analysis

In order to convert input queries from the dialogue manager's internal representation to the format that the external system uses, such as structured query language (SQL), the dialogue manager typically interfaces with external software, like a database or an expert system. The domain-specific components manage this interface. This can be managed using the natural language query processing system, which converts natural language into SQL queries. This component assembles the message intended for transmission by the user. It determines the content, structure, wording, and syntax of the communication. Existing systems employ basic techniques, like inserting obtained data into predetermined spaces inside a template. The response generation component creates the message, which the speech generation module then turns into spoken language. Two methods can be utilized for speech generation. One method is to utilize premade canned speech that can be customized with slots for inserting recovered or previously recorded samples, such as "Welcome, how can I assist you?" The second method is text-to-speech synthesis Arora et al. [2013]; Kaur and Singh [2023].

Compared to recent conversational artificial intelligence platforms and systems, which make use of a wide range of components, including large language models such as the Generative Pre-trained Transformer (GPT-3.5) Ye et al. [2023], which have greatly improved natural language interactions, intent classification, automated responses or feedback, etc., they have contributed to the overall improved user experience. These advancements have led to more seamless and human-like interactions with AI systems, ultimately enhancing user engagement and satisfaction. Additionally, the integration of these components has also allowed for more personalized and contextually relevant responses, further improving the overall user experience. Conversational agents can answer questions, give advice, and participate in dynamic dialogues since these components combined form the foundation of conversational agents. This enables conversational agents to be utilized in a broad variety of contexts. , such as customer service, healthcare, and education, making them versatile tools for enhancing communication and problem-solving in various industries. As technology continues to advance, conversational agents are expected to become even more sophisticated and capable of understanding and responding to complex human interactions. Table 4.21 gives a comparison between conversational agents and dialogue systems.

4.3.2 Conversational Agents Vs Dialogue Systems

In summary, according to the information in Table 4.21, conversational agents are a specific type of dialogue system that emphasizes individual interactions. Dialogue systems exceed these agents by overseeing and coordinating coherent and purposeful conversations. Throughout the remainder of this thesis, we will use the terms conversational agents and dialogue systems interchangeably for clarity.

4.3.3 Types of Conversational Agents

When classifying conversational agents, it is crucial to evaluate the technology and design concepts they utilize. Although numerous conversational AI/chatbot solutions are available to corporations, not all may be suitable for an organization's needs due to their unique characteristics. Earlier, a broad categorization of conversational agents (CAs) was presented, dividing them into task-oriented CAs and casual chatbots. The article by Dilmegani [2023] categorizes conversational agents into five main sub-categories.

4.3.3.1 Task-Oriented Agents

Task-oriented agents are specifically designed to perform specific tasks for users. Their goal is to provide reliable and relevant answers to certain questions, occasionally

Conversational Agent Vs. Dialogue Systems		
Focus	primarily emphasize the interaction between a user and a computer system through natural language.	aim to maintain coherence, context, and continuity across multiple exchanges.
Scope	designed to simulate human-like conversations, often serving specific purposes such as answering queries, providing information, or executing tasks.	encompass a broader framework designed to manage and structure extended conversations or dialogues.
Functionality	uses NLP and AI to comprehend user inputs and generate appropriate responses.	uses more sophisticated architectures and methodologies to understand user input, generate responses, and maintain the conversation's thread, ensuring context and purpose are maintained throughout..
Purpose	focus on individual interactions, addressing immediate user queries or commands without considering broader contextual continuity.	create more natural, purposeful, and meaningful conversations, considering the broader context of the interaction beyond isolated queries or commands.

Table 4.21: Difference between Conversational Agent and Dialogue System

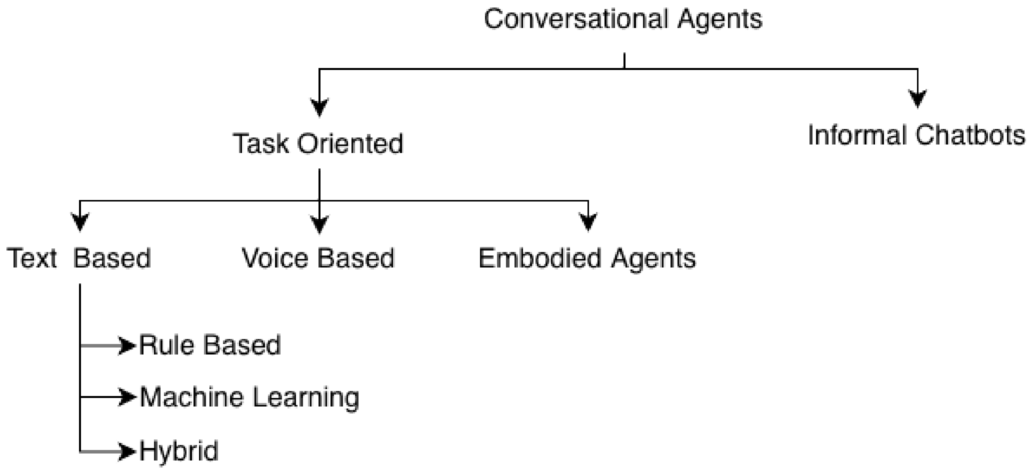


Figure 4.19: Types of Conversational Agents

requiring precise database searches and data analysis that can be loosely defined as cognitive processingLinnenbrink and Pintrich [2004]. These types of conversational agents, also referred to as goal-oriented agents, are designed to offer services and assistance for particular tasks like booking a flight, ordering food, or making reservations. They employ advanced language processing techniques such as natural language understanding (NLU), natural language generation (NLG), and machine learning (ML) to recognize and respond to customers' questions. Task-oriented conversational agents adopt a systematic conversation framework to obtain the key information needed to perform the defined task. For example, when a user wants to book a hotel, the agent will ask for the departure location, destination, travel date, preferred airlines, and any other necessary details before processing the request. Conversational agents can be further classified according to their tasks or the methods they use (Singh, 2022). We can categorize the methods into rule-based, machine-learning-based, and hybrid approaches. Next, we discuss rule-based conversational agents.

Rule-Based Dialogue Systems

These systems react to user inputs according to predefined rules. The system analyzes the user's text input for specific phrases and patterns to identify the user's intention and respond accordingly. Rule-based conversational agents operate through a series of essential steps. Figure 4.20 displays a fundamental rule-based design. The system examines the user's input, like a text message or voice command, to detect pertinent keywords and patterns. Subsequently, the agent chooses a predetermined reaction from its set of rules based on the identified cues. This response may involve delivering information, addressing inquiries, or directing the user through a particular procedure. These agents excel at managing common inquiries or routine procedures with consistency.

Rule-based conversational agents are constrained when engaging in open-ended or dynamic conversations. They excel in scenarios that necessitate well-defined and foreseeable interactions. These systems are easier to model and have a high level of predictability. The inflexibility of rule-based conversational agents can impede

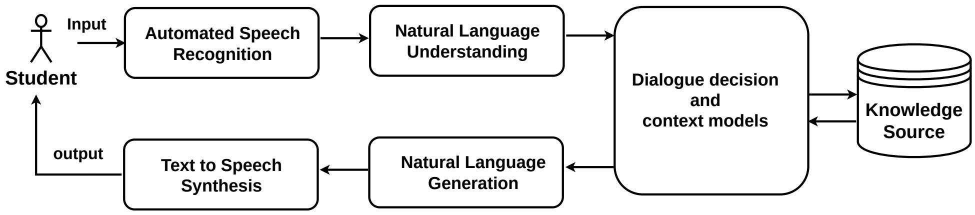


Figure 4.20: Typical Rule-based Architecture McTear [2021]

conversation flow and complicate customization, especially when the system integrates multiple rules. The study by Agarwal et al. Agarwal and Wadhwa [2020] talks about techniques like pattern matching, parsing, Markov chain models, and semantics that are used in rule-based conversational agents to have natural language conversations.

4.3.3.2 Artificial Intelligence Based Conversational Agents

AI-driven Conversational AI, commonly known as ML Conversational AI embodies a more sophisticated and adaptable method for engaging in conversations. These agents utilize machine learning, foundation models, and natural language processing to comprehend and provide human-like responses in conversational settings Kusal et al. [2022]. Conversational AI systems based on artificial intelligence are trained using large datasets that contain both text and speech data, Agarwal and Wadhwa [2020]. The machine learns to understand and analyze human language through this data. Later, the system uses the gathered data to engage with humans in a way that imitates natural communication. The system continuously improves the quality of its responses by learning from its interactions. The study Kusal et al. [2022], emphasizes that previous conversational agents relied on either pattern-based or keyword-based methodologies for conversing in natural written language. These methods involve comparing a user's question to existing replies in a database and then sending an appropriate response back to the user.

These conversational agents demonstrated constraints, such as the incapacity to understand users' emotions and thoughts, and could not understand the user's feelings or tone. Another drawback of these conversational AIs is their sole reliance on written language, whereas humans communicate through various modalities or senses. To address these challenges, researchers employed natural language processing and machine learning techniques. These advancements have enabled implementation across various application areas and allowed for a degree of flexibility while simulating genuine conversation to some extent. The artificial intelligence methodology can be divided into retrieval-based and generative-based methods Pandey and Sharma [2023]. In general, the advancement and use of machine learning have enabled the utilization of methods rooted in neural networks. Extensive datasets train the neural network to produce appropriate and coherent replies. Input may consist of text, pictures, or spoken speech. Also, models have been created to convert speech to text, as referenced in Serban et al. [2017].

4.3.3.3 Hybrid Conversational Agents

Hybrid conversational agents are becoming a potent solution in the dynamic field of conversational AI, blending automated intelligence with human interaction. Hybrid conversational agents are leading technology in providing instructional feedback that mimics human interaction. Despite several research publications on the use of hybrid conversational agents in various fields, there is a dearth of research articles that specifically focus on defining and describing the properties of hybrid conversational agents. According to Team, hybrid conversational agents combine rule-based and machine learning-based methods to create a flexible and effective conversational AI system. The goal is to leverage the benefits of both strategies to create a conversational agent (CA) that is more durable and adaptable. Hybrid CAs include multiple significant characteristics Burgin et al. [2022]. These consist of rule-based elements that are efficient in handling typical and uncomplicated inquiries. Furthermore, machine learning (ML) and natural language processing (NLP) components help conversational agents understand and produce human-like discourse. We use intent recognition, entity recognition, and fallback techniques to ensure smooth operation. Hybrid conversational agents may continually learn, retain user context, and manage multi modal capabilities, along with other functions, McTear [2022]. They are popular due to their adaptability, operational efficiency, and individualized interactions, making them valuable in areas such as personalized learning, customer service, e-commerce, and healthcare. One of the important tasks performed by conversation agents is student chat intent classification. The next subsection discusses this.

4.3.4 Chat Intent classification

Intent analysis is a method in natural language processing (NLP) and machine learning that is employed to ascertain the fundamental goal or objective behind a written or spoken statement, usually within the framework of human-computer interactions. The main objective of intent analysis is to ascertain the specific action or information that the user is seeking when they type a query, message, or command. In conversational agents, a representative intent classification pipeline encompasses tokenization of the input query. Tokenization tasks involve fragmenting the input sentences into words and subwords. The request from fig. 4.21 will be fragmented into tokens such as "I," "am," "having," "issues," "with," "the," "submitted," "query."". ""I" "could" "not" "understand" "the" "system" feedback." These tokenized words are then vectorized. Machine learning algorithms can use word vectorization to transform words or text into numerical vectors for tasks like natural language processing and sentiment analysis. It allows computers to understand and analyze textual data by representing words as mathematical entities in a high-dimensional spaceEgger [2022]. In most current implementations, transformers, specifically the A Bidirectional Encoder Representations from Transformers (BERT) model, handle the classification block. Previous implementations employed state-of-the-art models as the support vector machines, which analyze data for classification and regression analysis. In essence, by accurately identifying the intent behind student queries, conversational agents can improve their response time and provide more personalized support. Additionally, intent classification can help track trends and patterns in student inquiries, leading to more effective resource allocation and curriculum development.

4.3.4.1 Knowledge Graph

Knowledge graphs are an organized representation of real-world things, concepts, and their interrelationships. Applications like search engines, recommendation systems,

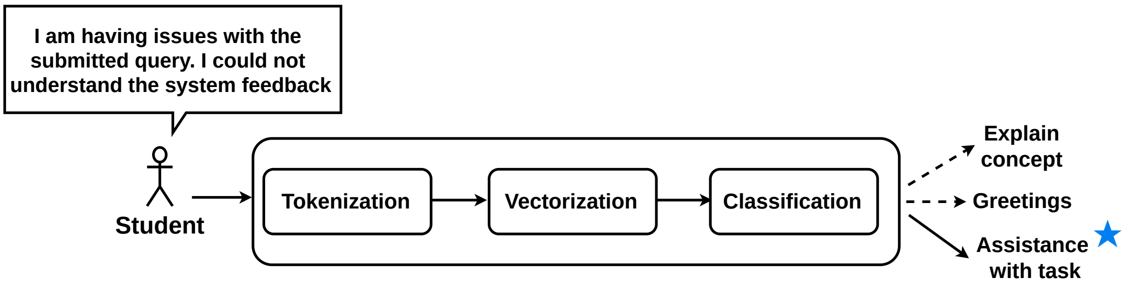


Figure 4.21: Intent Analysis from Sreelakshmi et al. [2018]

and natural language processing utilize them. Knowledge graphs help organize and connect data in a way that is easily accessible and understandable for users. They are not just storage systems for data; they create complex relationships that emphasize context, importance, and semantic interpretations. By doing this, they can add context, facilitate semantic searches, and create a cohesive data perspective. Chen et al. [2020]. The Google Knowledge Graph Fensel et al. [2020] is a major example, along with other notable companies such as Amazon, eBay, Facebook, IBM, LinkedIn, and Uber. The study focuses on integrating cognitive computing, knowledge representation and reasoning, information retrieval, natural language processing, and data mining technologies. We can classify knowledge graphs according to their content types, subject matter scope, and currency. The categories include textual, visual, multi-modal, general, domain-specific, dynamic, and static. Applications like search engines, recommendation systems, and question-answering systems widely utilize these categories. The structured representation of knowledge in knowledge graphs enables more efficient data processing and retrieval compared to traditional, unstructured data sources. The resource distribution framework (RDF) is needed to make semantic networks in knowledge graphs because it gives a standard way to show data as RDF triples, which are made up of a subject, a predicate, and an object Sintek and Decker [2002]. These triples are fundamental statements that establish a relationship between two entities through a predicate. This approach is a basic structure for encoding data in a knowledge network. RDF triples are crucial for representing the architecture of a knowledge graph. Resource distribution framework (RDF) triples consist of entities, which refer to subjects and objects, and predicates, which represent the relationships or attributes connecting these entities. Using RDF-based representation allows for the creation of a connected network of information, which forms the basis for a knowledge graph.

This interconnected structure enables machines to understand the relationships between different entities and make inferences based on the data. By utilizing RDF triples, knowledge graphs can provide valuable insights and facilitate more efficient data retrieval and analysis.

Knowledge graphs described in RDF are effective for integrating, consolidating, connecting, and reusing data. The benefits include expressiveness, efficiency, compatibility, and standardization, Knowledge Graph [2023]. Thus, integrating RDF-KGs with conversational bots improves the ability for smart and adaptable interactions. RDF offers various advantages, including enhanced expressiveness that allows for the development of intricate and interconnected data representations. Moreover, RDF-driven knowledge graphs are effective in managing complex connections and ensuring compliance with various data sources. Conversational agents leverage the structure and links inside RDF-modeled knowledge to understand user questions, navigate complex information linkages, and offer highly relevant solutions. Conversational agents benefit from the symbiotic relationship with the RDF-KG architecture, allowing them to easily access, analyze, and utilize data. This improves their capacity to offer users information that is both accurate within its context and comprehensive. When you combine conversational agents with RDF-KGs, you can take advantage of the benefits of structured data representation and create new, knowledge-driven conversational experiences. Khan [2023]; Onando [2021].

4.3.5 Retrieval-Based Conversational Agents

Retrieval-based strategies convert non-linguistic structured input queries into natural language representations, Kusal et al. [2022]. As defined by Manzoor et al. Manzoor and Jannach [2022], a retrieval-based conversational agent uses a predefined response repository and ranking model to select the most suitable response for a user's input, consisting of an offline and an online part. A retrieval-based conversational agent performs three primary functions: intent classification, entity detection, and response understanding. As explained in these research articles Sengupta et al. [2021]; Xu and Sarikaya [2014], intent classification involves determining the objective or goal of the input. Intent classification aims to comprehend the underlying purpose or motivation behind the given input query. Also important for contextually relevant feedback is entity identification, which involves the identification and isolation of individual pieces of information. These entities, when paired with intent, enable the agent to comprehensively comprehend the user's input query. A common setup for retrieval-based conversational agents is for them to first get a bunch of response options (text messages) and then use a text similarity model to figure out how similar the message is to the response options. This approach allows the conversational agent to select the most appropriate response based on the similarity score. By considering both intent and entity identification, the agent can provide more accurate and contextually relevant feedback to the user. Additionally, this retrieval-based setup enables the agent to handle a wide range of input queries effectively.

4.3.6 System Schematic

The main goal of our learning management system is to establish a favorable environment for students to practice and develop their abilities in a structured query language. Figure 4.22 shows an overview of the strategy. Each student in our learning management system belongs to a specific exercise group and can access their task activity statistics. If the frequency of errors surpasses a certain threshold, the Oracle agent will initiate an interaction. The student has the option to either accept or ignore the chat invitation from the Oracle agent.

If a student accepts the offer of assistance, the subsequent engagement will adhere to a well-defined format. Within the structured approach, the agent will present a series of inquiries to determine the student's proficiency in relation to the assigned task. Based on the responses, the agent provides guidance to the student. The student also has the ability to initiate an engagement with the Oracle agent. We categorize

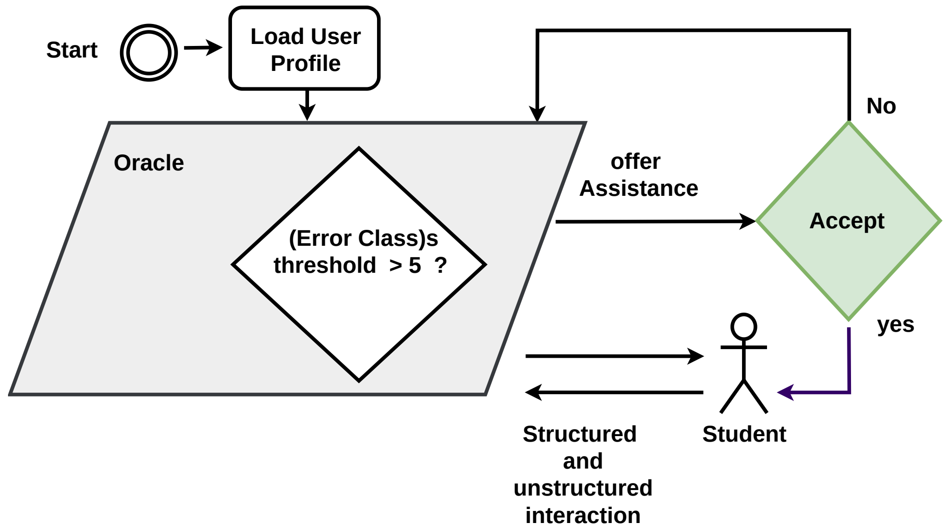


Figure 4.22: Interaction flow

this type of conversation as unstructured, following a question-and-answer format. Figure 4.23 depicts the schematic of the system. The subsequent section will provide a description of the subsystems.

4.3.7 Implementation

Our learning management system, a web-based interactive tool for learning and practicing SQL, includes this system as a component. Within our educational platform, students have the ability to engage in many activities, including the formation and testing of queries against a database, with the added benefit of receiving instant feedback. The primary objective of the project is to improve our learning platform by incorporating automated instructional feedback that closely resembles instructor guidance. This would enable students to effectively use the platform and easily acquire Structured Query Language (SQL) skills. The first stage in our pipeline involves the creation of the knowledge graph from a specified textbook. We used optical character recognition and feature representation, as discussed in the next subsection, as a strategy to achieve this.

4.3.8 Knowledge Graph Generation System

Most people use Optical Character Recognition (OCR) technology to convert printed or handwritten documents into digital format, which makes them easier to use, store, and access. We can classify an optical character recognition (OCR) system into two categories: printed character recognition and handwritten character recognition. The latter is particularly difficult due to the lack of consistency in handwritten characters. In contrast, the consistent and measurable size of printed letters reduces the difficulty of their recognition Islam et al. [2017], which is why we use it in our work.

The various phases of optical character recognition are as follows:

- **Pre-processing**: After images are acquired, several pre-processing procedures are employed to improve the image quality. Consequently, the photographs are more

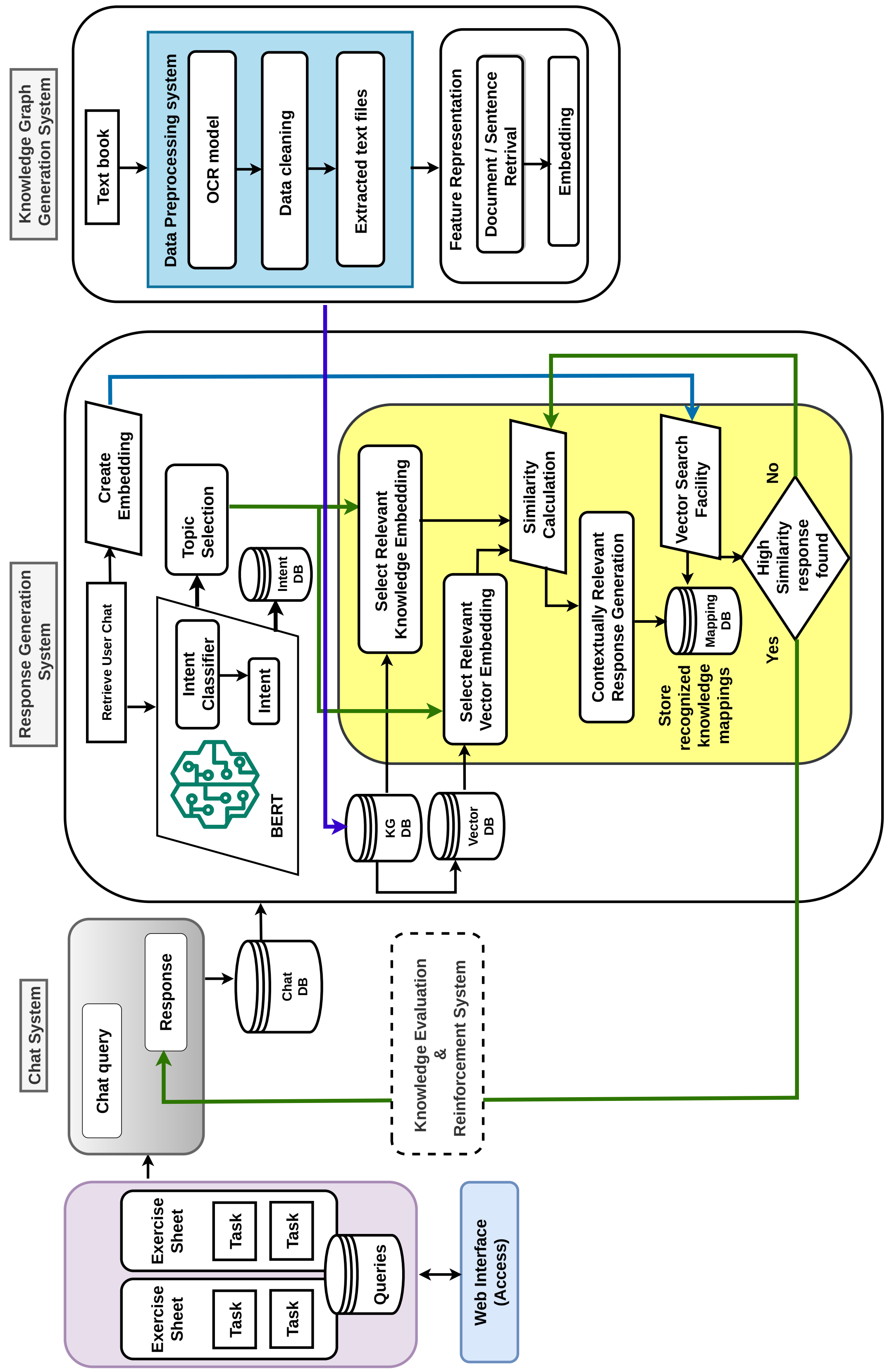


Figure 4.23: Oracle System

valuable for future use. Further in this stage, techniques such as skew reduction, thinning, and noise removal are utilized.

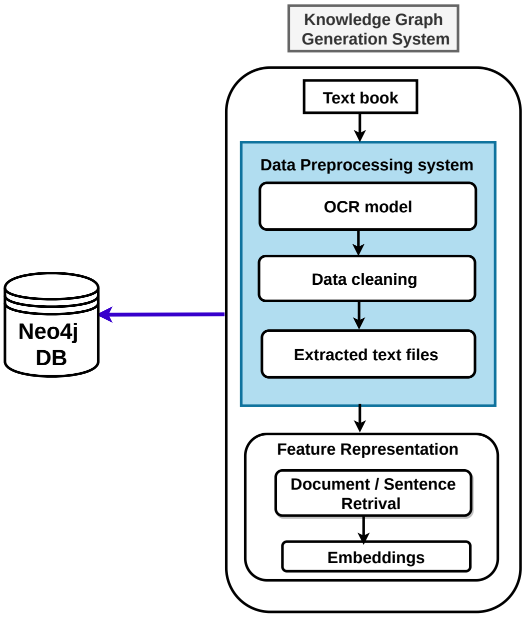


Figure 4.24: Derived Knowledge Graph

- **Segmentation**: Here the characters are separated to make it more readable.
- **Feature Extraction**: Features from the segmented images are extracted, and these features aid in character recognition.
- **Classification**: Once the features are extracted, a classification algorithm is applied to identify and categorize the characters based on their unique characteristics. This step plays a crucial role in accurately recognizing and distinguishing different characters.
- **Post-processing**: Extracted features from the segmented images contribute to the process of character recognition. Following the classification process, post-processing techniques, such as error correction and verification, are employed to enhance the precision of character recognition. These strategies aid in reducing any misinterpretations or errors that may have arisen in the preceding steps.

As shown in figure 4.23, the contents in *keyword.xlsx* are used as a reference for extracting information from the textbook and are imported into a Data Frame *keyword_mapping*, which contains columns such as topic, subtopic, and page_numbers. Each row in *keyword_mapping* is analyzed, and the page numbers are utilized to transform the appropriate pages from the textbook into images using the function *conv_pdf_to_image*. After the conversion, the text is extracted using the *pytesseract.image_to_string* method and is repeated for each subtopic, culminating in comprehensive information extraction. The extracted content is subsequently compiled into *data_dict*, which is a nested dictionary. Top-level keys represent topics. Second-level keys represent subtopics. Subtopic entry includes page numbers and extracted texts. Algorithm 1 illustrates this description. The next step is feature representation, which aims to extract meaningful and informative features from the data, eliminating the need for manual feature engineering. By learning representations directly from the data, feature learning algorithms can adapt to different

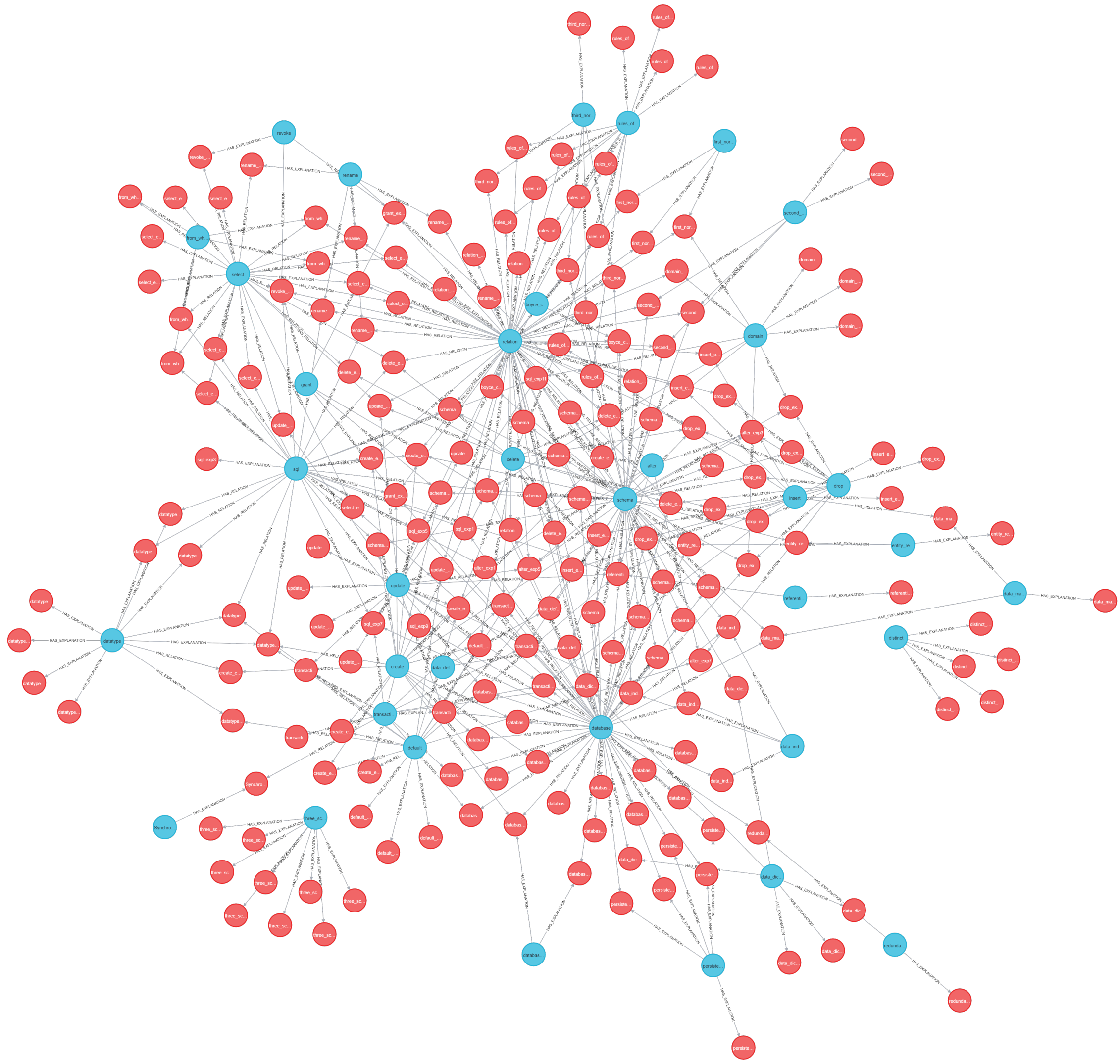


Figure 4.25: Derived Knowledge Graph

tasks and improve performance in various domains. As elaborated in Algorithm 2, *topic.txt* contains the text file that contains the description of the mentioned topic, which is used to create the knowledge graph. Figure 4.25 shows a section of the generated knowledge graph. It currently has 196 nodes, 30 labels, 166 explanations, 277 relationships, and 111 relationships.

Each line of the text file is used to create the corresponding line embedding using the function *create_embedding*. *dict_elem* dictionary contains the key and the values as the line and its corresponding embedding. The key is just the line number. The final dictionary *node_data* contains the key as the topic name and the value as the *dict_elem* dictionary. This dictionary is then used to create nodes and their relations in the knowledge graph, which is used in the response generation system discussed in the next subsection.

4.3.9 Response Generation System

Once a learner begins using our system, we generate embeddings for their queries. The embeddings are compared to the embeddings stored in the vector database.

Algorithm 4 Data Preprocessing

Require: textbook, keyword.xlsx

Initialize dictionary $data\_dict = \{\}$
$keyword\_mapping \leftarrow read\_file(keyword.xlsx)$
for row **in** $keyword\_mapping$ **do**
    extract $topic, subtopic, page\_numbers$ **from** row
    Initialize string $extracted\_text$
    for $page\_num$ **in** $page\_numbers$ **do**
        $page\_img \leftarrow conv\_pdf\_to\_image(page\_num)$
        $text \leftarrow pytesseract.image\_to\_string\ (page\_img)$
        $extracted\_text$ append $text$
    end for
    $data\_dict\ append$
    $\{topic : [subtopic, page\_num, extracted\_text]\}$
end for
return $data\_dict$

Algorithm 5 Feature Representation

Require: text file $topic.txt$

foreach $topic.txt$ **do**
initialize dictionary $node\_data$
$node\_data \leftarrow \{$*'label':topic*$\}$
$file\_lines \leftarrow read\_lines(topic.txt)$
initialize $count \leftarrow 0$
for $lines$ **in** $file\_lines$ **do**
    $count \leftarrow count + 1$
    $line\_emb \leftarrow create\_embedding(line)$
    initialize dictionary $dict\_elem$
    $key \leftarrow generate\_key(count)$
    $dict\_elem$ **append** {key:{'disc':
    $line,' emb' : emb\}\}$
    $node\_data$ **append** $dict\_elem$
end for
$create\_node(NEO4J\_CREDS, node\_data)$
initialize $start\_node \leftarrow < topic >$
initialize end_node with keys in the node_data dictionary except $topic$
$create\_relationships(NEO4J\_CREDS,$
$start\_node, end\_nodes)$
end foreach

If a response with a high degree of similarity is detected, it is forwarded to the learner as shown in Figure 4.26. If there are no embeddings with a sufficiently high degree of similarity, the underlying intention of the chat query is identified and employed to choose relevant topics that have a strong similarity, together with their corresponding vector embedding from the vector database. Using these two inputs, we do similarity assessments to choose replies that are contextually relevant. We

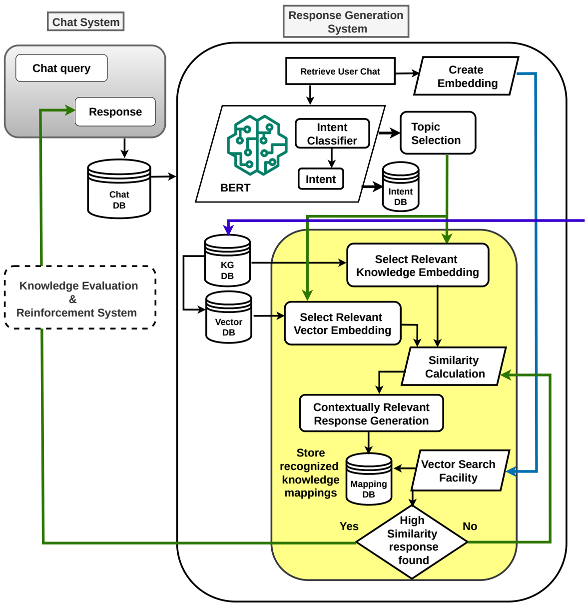


Figure 4.26: Response Generator Sub-system

store the responses in the mapping database and use them for both current and future responses. Algorithm 3 further describes the intent classification. Here, nodes is a list that contains the names of all the concept nodes in the knowledge graph, and then it is preprocessed to handle the "_" and spaces. This list is then used to classify the user question and understand the intent of the question using BERT and is stored in *intent*.

Algorithm 6 Intent Classification

Require: user_question
$nodes \leftarrow get\_concept\_nodes\_from\_kg()$
$nodes \leftarrow preprocess\_name(nodes)$
$intent \leftarrow map\_question\_to\_node(user\_question,$
$nodes)$
return $intent$

Algorithm 7 Response/Feedback Generation System

Require: user_question_embedding,intent
Initialize list $embedding, explanation$
$explanation, embedding \leftarrow get\_exp\_emb(intent)$
initialize list $similarity$
WHILE emb **in** $embedding$
$sim \leftarrow calculate\_sim(user\_question\_embedding, emb)$
$similarity$ append sim
ENDWHILE
$response \leftarrow explanation[max\_sim\_index\_embedding]$
return $response$

List embedding contains the data stored in the ***emb*** attribute of the related nodes of the intent in the knowledge graph, and explanation contains the real data of the corresponding embedding. For each embedding, the similarity is calculated with the user question embedding and is stored in the list. similarity. response holds the explanation corresponding to the maximum similarity index. The purpose of calculating the similarity between the user's question embedding and the embeddings in the knowledge graph is to find the most relevant explanation for the given query. By comparing the similarities, we can determine which explanation best matches the user's question and provide it as a response. Algorithm 4 further describes the response generation.

4.3.9.1 limitations

Conversational agents can greatly benefit from knowledge graphs (KG) as they provide several advantages, such as contextual comprehension, integration of data, management of intricate inquiries, and provision of individualized experiences. Nevertheless, capitalizing on these advantages is not without obstacles. Obstacles such as the significant cost of converting data, the intricate syntax of Knowledge Graphs cypher language, the computing requirements, privacy issues, and the management of language ambiguity can hinder their efficient application. Ensuring a proper balance of these parameters is crucial for maximizing the efficiency of conversational agents that depend on knowledge graphs. The growing body of research on knowledge graphs consistently sheds light on these problems and gives the AI community advice on how to effectively deal with them, which speeds up the progress and improvement of conversational agents' abilities.

4.3.9.2 Evaluation and Discussion

The system's evaluation matrix involves computing precision, F1 score, and accuracy using the metrics of true positives, false positives, false negatives, and false positives. If the user directly asks the conversational agent a question about a concept or a description of a certain exercise task, the agent will offer expert responses to the user depending on the recognized intent. This not only improves the user's comprehension of the SQL programming language but also the learning experience.

4.3.9.3 System and Model Evaluation

The accuracy metric Dalianis and Dalianis [2018] calculates the frequency with which a correct classification is provided for an intent. We calculate the accuracy as shown below.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

The precision metric Dalianis and Dalianis [2018] facilitates derivation of the percentage of total positive predictions that are true positive (TP) predictions of the intent of a user's query. The range of allowed accuracy values is $[0 - 1]$. In a scenario where all the expected true instances have been properly tagged as relevant, the precision value will be 1. Note that if there are no true predictions or no predicted

$$Precision = \frac{TP}{TP + FP}$$

Figure 4.27: Formula for precision

cases marked as true, the prediction values cannot be lower than zero. Figure 4.27. ”FP” designates a false prediction.

The recall metric Dalianis and Dalianis [2018] allows for the calculation of the proportion of accurate predictions out of the total number of accurate forecasts. The formula is illustrated in figure 4.28. It should be emphasized that there is a trade-off between recall and accuracy. As the value of the recall parameter falls, the precision parameter increases.

$$Recall = \frac{TP}{TP + FN}$$

Figure 4.28: Formula for recall metric

The F-measure is a metric that combines precision and recall into a single value. The F-measure has a maximum value of 1.0 and a minimum value of 0 Dalianis and Dalianis [2018]. The F-measure calculation formula is depicted in figure 4.29.

$$F = \frac{2 \times Recall \times Precision}{(Recall + Precision)}$$

Figure 4.29: Formula for F-measure metric

The baseline for our evaluation is the BERT model in its basic form. We built the baseline BERT model as the simplest model for a general natural language processing (NLP) task. It comprises the general intent datasets and therefore performs inadequately when a user asks for help solving any SQL programming language problem. Furthermore, it fails to distinguish between the SQL keywords ”create” and a task description such as ”create a database.”

The results in Table 4.22 highlight the importance of domain-specific training for conversational agents to effectively answer questions within a particular subject area. Targeted training in the specific domain a conversational agent intends to operate in can significantly enhance its performance.

Figure 4.30 shows the evaluation results for the knowledge graph. Hits@N refers to the count of elements in the ranking vector obtained from the model that are located inside the top N positions. It quantifies the ratio of accurate relations found within the top N positions of the candidate relation sets. Mean reciprocal rank (MRR) is a mathematical function that calculates the average value of the reciprocal of the

Table 4.22: Evaluation for Intent Detection

Parameters	Baseline evaluation comparison	
	Domain specific BERT	*General intent based BERT*
Precision	0.8	0.44
F1 Score	0.76	0.42
Accuracy	0.85	0.39

Hits@N	$\frac{\# Hits}{Total \# queries}$	N = 1	0.714
		N = 2	0.857
		N = 3	0.914
MRR	$\frac{1}{\lvert Q\rvert}\sum_{i=1}^{\lvert Q\rvert}\frac{1}{Rank(i)}$	0.026	
MR	$\frac{1}{\lvert Q\rvert}\sum_{i=1}^{\lvert Q\rvert} Rank(i)$	1.033	
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	0.771	
Precision	$\frac{TP}{TP+FP}$	0.833	
Recall	$\frac{TP}{TP+FN}$	0.893	
F- measure	$\frac{2.P.R}{P+R}$	0.856	

Figure 4.30: Knowledge Graph Evaluation

items contained in a vector of rankings. It serves as a metric to assess the system's performance in relation to the retrieved elements. The term mean rank (MR) refers to the average position of the correct test facts or triples within a ranking vector (i.e., the average of the projected ranks). A lower MR number indicates superior performance. However, larger values are desirable for MRR and Hits@N.

4.3.10 User Evaluation

Our platform's development primarily focuses on education, so our participants consist of students. At present, we have registered these students in our database classes. We incorporated the conversational agent into our educational platform and asked the registered students to evaluate the system by the end of the semester. We connected a Google form to our educational platform to gather the survey responses.

We designed the survey questionnaires to gather feedback on users' satisfaction levels. The interaction is in English and German.

These survey questions were based on ease of use, system interactivity, technical correctness, and usability Merdivan et al. [2020]. As shown in the figure 4.31, 50% users found the system's user interface and technical approaches to be good, and 30% found them fair. Similarly, the majority of the votes showed that users will continue to use the conversational agent for further tasks, as shown in Figure 4.32. The adoption and acceptance inquiries reveal that over 50% of respondents express their intention to consistently utilize the agent, at the very least on certain occasions. The students also expressed that their learning experience improved as they maintained engagement with the conversational agent during their online learning sessions. This is depicted in Figure 4.33.

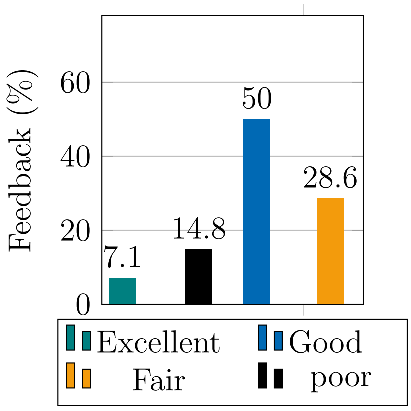


Figure 4.31: Technical Correctness

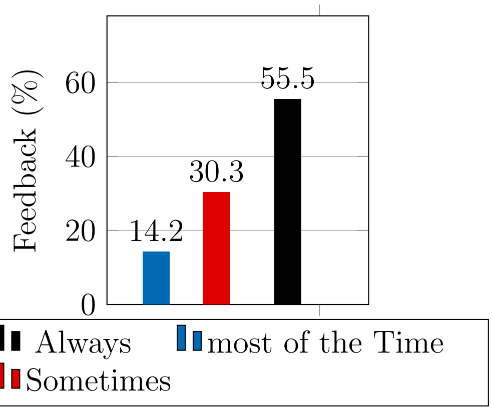


Figure 4.32: Usability

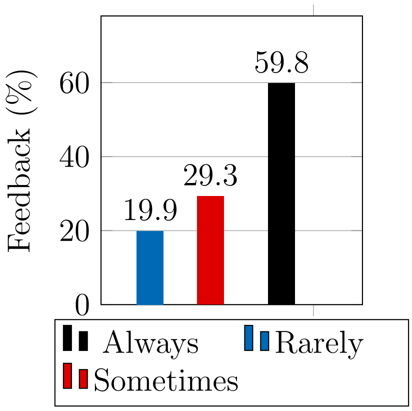


Figure 4.33: Ease of Use

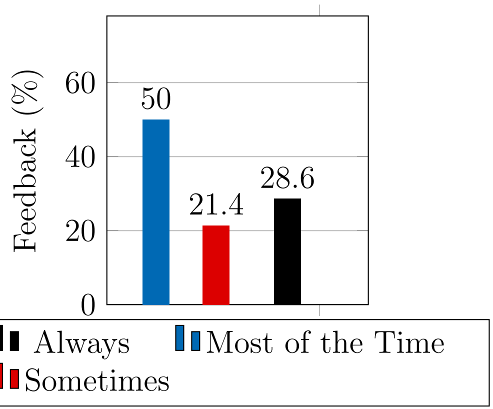


Figure 4.34: System Interactivity

We utilized modern advances in word embeddings, knowledge graphs, and large language models to create conversational agents that are intended to augment human instructors in the future. This is an ongoing project with the potential to transform how students engage with educational content and receive tailored suggestions. The objective is to enhance the learning experience for students of varying proficiency levels by making it more engaging and effective. Another ongoing project is the development of a task generation and evaluation agent. In order for an agent to effectively mediate students' learning engagements, it must possess the ability to evaluate these tasks, which also implies that it must be capable of generating tasks that resemble those of a human. In the next section, I discuss our efforts in this research area.

4.3.11 Task Generation and Evaluation agent

Several studies have been conducted in the past to explore the evaluation of text-to-text generative models with the aim of enhancing learning in educational contexts. In the following, we review related works on question generation and evaluation systems. In rule-based approaches for question answer generation systems, human specialists develop rules or templates to transform a given input text into a set of questions Heilman [2011]. The primary procedure entails preprocessing a given text in order to select specific answers for which questions can be generated based on a specified template. The most difficult aspect of this strategy is the need for a domain-specific human expert. In contrast to conventional approaches that rely heavily on fixed heuristic rules for converting sentences into associated questions, neural question generation models utilize the encoder-decoder architecture and attention mechanism to produce a wide range of meaningful questions from natural language sentences Du et al. [2017]Serban et al. [2016]. This strategy further involves the utilization of various methodologies to integrate the answer information into the generation model. One approach involves utilizing an answer position indicator Liu et al. [2019]Zhou et al. [2018], while another approach involves employing an encoding mechanism for the answers Kim et al. [2019]. Additionally, there exists a method that involves embedding the relative distance between the context words and the answer Sun et al. [2018]. Nevertheless, even when considering context and answer information as input, the issue of question generation remains the problem of mapping one input to multiple outputs. The utilization of large language models for question and answer generation is demonstrated in Sarsa et al. [2022] Radford et al. [2019]. In their study, Radford et al. Radford et al. [2019], demonstrated the capabilities of OpenAI Codex in generating programming exercises, complete with sample solutions and test cases, as well as providing code explanations. The assessment of these outputs was conducted using both qualitative and quantitative methods. The findings of their study indicate that a significant portion of the content generated through automated means exhibits characteristics of novelty and coherence. Furthermore, in certain instances, the generated questions were deemed suitable for immediate utilization without any further modifications. During the process of exercise creation, it was observed that the programming concepts and contextual themes embedded within the exercises can be significantly influenced by the provision of keywords as input to the model. Even though there still needs to be some kind of oversight to make sure that the generated content is educationally sound before it is given to students, the analysis shows that there is a lot of value in using large-scale generative machine learning models to come up with question-and-answer pairs. Compared to the above research efforts, the research direction taken by Radford et al. Radford et al. [2019] is in line with our objective.

4.3.12 Brief Background for Our Strategy

Automated question generation is a natural language processing (NLP) task in which questions are automatically generated based on input data such as text documents, paragraphs, or sentences, using a transformer-based model such as T5 (Text-to-Text Transfer Transformer model) Nguyen et al. [2022]. The T5 model utilizes a sequence-to-sequence approach for learning by taking in a sequence of input text (the context) and producing a sequence of output text (the generated questions) Raffel et al. [2020]. The core idea behind T5 is that it handle every NLP task as a ”text-in, text-out” problem. To gain insight into the T5 language model, we describe the transformer.

Transformer is a neural network architecture that is capable of processing sequential data, including texts, audios, videos, and images, without recurrent or convolutional layers. Its fundamental layer is Attention, and it consists of fully connected, normalization, embedding, and positional encoding layers Guo et al. [2022]; Khan et al. [2022]; Vaswani et al. [2017]. Originally, it was intended for neural machine translation, where a source sentence is first encoded into a fixed-length vector, after which a decoder then outputs a translation from the encoded vector. However, in scenarios where the input text is long, the computational cost of using the fixed-length internal representations for the capture of the semantic details of the input text becomes high. Attention is employed in addressing this challenge by allowing the neural network to focus on the sections of input data that contain meaningful information and pay less attention to the rest of the input. Given an encoder-decoder architecture with attention, several innovative strategies, and billions of parameters, we get large language models that are capable of executing natural language downstream tasks via zero-shot learning.

	Layers	Width	Heads	Parameters
BERT-Large	24	1024	16	340 Million
RoBERTa	24	1024	16	355 Million
Turing-NLG	78	4256	28	17 Billion
LaMDA	64	8192	128	137 Billion
GPT-3	96	12,228	96	175 Billion
PaLM	118	18,432	48	540 Billion

Table 4.23: Large Language Models

In Table 4.23, we list some of the main stream large language models. In the table, the term ”layers” refers to the quantity of encoder-decoder models that are layered on top of each other. The parameter ”width” denotes the dimension of the model. Additionally, the parameter ”heads” represents the number of attention layers in the multi-head attention mechanism. Lastly, the term ”parameters” signifies the total count of parameters utilized by the model.

In an educational setting, these models can compare the answers of each student to the answer key, which is a list of the correct answers. This is done by using semantic textual similarity scoring, which creates a similarity score that shows how similar the student’s answer is to the answer key in terms of meaning and context. A higher similarity score indicates a greater level of correctness with respect to the student’s answer. Automated evaluation with models trained for semantic textual similarity can handle diverse phrasings and synonyms. Therefore, it can recognize a wide range of correct answers, provided they convey the same meaning as the expected answer. While MCQs provide a binary output—correct or incorrect—short-answer questions evaluated with tools like STS-BERT can provide a continuous score based on the degree of semantic similarity to the correct answer. This nuanced scoring allows for a more precise assessment of a student's understanding. In the context of our methodology, in this paper, we focus on short answer questions since many of the studies conducted have proven that short answer exercises can enhance students long-term memory, thereby improving their learning performance.Tsai et al. [2021]. In the next section, we describe our implementation.

4.3.13 Implementation

Our strategy is to utilize a Question-Answer Pair Generation pipeline (using the t5-small model) and an Answer Evaluation pipeline (using the SBERT model), along with a simple Anvil web app user interface, to act as an assessment tool for the user. The user initiates the process by providing a PDF file and certain parameters based on which the question-answer pair is generated. Subsequently, the questions are provided to the user for answering, following which the user can submit his or her response in the form of a short answer(s) for further evaluation. The evaluation of the answers is done by comparing the answer given by the user for each generated question to a previously generated answer to the corresponding question. This comparison is done based on semantic similarity using the answer-evaluation model, and a score is assigned to each answer given by the user. This score is then sent to the user through the UI, where he can see the individual score for each question and the overall grade of his performance. This process can be repeated any number of times with any other PDF document of our choice with either the same or different parametric values.

4.3.14 System Schematic

The system schematic is depicted in Fig 4.35.

The following steps explain the workflow of the whole system.

1. User inputs the required data (PDF and parameters) in the question generation form of the web app and submits through the provided submit button.
2. The user-provided PDF and parameters are sent to the Question-Answer Pair generation pipeline.
3. The generated question-answer pair and the question-answer pair dictionary are sent to the Answer evaluation pipeline.
4. After similarity check, the result is provided to the Question-Answer pair generation pipeline and then stored into the dictionary.
5. Now the generated question list is sent to the answer sheet form so that the user can answer.
6. After the user submits the answer(s), the answer(s) and the generated dictionary are sent to the Answer evaluation pipeline for further processing.
7. Now a score dictionary is generated, which contains question as a key and score as a value. This is sent to the score sheet form so that user can assess how well he performed through the score for each question and the normalized final score.

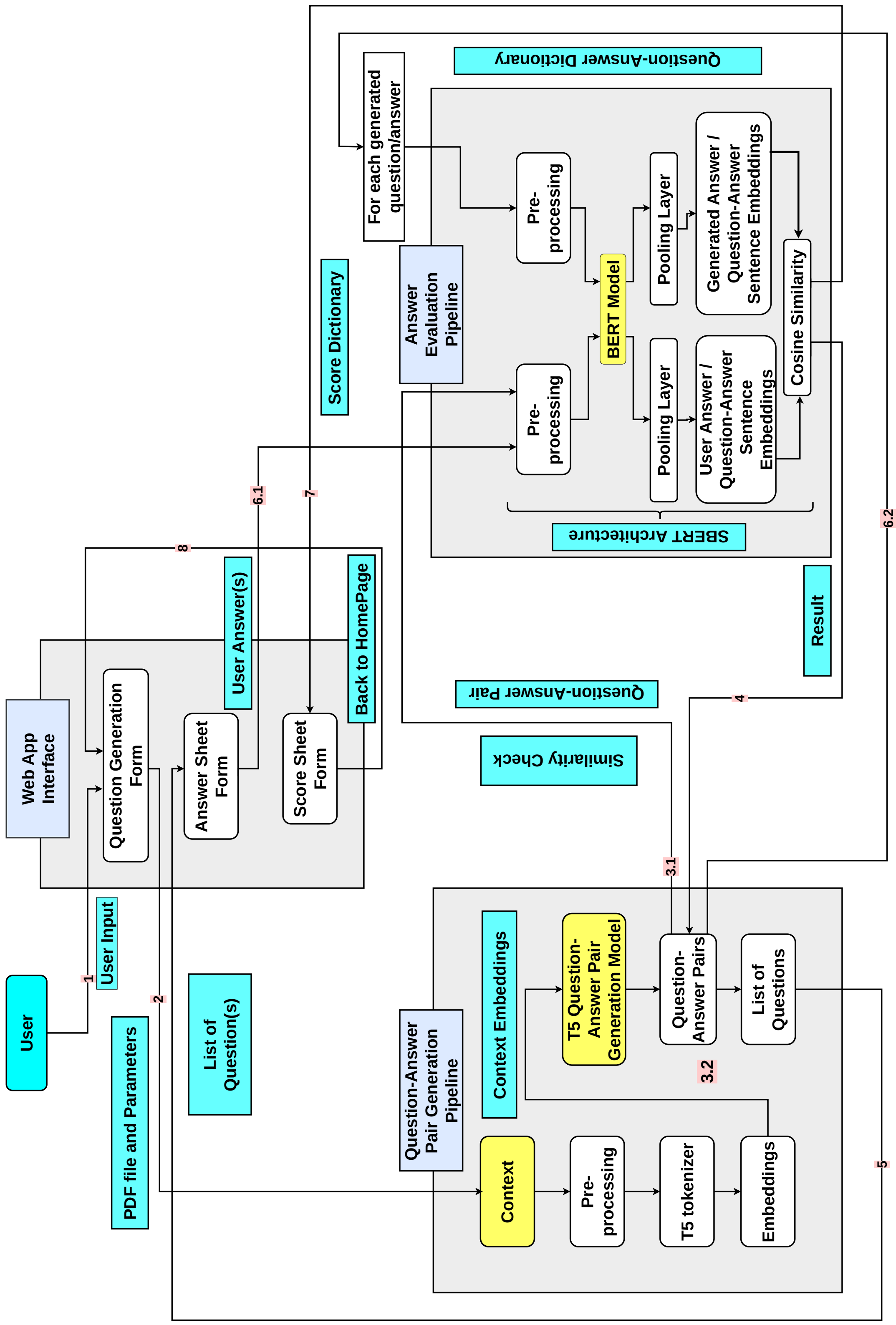


Figure 4.35: System Architecture

8. After viewing the score sheet form, the user can navigate back to the homepage using the Home button provided in the UI.

In the next subsection, we elaborate on the question-answer pair generation pipeline.

4.3.15 Model Architecture

The process of generating question-answer pairs using the T5 model and the evaluation of answers using the SBERT model is illustrated in the Question-Answer Pair Generation Pipeline & Answer Evaluation Pipeline of Fig4.35 respectively. The web app user interface, which facilitates communication between the user and the previously stated models, is depicted in the Web App Interface of Fig4.35.

Question-Answer Pair Generation Pipeline

Using the pre-trained T5 model, we are generating question-answer pairs, and these are saved into a Python dictionary with the question as the key and the corresponding answer(s) as the value. The format of the dictionary is {Question : [Answer]}.

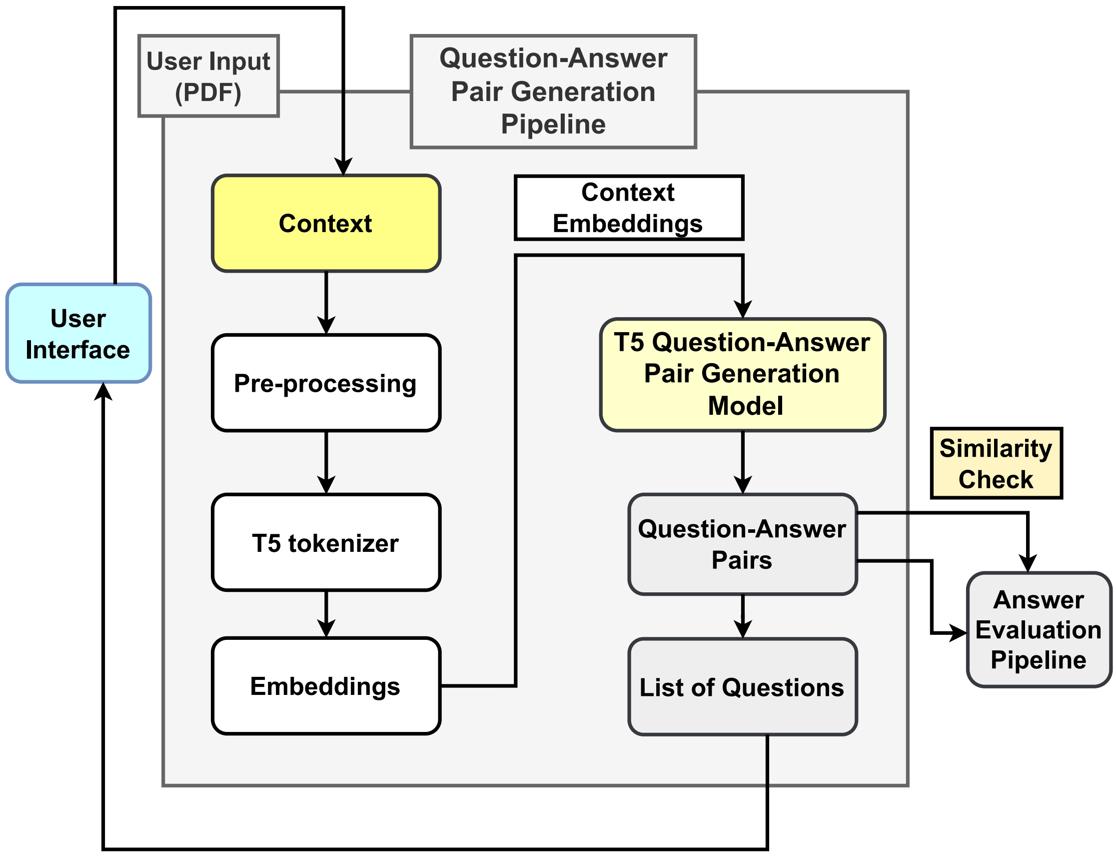


Figure 4.36: Question-Answer Pair Generation Pipeline

The process of question generation starts when the user interacts with the pipeline through the web app interface. The step-by- step description of the process is as follows:

- **User Input:** The user inputs (PDF file and parameters) are provided to the pipeline.
- **Context:** According to the given parameters, the context is extracted from the PDF.

- **Pre-processing:** Standard preprocessing of the extracted context is carried out.
- **T5 tokenizer:** To convert the preprocessed data into a format suitable for processing by the T5 model.
- **Embeddings:** Representing individual words as dense vectors, where each element in the vector represents a feature or property of the word.
- **Model:** We are using a t5-small model, which was then fine-tuned on QuAC, and a generated dataset based on scientific research papers.
- **Question-Answer Pair Generation:** Using the above model, the question-answer pairs are generated, and these are stored in a Python dictionary for easy access.
- **List of questions:** The list of questions is generated using the keys of the dictionary. This list is then provided to the user through the UI, where the user can attempt to answer the questions and then submit them for evaluation.

The QA Generation algorithm extracts questions and answers from PDF, removing figure and table content as a preprocessing step for the PDF document. It generates QA pairs by processing paragraphs on each page while staying within specified QA limits per page. The algorithm then calls our pre-trained model to create new questions and answers, maintaining diversity. It checks for question similarity in the existing dictionary, updating the answers list for similar questions. This process produces a QA dictionary, enabling efficient and diverse QA pair extraction from PDF documents.

During the question-answer generation period, the Answer Evaluation pipeline is utilized in the background to check whether a question that is similar to the currently generated question exists in the aforementioned dictionary. This approach results in the generation of diverse questions and a corresponding possible list of answers for each of the questions generated, which is discussed in detail in the Answer Evaluation pipeline. The process is further illustrated in Algorithm 18.

Details regarding the pipeline components and functionalities are as follows:

- **Transformer model:** T5-small
- **Dataset used for Fine-tuning:** QuAC and Generated Dataset
- **Process:** Question-Answer Pair generation
- **Input:** PDF file, Number of questions to be generated, the start and end page number from which the questions are to be generated.
- **Output:** Question-Answer pair from which we are generating

Answer Evaluation Pipeline

The Answer Evaluation Pipeline Winastwan [2023] is utilized for the following purposes:

- **Generation of diverse/dissimilar questions:** This task is performed during the question-answer pair generation in the Answer Evaluation pipeline. For each

Algorithm 8 QA Generation Algorithm

Require: pdf_document, starting_page, ending_page, num_qa_required, max_qa_per_page

Ensure: Question_Answer_Dictionary

```
Initialize QA_Dictionary = {}
Define figure_regex, table_regex
Calculate num_qa_per_page_ceil = ⌈num_qa_required / (ending_page − starting_page + 1)⌉
if num_qa_per_page_ceil < max_qa_per_page then
    Read pdf_document
    for page in starting_page to ending_page + 1 do
        page = read(page)
        page = apply_regex(page, figure_regex, table_regex)
        paragraphs = gen_paragraphs(page, num_qa_per_page)
        for paragraph in paragraphs do
            question, answer = extract_qa(paragraph)
            if count < num_qa_required then
                generated_ques, generated_ans = QuesAnsModel(t5 − small)
                if len(QA_Dictionary) = 0 then
                    QA_Dictionary[generated_ques] = [generated_ans]
                    count = count + 1
                end if
                if count > 0 then
                    similar_ques = Process_Q(generated_ques, QA_Dictionary.keys())
                    if similar_ques = "" then
                        QA_Dictionary[generated_ques] = [generated_ans]
                        count = count + 1
                    else
                        answer_list = QA_Dictionary[similar_ques]
                        similar_ans = Process_A(generated_ans, answer_list)
                        if similar_ans = False then
                            answer_list.insert(generated_ans)
                            QA_Dictionary[generated_ques] = answer_list
                        end if
                    end if
                end if
            end if
        end for
    end for
    Return QA_Dictionary
end if
```

question and answer pair that gets generated, we invoke the evaluation pipeline to check for similar questions. If a similar question already exists, then the generated answer is compared against the list of answers to the similar question. If a similar answer is found, then the generated answer is discarded, or else the answer is added to the answer list. However, if a similar question is not found, then we consider the generated question-answer pair a new one and add it to the question-answer dictionary.

- **Evaluation:** Once the user submits the answer(s), the evaluation process begins. In this process,the answer(s) given by the user are compared with the answer(s) of the corresponding question in the question-answer dictionary. If we have multiple answers for a generated question, we compare the answer of the user to each answer in the answer list. The answers that are most similar to the given answer are used for evaluation, and based on this, the score is provided accordingly. We work on the assumption that a single question can have multiple possible answers.

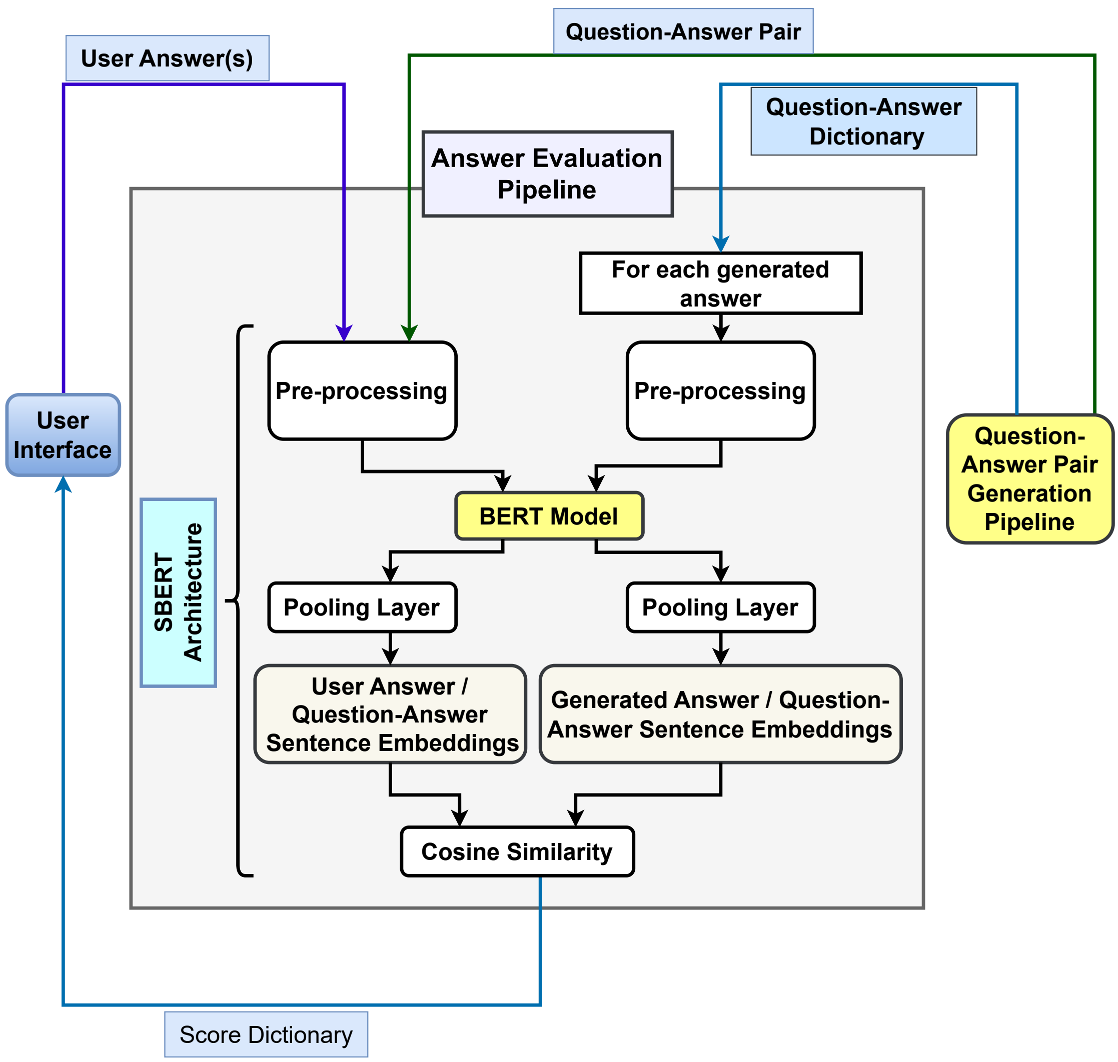


Figure 4.37: Answer Evaluation Pipeline

The Fig 4.35 gives us an outline of the pipeline. As explained above, this pipeline is accessed in two scenarios. For both the scenarios, the steps carried out are same as follows:

- **Input:** The input can either be user answer(s) or question-answer pair from Question-Answer Pair Generation Pipeline and the Question-Answer Dictionary
- **Pre-processing:** Standard pre-processing of the extracted context is carried out.
- **BERT Model:** Here the model carries out STS (Semantic Textual Similarity) wherein it compares the similarity between one text to another using the cosine similarity measure. The model was trained on the STSB and SICK datasets.
- **Pooling layer:** This layer is used to generate sentence/text embeddings instead of token-level embeddings.
- **Sentence Embeddings:** The generated embeddings can then be compared to each other with the help of cosine similarity, which thereby achieves the STS task.

All the steps mentioned above, starting from the pre-processing step to the final cosine similarity check, comprise the SBERT's model architecture. The process is further illustrated in Algorithm 29.

Algorithm 9 Process_A : Answer comparison algorithm

Require: answer, answer_list

Initialize $similar\_answer$ =**False**, $compare\_answers = []$
for $value$ **in** $answer\_list$ **do**
  if $similar\_answer =$ **False then**
    $compare\_answers$.append($answer$)
    $compare\_answers$.append($value$)
    if $AnswerEvalModel(compare\_answers > 0.8$ **then**
      $similar\_answer \leftarrow$ **True**
    end if
    $compare\_answers$.clear()
  end if
end for
return $similar\_answer$

Details regarding the pipeline components and functionalities are as follows:

- **Transformer model:** SBERT
- **Dataset used for Fine-tuning:** STSB-multi-mt and SICK
- **Process:** Similarity check (Cosine)
- **Input**
  - **Generation of dissimilar question:** Question-Answer pair.

  - **Evaluation:** The answer given by the user and the list of generated answers corresponding to the question answered by the user.

- **Output**
  - **Generation of dissimilar question:** Dissimilar question and answer list.
  - **Evaluation:** Score for answers given by the user.

Web App User Interface

In order to make our system user-friendly, we have created a simple Python-based web app user interface (using Anvil Web App Builder) to ensure a seamless user experience.

Initially, we established the connection between the Anvil webapp server and Google Colab through server link using our server key. We also used @anvil.server.callable decorator to sent and receive data between Colab and the front-end.

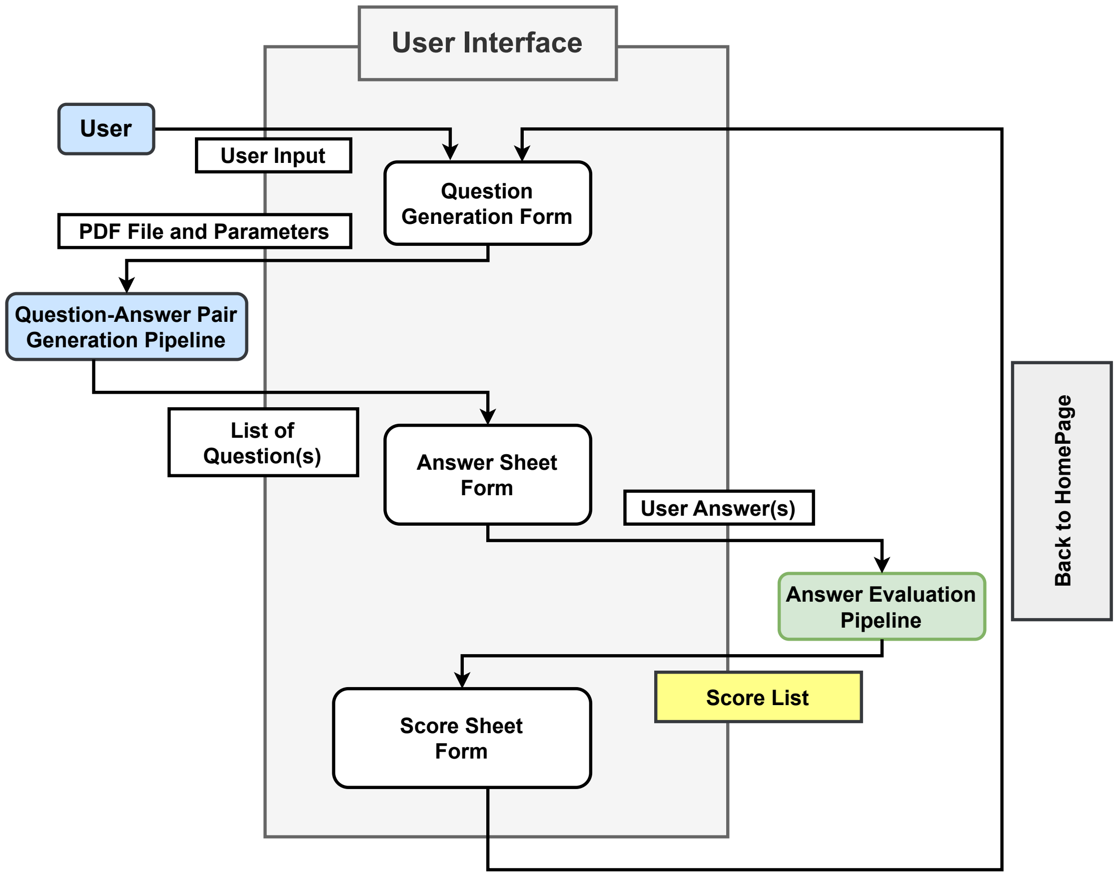


Figure 4.38: Web App User Interface

- **Question Generation Form:** The initial interaction between the user and the system is through the Web App Interface, Figure 4.38.

  The user can upload a PDF file of their choice, along with which they can provide additional parameters such as the start and end page numbers as well as the

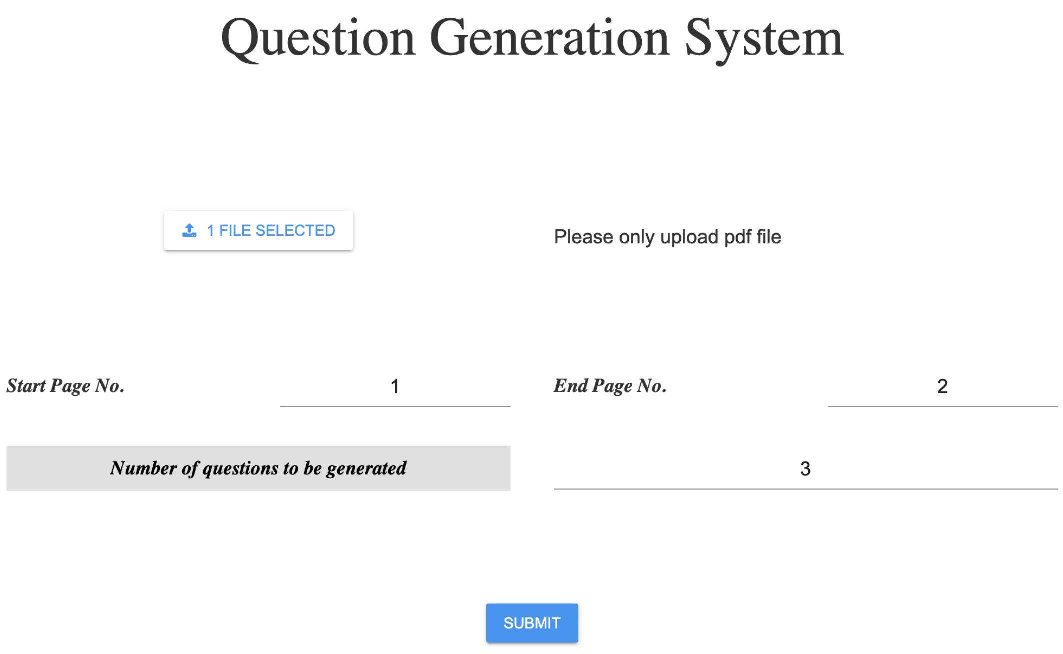


Figure 4.39: Question Generation Form

number of questions to be generated. Upon providing these user inputs, a set of questions is generated from the specified pages.

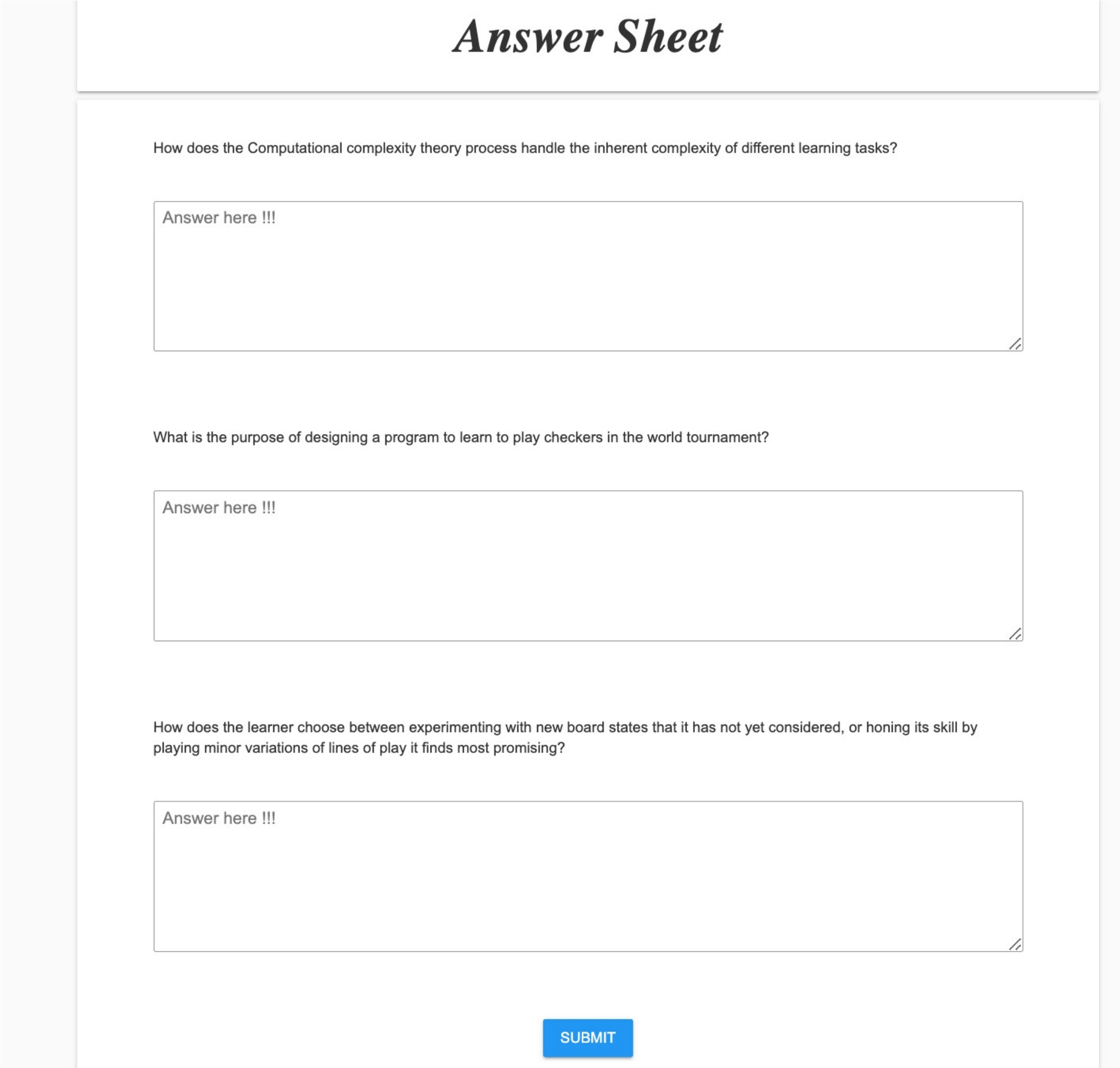
- **Answer Sheet Form:** Once the questions are generated, the user is navigated to the second level of the interface to view and answer the list of generated questions, as further shown in 4.35.

  Each question has a separate input field for the user to type in their answer. After answering all the questions, the user can submit their answers to be evaluated.

- **Score Sheet Form:** In the end, the user is provided with an immediate feedback in the form of the evaluation score in Score Sheet Form 4.35.

  Each question's user given answer is scored individually by computing the similarity score with respect to the corresponding answer(s) available in the system generated question-answer dictionary. Based on a certain threshold the scores are assigned to the user between 0 and 1. The similarity score below threshold returns score 0 and above threshold returns score as per similarity value. Additionally, the final score is normalized to a value between 0 to 1.

After the score is presented to the user, we again navigate back to the homepage, i.e., the Question Generation Form, so that the user can continue with the assessment generation of new topics or literature.

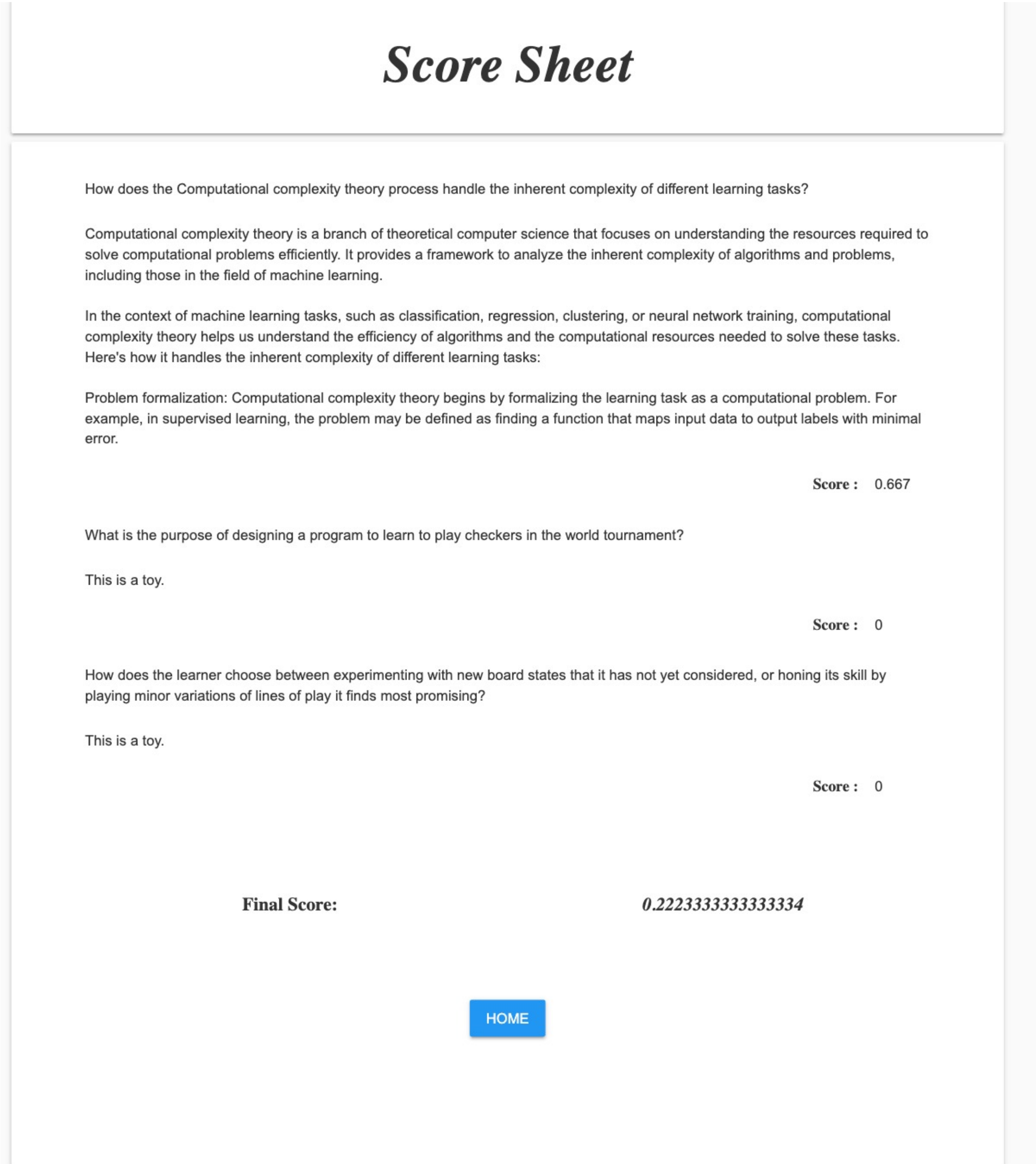


4.3.16 Evaluation and Result

In this section, we provide the datasets used and the evaluation metrics, along with the results of the performance evaluation for both the Question-Answer Pair Generation task and the Answer Evaluation task.

4.3.17 Dataset Used

The proposed Question-Answer Pair Generation model has been trained on QUAC Choi et al. [2018] and a custom-generated dataset. The motivation to use these datasets was to focus on abstractive question answers and scientific text. As there were no specific datasets available for scientific text, we created a custom-generated dataset. This dataset was manually generated by us and consists of context and question-answer pairs created from the relevant research papers gathered during the literature survey. All these papers are mentioned in the references.



The proposed answer evaluation model has been trained on the STSB and SICK datasets. These data sets cover a wide range of domains and types of texts, which are well-established sources for evaluating the performance of STS models.

4.3.18 Evaluation Metrics

The performance of the question-answer pair generation model is assessed using the metrics listed below.

- BLEU Papineni et al. [2002] measures the precision that scores word similarity between candidate and reference sentence.

Input Context to the Model
We present QuAC, a dataset for Question Answering in Context that contains 14K information-seeking QA dialogs (100K questions in total). The dialogs involve two crowd workers: (1) a student who poses a sequence of freeform questions to learn as much as possible about a hidden Wikipedia text, and (2) a teacher who answers the questions by providing short excerpts from the text. QuAC introduces challenges not found in existing machine comprehension datasets: its questions are often more open-ended, unanswerable, or only meaningful within the dialog context, as we show in a detailed qualitative evaluation. We also report results for a number of reference models, including a recently state-of-the-art reading comprehension architecture extended to model dialog context. Our best model underperforms humans by 20 F1, suggesting that there is significant room for future work on this data. Dataset, baseline, and leaderboard available at http://quac.ai
Our core evaluation metric, word-level F1, is implemented similarly to SQuAD : precision and recall are computed by considering the portion of words in the prediction and references that overlap after removing stopwords.12 For no answer questions, we give the system an F1 of one if it correctly predicts no answer and zero otherwise.13 Like SQuAD, we compute the maximum F1 among all references; however, since many questions have multiple valid answers, this metric varies significantly with the number of reference annotations. To make oracle human and system performance comparable, given n references, we report the average of the maximum F1 computed from each n 1 subset with respect to the heldout reference. Additionally, since averaged F1 can be misleading for questions with multiple valid answers, we introduce the human equivalence score (HEQ), a performance measure for judging whether a system's output is as good as that of an average human.14 HEQ measures the percentage of examples for which system F1 exceeds or matches human F1. We compute two variants: (1) the percentage of questions for which this is true (HEQ-Q), and (2) the percentage of dialogs for which this is true for every question in the dialog (HEQ-D). A system that achieves a value of 100 on HEQ-D can by definition maintain average human quality output over full dialogs. For dialog acts, we report accuracy with respect to the majority annotation, breaking ties randomly.
Sanity check Overall, the poor sanity check results imply that is very challenging. Of these, following the transition matrix (TM) gives the best performance, reinforcing the observation that the dialog context plays a significant role in the task. Upper bounds The human upper bound (80.8 F1) demonstrates high agreement. While Gold sentence + NA does perform well, indicating that significant progress can be made by treating the problem as answer sentence selection, HEQ measures show that span-based approaches will be needed achieve average human equivalence. Finally, the Gold NA + TM shows that cannot be solved by ignoring question and answer text Baselines Text similarity methods such as bagof-ngrams overlap and InferSent are largely ineffective on , which shows that questions have little direct overlap with their answers. On the other hand, BiDAF++ models make significant progress, demonstrating that existing models can already capture a significant portion of phenomena in . The addition of information from previous turns (w/ 1-ctx) helps significantly, indicating that integration of context is essential to solving the task. While increasing the context size in BiDAF++ continues to help, we observe saturation using contexts of length 3, suggesting that more sophisticated models are necessary to take full advantage of the context. Finally, even our best model underperforms humans: the system achieves human equivalence on only 60% of questions and 5% of full dialogs.

- ROUGE-1 Lin [2004] refers to the overlap of unigrams (each word) between the system and reference summaries.
- METEOR Lavie and Agarwal [2005] is based on the harmonic mean of recall and precision, with recall weighted higher than precision.
- MSE-Loss Function measures the average squared difference between the predicted similarity scores and the ground truth similarity scores.

4.3.19 Evaluation Results

In our assessment of the Question-Answer Pair Generation task, we embarked on a comprehensive comparison involving our model and the GPT 3.5 Turbo variant. These models underwent meticulous evaluation against human-crafted answers, utilizing well-established word-overlap metrics. Subsequently, we closely scrutinized and compared the performance of both models.

Our evaluation process entailed subjecting the two models to a set of five selected questions, drawn from a specific contextual domain. The resulting insights from Fig4.40 reveal a distinct pattern. Except for questions 1 and 5, GPT consistently outperforms our model. This performance disparity is notably attributed to the sheer parameter magnitude of GPT, boasting 175 billion parameters compared to our model's configuration with 64 million parameters in the T5-small setup.

Notably, in terms of precision as discerned through ROUGE scores in Fig4.42, our model stands out as a superior contender. It excels in three out of the five instances

Sample Questions & Answers Generated from the Models	
Proposed Model	**GPT-3.5 Turbo**
Question 1: What are the challenges presented in the QuAC dataset? Answer: The challenges encountered in existing machine comprehension datasets include open-ended, unanswerable, or meaningful within the dialog context, as shown in qualitative evaluation.	Question 1: What challenges does the QuAC dataset introduce that are not present in existing machine comprehension datasets? Answer: The QuAC dataset introduces challenges such as open-ended, unanswerable, or context-dependent questions, which are not commonly found in existing machine comprehension datasets. These questions may only be meaningful within the specific context of the dialog. These challenges were identified through a detailed qualitative evaluation of the dataset.
Question 2: How does the human equivalence score (HEQ) measure the performance of a system's output as good as that of an average human? Answer: For questions with multiple valid answers, the metric varies significantly with the number of reference annotations. To make oracle human and system performance comparable, given n references, we report the average of the maximum F1 computed from each n 1 subset with respect to the heldout reference.	Question 2: How is the word-level F1 evaluation metric implemented in comparison to SQuAD, and how is it adjusted for questions without answers? Answer: The word-level F1 evaluation metric is implemented similarly to SQuAD, measuring precision and recall by considering overlapping words between predictions and references, excluding stopwords. For questions without answers, a system receives an F1 score of one if it predicts no answer correctly. The maximum F1 is computed like SQuAD, but its variation increases due to multiple valid answers. Human and system performance comparability is achieved by averaging maximum F1 values from subsets of references. The human equivalence score (HEQ) assesses if a system's output matches or surpasses human performance, with two variants: HEQ-Q for questions and HEQ-D for dialogs. An HEQ-D value of 100 indicates maintaining human-quality output over full dialogs. For dialog acts, accuracy is reported based on the majority annotation, resolving ties randomly.
Question 3: What does the poor results obtained from the sanity check imply about the QuAC task? Answer: The lack of results from previous turns (w/ 1-ctx) helps significantly, indicating that integration of context is essential to solving the task. While increasing the context size in BiDAF++ continues to help, we observe saturation using contexts of length 3, suggesting that more sophisticated models are necessary to take full advantage of the context.	Question 3: What insights do the sanity check results, upper bounds, and baseline experiments provide about the challenges and performance of different approaches in the context of the task? Answer : The results of the sanity check suggest that achieving satisfactory performance in the task is challenging. Transition matrix (TM) approach performs best, highlighting the importance of dialog context. Human upper bound shows high agreement (80.8 F1). While Gold sentence + NA is effective, HEQ measures indicate span-based methods are necessary for average human equivalence. Gold NA + TM underscores the importance of considering question and answer text. Baseline experiments show text similarity methods are ineffective; BiDAF++ models make significant progress, context integration is vital. Best model underperforms humans, achieving human equivalence on 60

Table 4.24: Sample tasks without reflection script

evaluated, showcasing its prowess in generating precise and accurate responses. Conversely, GPT consistently achieves remarkable recall across all instances, signifying its strength in comprehensive context recall. And the F1 score depicted in Fig4.41, except for one instance GPT is outperforming our model.

Delving deeper into the results, Fig4.43, provides additional insights into our model's proficiency. The graph illustrates our model's commendable performance in 60% of the cases analyzed.

We have used MSE Loss to quantify the discrepancy between the predicted similarity scores and the ground truth similarity scores in the answer evaluation task. This measure has been considered for evaluation as it is particularly well suited to continuous-valued similarity scores since it penalizes larger differences between predicted and actual scores more severely. Fig4.44 presents the MSE Loss values

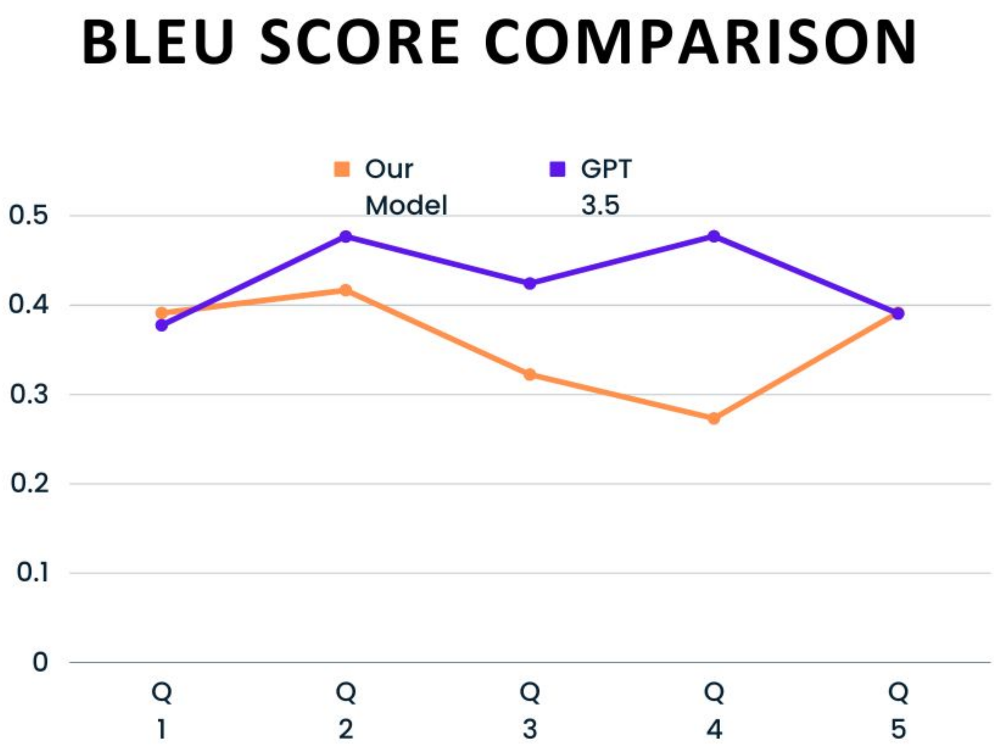


Figure 4.40: BLUE Score comparison

obtained for our model after each successive epoch, on the concatenated dataset comprising both STSB and SICK datasets. The low MSE Loss values indicate the model's effectiveness in capturing semantic textual similarity, thereby showcasing its capability of accurately measuring semantic relationships between sentences.

4.3.20 Task Generation and Evaluation Agent Summary

In this ongoing effort, we described our ongoing efforts at generating and evaluating tasks using the Text-to-Text Transfer Transformer (T5) and SBERT. Our system provides a robust and scalable solution for efficient question generation, evaluation, and feedback. The combination of SBERT architecture and T5 proved beneficial in the automation of assessments, which is scalable and effective for educational settings. Our approach significantly reduces manual effort to generate and evaluate questions, providing students with objective and timely feedback on their performance. Additionally, the integration of T5 and SBERT allows for a more personalized learning experience, as the system can adapt to individual student needs and preferences. Our future work consists of optimizing the model performance using more extensive context-specific datasets as well as incorporating larger variants of the t5 model. Currently, the model works in a context-specific manner, adhering to one particular

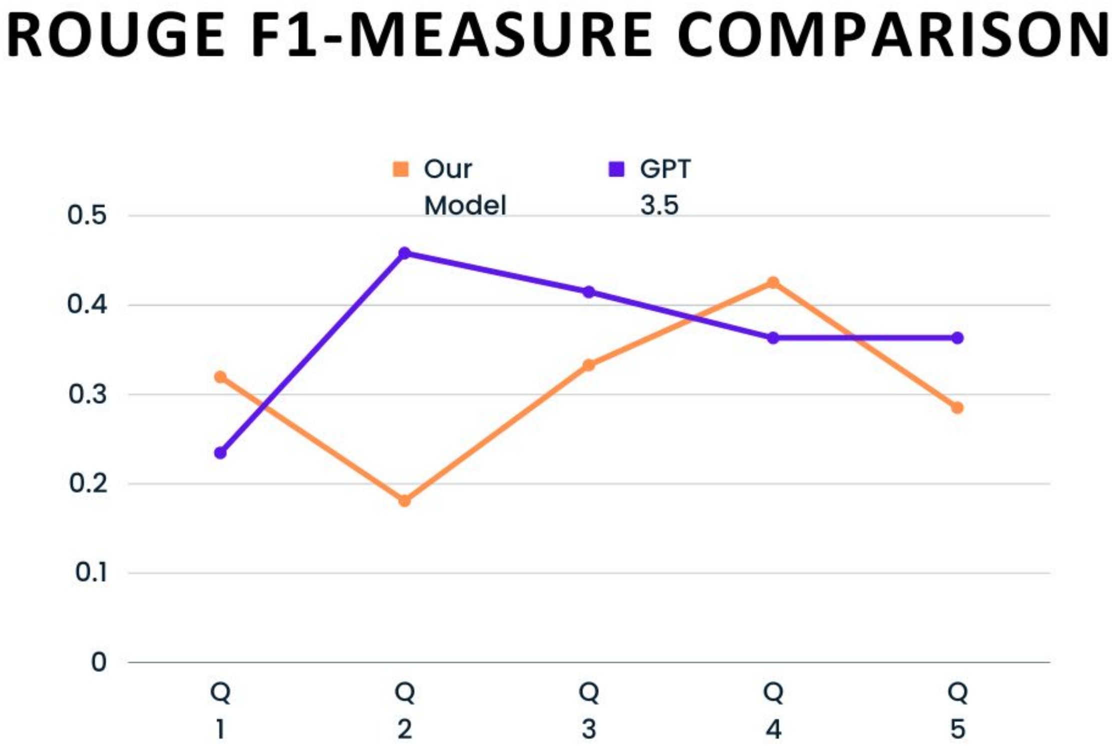


Figure 4.41: ROUGE F1-Measure Comparison

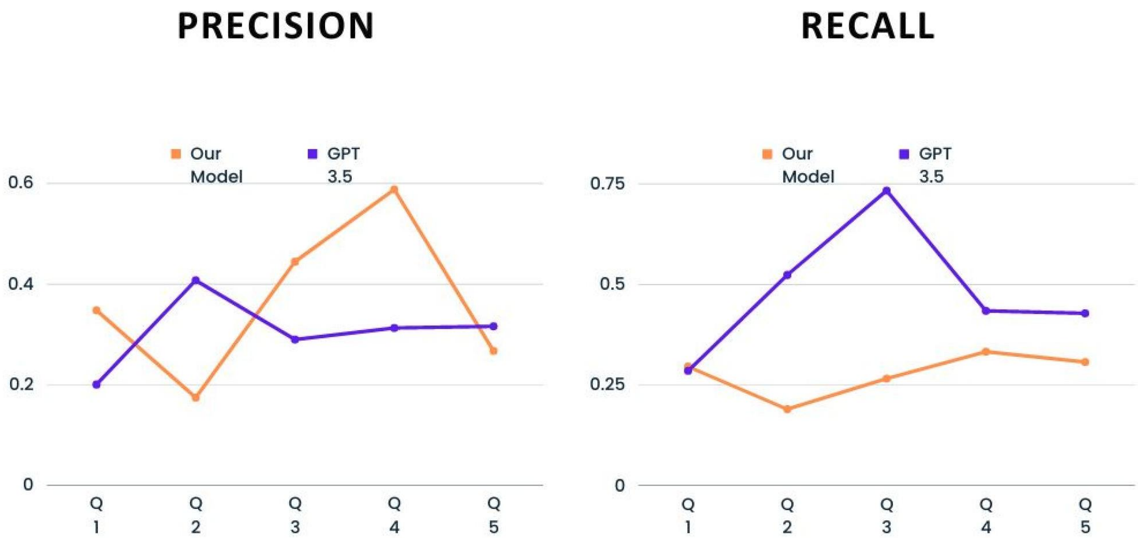


Figure 4.42: Precision-Recall ROUGE score comparison

domain (scientific text); however, this can be extended to various other domains (for example, SQL) by making use of transfer learning.

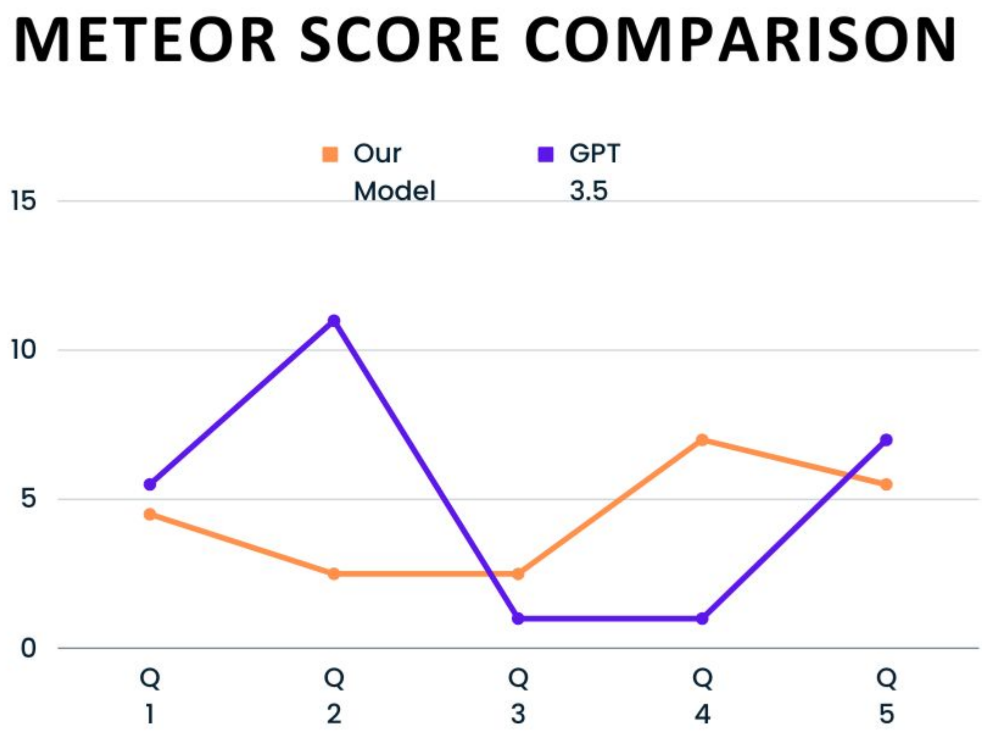


Figure 4.43: METEOR Score Comparison

4.3.21 Summary

Learning is crucial for acquiring knowledge, but traditional teaching methods often lack incentives to cultivate practical knowledge and instead encourage rote memorization rather than comprehension. Recently, numerous instructional techniques have been implemented to convey crucial lessons and enhance comprehension of a subject. Real-time teacher feedback is essential for learners to acquire information and skills. Yet, offering immediate feedback tailored to each individual is frequently impractical due to constraints on instructional resources.

This particularly applies to students encountering difficulties in solving lecture-related tasks, as was the case in our structured query language (SQL) lectures and exercises. We have developed a system that can automatically match relevant lecture slides to SQL jobs, providing students needing extra help with exercise recommendations based on specific lecture pages. We began our work on the recommendation system by turning the PDF containing the course slides into a string for analysis. SQL keywords on each page are identified and evaluated depending on their effectiveness in differentiating one page from another. Next, we needed to analyze the SQL exercises in order to compare them with the course materials. The SQL exercises are retrieved from a database and examined in a similar manner as the lecture slides.

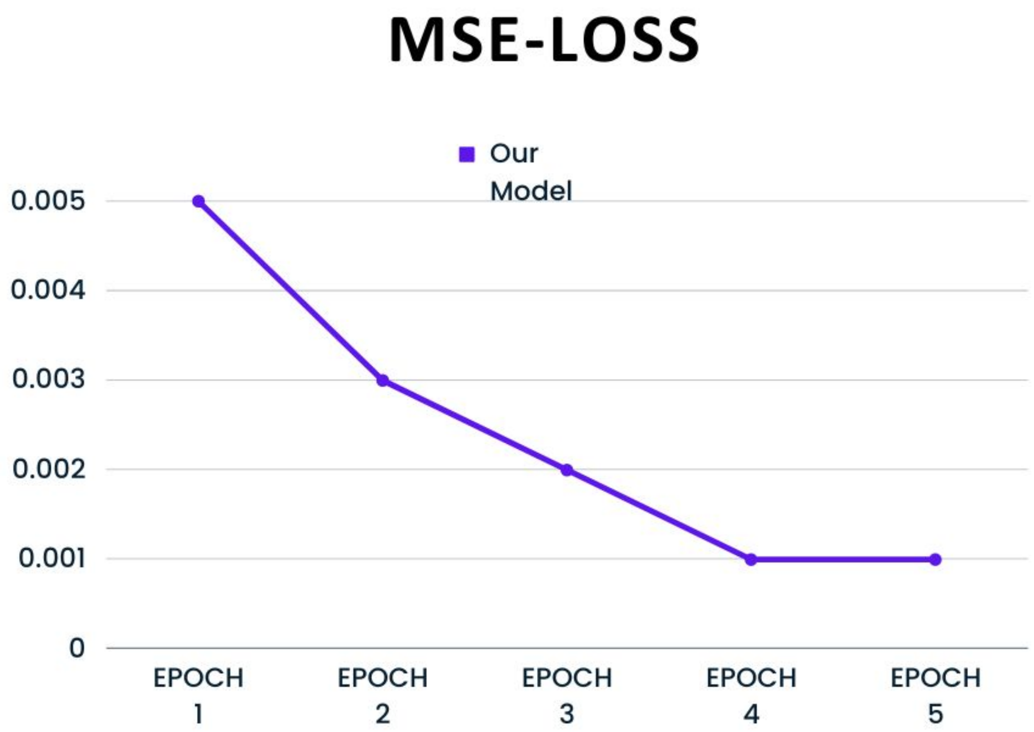


Figure 4.44: MSE-LOSS Curve

Our system calculates the cosine similarity for each combination of exercises and slides after deriving the tf*idf values for both. The similarity values are compared, and the pages with the highest cosine values are chosen to be recommended to pupils. Several optimization strategies have enhanced this procedure. Instructors can specify a minimum cosine value that must be achieved for slides to be recommended to students using our recommendation system. The recommendation can be enhanced by including slides that correspond to a specific fraction of the highest cosine values. A correlation has been identified in our system indicating that linking SQL exercises with lecture slides, in conjunction with cosine similarity, can generate practical recommendations.

5. Summary Recommendation

Here, we provide a concise overview of the chapters of this dissertation. Furthermore, we highlight the key contributions related to each of our study aims. We will provide guidelines for future study that expand upon our contributions.

5.1 Summary of Study

Chapter1

In chapter 1, we detail the contributions of this thesis, and an overview of the respective challenges, objectives, and strategies taken to address them is given, as they will be well elaborated in the respective chapters. For each of these contributions, a comprehensive literature review on the topic, which provides a solid foundation for further research in the field, was carried out. In the coming chapters, we will also offer deductive insights into the subject matter. The contributions of this thesis are spread into four broad areas: the learning analytic research area, the collaboration mediation area, the recommendation system area, and the instructional feedback area. Furthermore, the thesis has two main chapters. One is focused on team collaboration and partner recommendation research, and the other will be focused on the automatic instructional feedback research area. The concepts and methodologies employed in collaboration and part-recommendation research are similar and thus will be discussed in this chapter. The research contribution on the learning analytic research is located in the index section, where it highlights the key findings and recommendations for further study.

Chapter2

In chapter 2, we introduced the learning interaction hierarchy, which affords a method of characterizing and modeling forms for learning engagements. This hierarchy allows us to understand the different levels of interactions that take place during individual learning and team interaction scenarios, which arise during course projects that require teams of students to work together. By categorizing and modeling these forms of engagement, we can gain insights into how learners interact with the course content, instructors, and their peers. This understanding can help in designing effective learning experiences and improving educational outcomes. In this thesis, we contribute a strategy for administering team collaboration, a platform that facilitates it, and a strategy to mediate between two collaboration systems.

Chapter3

In chapter 3, we studied collaboration, which is a vital component of university education as students unite to pursue shared goals, such as acquiring knowledge in certain subjects or engaging in team projects and group assignments. Collaboration not only fosters academic growth but also helps students develop important skills such as communication, problem-solving, and teamwork. Additionally, successful collaborations can lead to long-lasting professional relationships and networking opportunities that can benefit students in their future careers. Conversely, a failed collaboration results in a failure to achieve these goals and is specifically regarded as a bad encounter, perhaps influencing their future partnerships. In this research area, we contribute an approach that leverages multiplex partitioning to create and recommend collaboration teams of desired sizes. Furthermore, we devised a strategy for recommending collaboration team members, mediating teams, and designing a collaborative task for students in an introductory database course.

Chapter4

In chapter 4, we described our contribution towards the integration of instructional feedback during structured query language learning sessions. Our approach focused on providing real-time feedback to learners as they practiced writing SQL queries. This feedback was tailored to address common errors and misconceptions, helping students improve their query-writing skills more effectively. The primary objective of instructional feedback is to furnish learners with comprehensive information on their knowledge or performance, enabling them to make pertinent enhancements and adjustments. Additionally, it can motivate and encourage students to persevere in their efforts. Timely instructional feedback is normally provided soon after the submission or completion of a task. Immediately following the feedback, learners can compare their recent experiences and actions. We also described our learning analytic strategy and our current efforts in the use of conversational agents for the integration of human-like feedback into students learning sessions.

5.2 Contributions

Learning Analytic Area

Our key contribution in this research area is the conceptualization of a learning intervention technique, a tutorial walk-through, that familiarizes students with our learning management system and how to resolve prevalent errors, such as syntax errors, while engaging with their respective exercise tasks. Next, we devised the error class strategy, which affords instructors valuable insights about students' learning progress in courses focused on the structured query language. We further contributed a learning analytic dashboard that leverages the error classes to generate a retrospective evaluation of students social engagements.

Collaborative Learning Area

Our key contribution to this research area is an improved understanding of how teams collaborate, a platform that supports teamwork, and a strategy for recommending collaboration project partners by taking account of their personality dispositions and the learner network interaction hierarchy, which characterizes the various interaction modeling forms in learner-centered social networks. This aspect of this research provides valuable insights for organizations looking to enhance their team dynamics and optimize their SQL project management processes. The platform for team collaboration provides a user-friendly interface that facilitates seamless communication and coordination between team members, allowing them to work together efficiently on SQL tasks. Additionally, we collected data on the performance and satisfaction of participants to evaluate the effectiveness of the collaborative approach and identify areas for improvement in future iterations of the platform. Insight suggests that task reflection and repeated interaction are essential for skill acquisition. Also, collaboration facilitates skill acquisition and knowledge transfer. Furthermore, insights indicate that human behavior and personality are inextricably linked, and a person's interests and tastes are frequently the results of their personality. Lastly, individuals with similar personalities are likely to share interests and behaviors.

Automatic Instructional Feedback

Our most important contribution to the field of automatic instructional feedback is a way to give meaningful feedback during individual online exercise sessions. This is done by taking advantage of the similarities between theory and practice tasks that use structured query language (SQL) and analyzing SQL keywords. We also contribute a strategy for using conversational agents as an agency for the provision of personalized instructional feedback. Further insight suggests that immediate instructional feedback is essential for knowledge transfer, stress reduction, and increasing informal learning. Also, in scenarios where blogs are used, rating comments or posts based on interactions and correctness aids in reducing misconceptions and further increases knowledge creation. Additional insights in this research area indicate that active learning can be increasingly stimulated via informal communications and platforms that provide concise descriptions of the information needed to solve exercises and potentially generate lots of engagements.

In general, the utilization of learning analytic methodologies can effectively tackle various difficulties pertaining to learning. Thus, in this dissertation, we aimed to understand student engagement and the potential factors contributing to their academic struggles. Such an understanding is crucial for developing targeted interventions and support systems to improve student outcomes. By analyzing data on student behavior and performance, we were able to identify patterns and trends that can inform educational practices and policies. We argue that we achieved the set goal by contributing, in a nutshell:

1. A retrospective learning evaluation strategy that can be used for different courses.

2. insights into how to create collaborative environments and recommend effective teams

3. insights into how to create and offer automatic study recommendations

4. Strategies for conceptualizing and implementing learning management systems and tools that improve students learning experiences.

5.3 Future Work

As can be observed lately, the rate at which asynchronous distance learning platforms are used has greatly increased, and recent innovations in the natural language processing research area have led to the gradual but increasing adoption of automated systems in traditional contexts. According to Wang et al. Wang et al. [2023], automated question generation and answer evaluation systems (AQGAES) enhance the learning experience by automatically generating relevant questions and delivering timely feedback, which improves the students' learning interaction and knowledge acquisition and alleviates the burden of creating questions and evaluating assessments manually. And the major innovation at the center of this automation is the transformer.

Also, current research trends indicate that transformer-based pre-trained models can accomplish state-of-the-art performance on a variety of tasks, including speech processing and machine translation. Qiu et al. [2020]. Thus, a prospective path is the conceptualization and development of an intelligent agent-mediated learning platform based on current large language models. Furthermore, given the significant advantages of automated assessments, including scalability and consistency in evaluating student responses, large-language NLP models can rapidly evaluate a large number of responses while consistently applying the evaluation criteria to each response. This scalability ensures that assessments can be conducted efficiently, even in cases where the number of students or assessments is high. This is potentially a challenging as well as beneficial research direction, i.e., the conceptualization of an assessment model that takes into account the provenance of the submitted tasks.

Appendix

First Questionnaire

- Demographic (1) What is you Gender ?
- Demographic (2) How old are you?
- In which semester are you currently studying? Terms of studying Explanation: Terms of studying are all semesters spent on your major study.
- What is your current study?
- How would you rate your general theoretical knowledge of programming ?
- How would you rate your overall practical programming skills?
- For how many years have you been programming practically?
- How would you assess your general knowledge with SQL ?
- The following questions are about your attitudes towards the course Datenmanagement/Database Concepts.
  How would you classify yourself with regard to the following statements?
  If I study in appropriate ways, then I will be able to understand the whole content of the course.
  I'm certain I can understand even the most difficult content in this course.
  It is important for me to understand the content of the course.
  I am very interested in the content area of this course.
  If I try hard enough, then I will understand the content.
  I expect to do well in this course.
  I think the course material is useful for me to learn.
  If I don't understand the content of the course, it is because I didn't try hard enough.
  I'm certain I can master the skills being taught in this course.

- How would you rate your own experience with group work?
- Please rate the following statements about teamwork I think teamwork during study is important, because ...
- I like to work with other students in group activities. it is fun.
  comparing with doing individual assignments, it is more effective to learn by doing group work.
  I will need teamwork skills in my future job.

working in groups allows me to tackle more complex topics than working individually.
there are many opportunities for discussion and sharing ideas by working in groups.
I believe I can do well in the group work.
I believe I can support groupmates.
I believe I can work well with groupmates.
I believe I can play an important role in the accomplishment of the group task.

- To what extent do the following statements apply to you? I like to share my ideas with others.
  I am open to new ideas.
  I am tolerant of different ideas.
  I am able to express what I think in an appropriate way, not harming other group members.
  I always participate in an appropriate way.
  I am able to provide feedback on overall team's performance.
  I am able to provide feedback on individual team member's performance.
  I am able to monitor my group's progress.
  I am able to implement an appropriate conflict resolution strategy.
  I am able to recognize the source of conflict confronting my group.

- How do you evaluate online teamwork during Covid19?
  I think locally distributed online collaboration works at least as well as locally non-distributed collaboration.
  I am comfortable about communicating with group members electronically.

- Last but not least, here is a short . prior knowledge test

  Imagine you want to do a small project in Git together with two other programmers, What do you think are the conditions for good teamwork? Please be as concise

  What needs to be considered before or during the collaboration? What steps are necessary for this? Please be as concise as possible

Final Exam Questionnaire

- First, please rate the course "Database concepts" based on a few questions.
- To what extent are you satisfied with the course ?
- How well does the course helps you to improve your current understanding of SQL LANGUAGE?
- How well do the exercises help you understand the course content?

- How well do you feel prepared for the exam at the end of the semester?
- Please rate the collaboration in your group during the practical semester project in SQLvalidator by choosing a value for each question on a scale from 1 – 10
  1 How was your group's performance during the practical project from 1 (= very poor) to 10 (= very well)?
  2 How was your individual performance during the practical project from 1 (= very poor) to 10 (= very well)?
  3 How do you currently rate your own teamwork skills from 1 (= very poor) to 10 (= very well)?
  4 How satisfied are you with the cooperation in your group while working on the practical project from 1 (= not satisfied at all) to 10 (very satisfied)?
  5 How confident are you that your team would perform a future task successfully from 1 (= not confident at all) to 10 (= very confident)?
  6 How confident are you that your team would work well together on future tasks from 1 (= not confident at all) to 10 (= very confident)?
  3 The following items are about teamwork in your group during the whole course.
- To what extent do you agree with the following statements about the collaboration in your group?
  My team carried out an effective management and organization process.
  The organization has encouraged group members to take responsibility for their work within the team.
  The interaction process among group members has favored the development of teamwork skills.
  My work group members have given me support, help and encouragement at times when it was necessary.
  Teamwork has contributed to making me feel more involved in studying the subject.
  Having contact with the team has helped me carry out the academic tasks of the course.
  Collaborative learning has helped me to further develop my knowledge.
  Teamwork has allowed me to complement my knowledge with that of my colleagues.
  I have learned more interacting with my teammates than when I work alone.
  Interacting with my teammates, I improve the grades I would have obtained working individually on the task.
  The time allocated to organizing the group work is compensated by the learning that I have acquired.
- To what extent do you agree with the following statements about cohesion in your group?
  Team members understood group goals and were committed to them.
  Team members were friendly and interested in each other.
  Team members openly addressed problems within the team.
  Team members listened with understanding to each other.
  Team members included each other in the decision-making process.
  Team members recognized and respected individual differences in the team.

Team members contributed ideas and solutions to problems.
Team members valued the contributions and ideas of each other.
Team members recognized and rewarded team performance.
Team members encouraged and appreciated comments about team efforts.

- To what extent do you agree with the following statements ? During our collaborative work in the current course I improved my skills about ...

- How much did your teamwork skills improved in the course?
- How would you rate your own familiarity with group work?
- Did you have any problems in teamwork during the course where you would have liked to get help from instructors?

Extrovert or Introvert
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- Enter your Group and Role

- Do you prefer to work alone, or as part of a team?

- How much do you enjoy social gatherings?

- What is your ideal way of celebrating your birthday?

- Are you more comfortable when talking to people on a one-to-one basis or in a group discussion?

- How quickly do you become bored and restless when performing routine tasks?

- When travelling alone on a long train journey would you be likely to strike up a long conversation with a complete stranger sitting next to you?

- How often do you like to let your hair down, let yourself go and have a real good time?

- If you were asked to give a speech at a function, would you feel happy about doing this?

- How easily do you make friends?

- If you need to approach someone in high authority for a favor, would you prefer to ask them:

- How quickly are you on the dance floor at a social function?

- Would you describe yourself as a leader or a follower?

- What would be your reaction if someone asked you to sell some raffle tickets for charity?

- Do you think people see you as a fun person?

- What would be your reaction if the position of chair suddenly became vacant on a committee on which you were sitting?

- How often do you let your opinions be known?

- Do you enjoy being the centre of attention?

- Which of the following words would you say is the most applicable to you?

- Do you enjoy making small talk at buffet lunches?

- Do you prefer to discuss things face-to-face or over the telephone?

- Would you go out of your way to meet 'the right people'?

- Which of the following words would you say is the most applicable to you?

- Do you enjoy performing your party piece at Christmas parties and other occasions?

- Would you appear naked on a charity calendar?

- Do you ever run out of things to say when talking to someone you have just met?

Tough or tender
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- Enter your Group and Role
- I always seem to find myself rooting for the underdog.
- I admire people who are prepared to admit they were wrong.
- I feel great sympathy for street beggars.
- I believe that there is such a thing as love at first sight.
- I always feel some sympathy for celebrities who are having a bad time in the press.
- I am turned off completely by vulgar jokes and sexual innuendo.
- After a serious argument with my partner all I want to do is make up as quickly as possible
- If someone does me a bad turn I don't waste time thinking of revenge.
- My heart rules my head more than my head rules my heart.
- I would put in a good word for a work colleague who I thought deserved my support.
- I detest watching movies that contain excessive violence.
- I feel very sorry for people who always seem to be the butt of other people's jokes.
- I would encourage anyone to talk over their troubles with me.
- I have always ensured that I put aside some quality time to spend with my partner.
- I always buy my partner a card or present on St.Valentine's Day.
- On occasions my eyes have filled up with tears when watching a movie, be it happy or sad.
- Do you enjoy being the centre of attention?
- I would always go out of my way to help someone who is going through an emotional trauma.

- I would find it extremely difficult to tell anyone some real home truths.
- I have never found it difficult to forgive and forget.
- I like stroking cats and=or dogs.
- I find it difficult to say ‘No’ when asked for a favor.
- I am as supportive of others as I am ambitious for my own aspirations.
- I often feel happy for other people.
- People should be much more concerned about other people.

Success and Risk
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- Enter your Group and Role
- Getting on in business requires ruthlessness.
- I might lack some of the years of experience offered by other candidates but my success comes from the energy and determination that I have to make things happen.
- My success is due to my ability to think strategically while overseeing day-to-day activities.
- Success comes to a great team empowered by exemplary management.
- My success is due to my strong interpersonal skills.
- Drive and determination are the keys to my success.
- My success is due to my ability to think laterally and outside of the box.
- My success is due to my full understanding of the marketplace and competitors’ trends.

- The higher the risk, the higher the potential return.
- The importance of avoiding loss is often underestimated.
- Regulations stifle creativity.
- Success belongs to the bold.
- Provided the customer is happy, everything else should bode well.
- A problem shared is a problem halved.
- It is better to double margins than the customer base.

Optimist or pessimist
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- Enter your Group and Role
- I believe that superstitious beliefs, e.g. 'breaking a mirror brings 7 years' bad luck', are bunkum.
- I never even notice the fire regulations when staying in a hotel, let alone read them.
- I believe in keeping my aspirations high at all times.
- You must speculate to accumulate.
- When one door closes another one always opens.
- I never lose sleep through worrying.
- I am constantly on the lookout for opportunities to move on to new and exciting ventures.
- In life, there is an ideal partner for everyone.

- Every dog has his day.
- In the long run, things always turn out for the better.
- If I lent money to a friend, it would never occur to me that I might not get it back.
- I fully expect that one day I will be a big winner on the lottery or premium bonds.
- I never worry about my health.
- Things are never quite as bad as they appear.
- It is a waste of time going to the doctor with minor complaints such as a mild dose of 'flu.
- If at first you don't succeed, you should try, try and try again.
- I rarely or never worry about my financial situation.
- I am always hopeful that the next stroke of good fortune is just around the corner.
- It is always possible to find a silver lining to every cloud if you look hard enough and long enough.
- Ultimately, good will always triumph over evil.
- I look forward to the post arriving in the morning.
- I very rarely carry an umbrella around with me.
- I always look forward to the future with high expectations.
- Something positive always comes from adversity.
- I am all in favor of taking calculated risks.

Managing people and resources
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- It is better to focus on selling a few more products rather than worry about how much we are spending on stationery.
- Everyone makes mistakes so it is best if we report them immediately.
- I would feel uncomfortable in a situation where resources were being used that did not represent best value for money.
- I understand the importance of effective listening.
- To manage people well you have to get fully involved in the detail.
- Above all else, good management includes trusting people to do the job.
- Yes, managing people is important but it must come second to fulfilling the client's expectations.
- I wish more credit was given to all the positive outcomes that you can't put numbers on.
- I would not normally expect to be part of the important decision making process.
- I could make recommendations that went against my personal beliefs.
- Only those qualified in a subject area should contribute to a debate.
- I feel happiest when I can implement defined regulatory processes.
- I expect to take joint responsibility for important decisions and am comfortable to provide a justification for the conclusions reached.
- I would expect most decisions to be based predominantly on numerical information.
- When painful choices have to be made I find it difficult to commit myself.
- When information is incomplete a decision is best deferred.
- The decisions that really shape an organization or policy are best handed down from senior management.

- The views of someone who has been in an organization only a short time are not as valid as those of someone with long service.
- If you know something is right then it is important to keep telling people no matter how repetitive it becomes.
- A compromise is rarely good for business.

Communicating and Role
As part of the Scientific Teams project, We will try to understand the personality factors that affect learning. The following questions will be answered during weekly meetings

- Above all else my success to date is due to my ability to build and maintain business relationships.
- If a colleague is performing below par then they can expect honest, constructive feedback from me.
- When all the hard work has been done, the key points identified and the recommendations formulated then I feel comfortable if others have the job of selling the policy.
- I pride myself in being able to do a high-pressure job while dealing sensitively with people and issues.
- I have a very direct approach.
- Knowledge is a commodity and so I prefer to keep it to myself.
- I am happiest producing written material and much prefer that role to one that involves presenting an argument orally.
- Being opinionated is not always a bad thing.
- I am used to presenting recommendations to groups of people drawn from all levels of an institution or organization.
- I do not consider it a part of my current job to suggest ways in which something could be done more efficiently.

- Being personable can make up for many potential pitfalls.
- If you can get people to buy into a set of objectives or targets then everyone will work that bit harder towards a shared goal.
- I wish I could more often make novel links between previously unconnected issues.
- I want a job where my cool-headed approach will serve me well.
- I like nothing better than to get my teeth into a challenge.
- I work best when I can get on with my job with the minimum of distractions.
- I feel I perform best in a job where I need to be copied into every e-mail.
- My current job is 24/7 and my next one will be – it goes with the territory.
- I feel resentment if my working life starts to impinge on my home life.
- I prefer a high degree of order and tend to get stressed if things do not go to plan.

Bibliography

Ritu Agarwal and Mani Wadhwa. Review of state-of-the-art design techniques for chatbots. *SN Computer Science*, 1(5):246, 2020. (cited on Page 103)

Ranjana Agrawal and Madhura Phatak. A novel algorithm for automatic document clustering. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 877–882. IEEE, 2013. (cited on Page 94)

Toufique Ahmed, Noah Rose Ledesma, and Premkumar Devanbu. Synshine: improved fixing of syntax errors. *IEEE Transactions on Software Engineering*, 2022. (cited on Page 77)

Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. Network and profile based measures for user similarities on social networks. In *2011 IEEE International Conference on Information Reuse & Integration*, pages 292–298. IEEE, 2011. (cited on Page 11)

Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. User similarities on social networks. *Social Network Analysis and Mining*, 3(3):475–495, 2013. (cited on Page 11)

Ibrahim M Alarifi. *Learning Limitless Knowledge: Transformative Learning Pathway to Unlocking and Harnessing the Endless Power of Knowledge*. Balboa Press, 2023. (cited on Page 81)

Rodrigo Aldecoa and Ignacio Marin. Deciphering network community structure by surprise. *PloS one*, 6(9):e24195, 2011. (cited on Page 35)

James Allen, Andrew D James, and Phil Gamlen. Formal versus informal knowledge networks in r&d: a case study using social network analysis. *R&d Management*, 37(3):179–196, 2007. (cited on Page 29)

Hakam W Alomari, Vijayalakshmi Ramasamy, James D Kiper, and Geoff Potvin. A user interface (ui) and user experience (ux) evaluation framework for cyberlearning environments in computer science and software engineering education. *Heliyon*, 6 (5):e03917, 2020. (cited on Page 75)

Heidi Goodrich Andrade. Teaching with rubrics: The good, the bad, and the ugly. *College teaching*, 53(1):27–31, 2005. (cited on Page 80)

Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Analysis of ego network structure in online social networks. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 31–40. IEEE, 2012. (cited on Page 12 and 13)

Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Ego networks in twitter: An experimental analysis. In *2013 Proceedings IEEE INFOCOM*, pages 3459–3464, 2013. doi: 10.1109/INFCOM.2013.6567181. (cited on Page 13)

Valerio Arnaboldi, Marco Conti, Massimiliano La Gala, Andrea Passarella, and Fabio Pezzoni. Ego network structure in online social networks and its impact on information diffusion. *Computer Communications*, 76:26–41, 2016. (cited on Page 13)

Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin IM Dunbar. Online social networks and information diffusion: The role of ego networks. *Online Social Networks and Media*, 1:44–55, 2017. (cited on Page xix and 14)

Suket Arora, Kamaljeet Batra, and Sarabjit Singh. Dialogue system: A brief review. *arXiv preprint arXiv:1306.4134*, 2013. (cited on Page xxi, 98, 99, and 100)

CV Suresh Babu and PM Akshara. Revolutionizing conversational ai: Unleashing the power of chatgpt-based applications in generative ai and natural language processing. In *Advanced Applications of Generative AI and Natural Language Processing Models*, pages 228–248. IGI Global, 2024. (cited on Page 98)

Mitja D Back and David A Kenny. The social relations model: How to understand dyadic processes. *Social and Personality Psychology Compass*, 4(10):855–870, 2010. (cited on Page 23)

Pamela S Bacon. Collaboration: The next steps. *Library media connection*, 26(6): 36–37, 2008. (cited on Page 29)

Mary Baker. Four modes of collaboration are key to success in hybrid work. 2021. (cited on Page 30)

Vimala Balakrishnan and Chin Lay Gan. Students' learning styles and their effects on the use of social media technology for learning. *Telematics and Informatics*, 33 (3):808–821, 2016. (cited on Page 34)

Aydın Balyer and Kenan Özcan. School principals' instructional feedback to teachers: Teachers' views. *International Journal of Curriculum and Instruction*, 12:295–312, 2020. (cited on Page 1)

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999. (cited on Page 24)

Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):60–69, 2003. (cited on Page xix and 25)

Zane L Berge. Active, interactive, and reflective elearning. *The Quarterly Review of Distance Education*, 3(2):181–190, 2002. (cited on Page 2)

Paul Black and Dylan Wiliam. Inside the black box: Raising standards through classroom assessment. *Phi delta kappan*, 92(1):81–90, 2010. (cited on Page 79)

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. (cited on Page 34 and 36)

Queen Esther Booker, Fred L Kitchens, and Carl Rebman. Evaluating the big five personality test as a foundation for an online student course recommendation system. In *Proceedings of the 34th Annual South West Conference, San Diego*, pages 287–295, 2007. (cited on Page 32)

Steve Borgatti. Ego network analysis, 2011. (cited on Page 15)

Ulrik Brandes. *Network analysis: methodological foundations*, volume 3418. Springer Science & Business Media, 2005. (cited on Page 10)

Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, Antonino Nocera, and Domenico Ursino. Measuring betweenness centrality in social internetworking scenarios. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 666–673. Springer, 2013. (cited on Page 17)

Mary D Burbank and Don Kauchak. An alternative model for professional development: Investigations into effective collaboration. *Teaching and teacher education*, 19(5):499–514, 2003. (cited on Page 28)

Edward Burgin, Sourav Dutta, Haytham Assem, and Raj Nath Patel. Cage: A hybrid framework for closed-domain conversational agents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 636–640. Springer, 2022. (cited on Page 104)

Ronald S Burt. Structural holes and good ideas. *American journal of sociology*, 110 (2):349–399, 2004. (cited on Page 15 and 18)

Lorne Campbell and Deborah A Kashy. Estimating actor, partner, and interaction effects for dyadic data using proc mixed and hlm: A user–friendly guide. *Personal Relationships*, 9(3):327–342, 2002. (cited on Page 19)

Noel A Card, Philip C Rodkin, and Claire F Garandeau. A description and illustration of the triadic relations model: Who perceives whom as bullying whom? *International Journal of Behavioral Development*, 34(4):374–383, 2010. (cited on Page 23)

Brian V Carolan. *Social network analysis and education: Theory, methods & applications*. Sage Publications, 2013. (cited on Page 13)

Rosa M Carro and Victor Sanchez-Horreo. The effect of personality and learning styles on individual and collaborative learning: Obtaining criteria for adaptation. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1585–1590. IEEE, 2017. (cited on Page 34)

Philip Carter. *The complete book of intelligence tests: 500 exercises to improve, upgrade and enhance your mind strength.* John Wiley & Sons, 2009. (cited on Page 32, 33, and 34)

V. Cerf and R. Kahn. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, 22(5):637–648, 1974. doi: 10.1109/TCOM.1974.1092259. (cited on Page 9)

Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, 50 (4):1–37, 2017. (cited on Page 34 and 35)

Charlotte Chalmers, Elaine Mowat, and Maggie Chapman. Marking and providing feedback face-to-face: Staff and student perspectives. *Active Learning in Higher Education*, 19(1):35–45, 2018. (cited on Page 79)

Tomas Chamorro-Premuzic and Adrian Furnham. Personality predicts academic performance: Evidence from two longitudinal university samples. *Journal of research in personality*, 37(4):319–338, 2003. (cited on Page 32)

Sue-Jen Chen and Edward J Caropreso. Influence of personality on online discussion. *Journal of Interactive Online Learning*, 3(2):1–17, 2004. (cited on Page 32 and 34)

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. Knowledge graph completio n: A review. *Ieee Access*, 8:192435–192456, 2020. (cited on Page 105)

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, 2018. doi: 10.18653/v1/D18-1241. URL https://doi.org/10.18653/v1/D18-1241. (cited on Page 128)

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004. (cited on Page 34)

G. D. Colin. Cybernetic big five theory. *Journal of Research in Personality*, 56: 33–58, 2015. (cited on Page 32)

Lucas Cordova, Jeffrey Carver, Noah Gershmel, and Gursimran Walia. A comparison of inquiry-based conceptual feedback vs. traditional detailed feedback mechanisms in software testing education: an empirical investigation. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 87–93, 2021. (cited on Page 79)

Lluís Coromina, Jaume Guia, Germà Coenders, and Anuška Ferligoj. Duocentered networks. *Social networks*, 30(1):49–59, 2008. (cited on Page 17)

Renée Crawford and Louise Jenkins. Blended learning and team teaching: Adapting pedagogy in response to the changing digital tertiary environment. *Australasian Journal of Educational Technology*, 33(2), 2017. (cited on Page 26)

Nick Crossley, D Bellotti, Gemma Edwards, Martin G Everett, Johan Koskinen, and Mark Tranmer. Analyzing ego-net data. *Social Network Analysis for Ego-Nets. Thousand Oaks, CA: Sage*, pages 76–104, 2015a. (cited on Page 15)

Nick Crossley, Elisa Bellotti, Gemma Edwards, Martin G Everett, Johan Koskinen, and Mark Tranmer. *Social network analysis for ego-nets: Social network analysis for actor-centred networks*. Sage, 2015b. (cited on Page 13)

Gabor Csardi, Tamas Nepusz, et al. The igraph software package for complex network research. *InterJournal, complex systems*, 1695(5):1–9, 2006. (cited on Page 41)

Sergio Currarini, Jesse Matheson, and Fernando Vega-Redondo. A simple model of homophily in social networks. *European Economic Review*, 90:18–39, 2016. (cited on Page 14)

Thiago Reis da Silva and Eduardo Henrique da Silva Aranha. Online game-based programming learning for high school students—a case study. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE, 2015. (cited on Page 75)

Hercules Dalianis and Hercules Dalianis. Evaluation metrics and evaluation. *Clinical Text Mining: secondary use of electronic patient records*, pages 45–53, 2018. (cited on Page 113 and 114)

Dan Davies, Divya Jindal-Snape, Chris Collier, Rebecca Digby, Penny Hay, and Alan Howe. Creative learning environments in education—a systematic literature review. *Thinking skills and creativity*, 8:80–91, 2013. (cited on Page 29)

Miranda de Hei, Corina Tabacaru, Ellen Sjoer, Ralph Rippe, and Jos Walenkamp. Developing intercultural competence through collaborative learning in international higher education. *Journal of Studies in International Education*, 24(2):190–211, 2020. doi: 10.1177/1028315319826226. (cited on Page 30)

Federico De Lorenzis, Filippo Gabriele Pratticò, Maurizio Repetto, Enrico Pons, and Fabrizio Lamberti. Immersive virtual reality for procedural training: Comparing traditional and learning by teaching approaches. *Computers in Industry*, 144: 103785, 2023. (cited on Page 98)

Jonathan De Quidt, Lise Vesterlund, and Alistair J Wilson. 20. experimenter demand effects. *Handbook of research methods and applications in experimental economics*, page 384, 2019. (cited on Page 7)

Andrea De Salve, Barbara Guidi, Laura Ricci, and Paolo Mori. Discovering homophily in online social networks. *Mobile Networks and Applications*, 23(6):1715–1726, 2018. (cited on Page 24)

Rich DeJordy and Dan Halgin. Introduction to ego network analysis. *Boston MA: Boston College and the Winston Center for Leadership & Ethics*, 2008. (cited on Page 15)

Philip Denton, Judith Madden, Matthew Roberts, and Philip Rowe. Students' response to traditional and computer-assisted formative feedback: A comparative case study. *British Journal of Educational Technology*, 39(3):486–500, 2008. (cited on Page 79)

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54:755–810, 2021. (cited on Page 98)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (cited on Page 98)

Navid Dianati. Unwinding the hairball graph: Pruning algorithms for weighted complex networks. *Physical Review E*, 93(1):012304, 2016. (cited on Page 45 and 47)

Edwin Diday and JC Simon. Clustering analysis. In *Digital pattern recognition*, pages 47–94. Springer, 1976. (cited on Page 83)

Pierre Dillenbourg, Séverin Lemaignan, Mirweis Sangin, Nicolas Nova, and Gaëlle Molinari. The symmetry of partner modelling. *International Journal of Computer-Supported Collaborative Learning*, 11(2):227–253, 2016. (cited on Page xix, 22, and 23)

C Dilmegani. In-Depth Guide to 5 Types of Conversational AI in 2023 — research.aimultiple.com, 2023. [Accessed 22-11-2023]. (cited on Page 100)

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*, 2017. (cited on Page 117)

RIM Dunbar et al. Mind the gap: or why humans aren't just great apes. *Lucy to language: The benchmark papers*, pages 3–18, 2014. (cited on Page 13)

Robin IM Dunbar. Coevolution of neocortical size, group size and language in humans. *Behavioral and brain sciences*, 16(4):681–694, 1993. (cited on Page 14)

Robin IM Dunbar. Cognitive constraints on the structure and dynamics of social networks. *Group Dynamics: Theory, Research, and Practice*, 12(1):7, 2008. (cited on Page 13)

Robin IM Dunbar and Matt Spoors. Social networks, support cliques, and kinship. *Human nature*, 6(3):273–290, 1995. (cited on Page 13 and 14)

David Easley, Jon Kleinberg, et al. Networks, crowds, and markets: Reasoning about a highly connected world. *Significance*, 9(1):43–44, 2012. (cited on Page 22 and 25)

Roman Egger. Text representations and word embeddings: Vectorizing textual data. In *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications*, pages 335–361. Springer, 2022. (cited on Page 104)

Ainslie Ellis. *Student-centred collaborative learning via face-to-face and asynchronous online communication: What's the difference?* ERIC Clearinghouse, 2001. (cited on Page 31)

Jörn Erbguth, Marianne Schörling, Nathalie Birt, Susann Bongers, Pamina Sulzberger, and Jean-Henry Morin. Co-creating innovation for sustainability. *Gruppe. Interaktion. Organisation. Zeitschrift für Angewandte Organisationspsychologie (GIO)*, pages 1–15, 2022. (cited on Page 56)

Martin Everett and Stephen P Borgatti. Ego network betweenness. *Social networks*, 27(1):31–38, 2005. (cited on Page 17)

Katherine Faust. A puzzle concerning triads in social networks: Graph constraints and the triad census. *Social Networks*, 32(3):221–233, 2010. (cited on Page 12 and 21)

Richard M Felder and Rebecca Brent. Understanding student differences. *Journal of engineering education*, 94(1):57–72, 2005. (cited on Page 31)

Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. Introduction: what is a knowledge graph? *Knowledge graphs: Methodology, tools and selected use cases*, pages 1–10, 2020. (cited on Page 105)

Melissa Forbes. The value of collaborative learning for music practice in higher education. *British journal of music education*, 37(3):207–220, 2020. (cited on Page 30)

Joseph P Forgas, Martie G Haselton, and William Von Hippel. *Evolution and the social mind: Evolutionary psychology and social cognition*, volume 9. Psychology Press, 2011. (cited on Page 13)

John Fox. Teacher's corner: structural equation modeling with the sem package in r. *Structural equation modeling*, 13(3):465–486, 2006. (cited on Page 18 and 19)

Linton C Freeman, Douglas Roeder, and Robert R Mulholland. Centrality in social networks: Ii. experimental results. *Social networks*, 2(2):119–141, 1979. (cited on Page 14)

Martin Frické. The knowledge pyramid: a critique of the dikw hierarchy. *Journal of information science*, 35(2):131–142, 2009. (cited on Page 1)

Marilyn Friend. Myths and misunderstandings about professional collaboration. *Remedial and Special Education*, 21(3):130–160, 2000. (cited on Page 29)

Dennis J Gallagher. Personality, coping, and objective outcomes: Extraversion, neuroticism, coping styles, and academic performance. *Personality and individual differences*, 21(3):421–429, 1996. (cited on Page 33)

Kathleen M Gates, Teague Henry, Doug Steinley, and Damien A Fair. A monte carlo evaluation of weighted community detection algorithms. *Frontiers in neuroinformatics*, 10:45, 2016. (cited on Page 35)

Chuqin Geng, Wenwen Xu, Yingjie Xu, Brigitte Pientka, and Xujie Si. Identifying different student clusters in functional programming assignments: From quick learners to struggling students. *arXiv preprint arXiv:2301.02611*, 2023. (cited on Page 77)

Raji Ghawi, Mirco Schönfeld, and Jürgen Pfeffer. Extracting ego-centric social networks from linked open data. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 471–477. IEEE, 2019. (cited on Page 15)

Louise Gillett. Common ground: Urban architecture. *xxxx*, 2021. (cited on Page 9)

Elizabeth Avery Gomez, Dezhi Wu, and Katia Passerini. Computer-supported team-based learning: The impact of motivation, enjoyment and team contributions on learning outcomes. *Computers & Education*, 55(1):378–390, 2010. (cited on Page 27 and 52)

Richard Gonzalez and Dale Griffin. The correlational analysis of dyad-level data in the distinguishable case. *Personal Relationships*, 6(4):449–469, 1999. (cited on Page 17)

Richard Gonzalez and Dale Griffin. Modeling the personality of dyads and groups. *Journal of Personality*, 70(6):901–924, 2002. (cited on Page 20)

Richard Gonzalez and Dale Griffin. Dyadic data analysis. *xxx*, 2012. (cited on Page 17)

NP Gopalan and B Siva Selvan. *TCP/IP ILLUSTRATED*. PHI Learning Pvt. Ltd., 2008. (cited on Page 9)

Tatiana Gossen, Michael Kotzyba, and Andreas Nürnberger. Graph clusterings with overlaps: Adapted quality indices and a generation model. *Neurocomputing*, 123: 13–22, 2014. (cited on Page 11)

Sanjeev Goyal and Fernando Vega-Redondo. Structural holes in social networks. *Journal of Economic Theory*, 137(1):460–492, 2007. (cited on Page 15)

Peggy Grant and Dale Basye. *Personalized learning: A guide for engaging students with technology*. International Society for Technology in Education, 2014. (cited on Page 80)

Dale Griffin and Richard Gonzalez. Models of dyadic social interaction. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358 (1431):573–581, 2003. (cited on Page 17)

Barbara Guidi. *A distributed dunbar-based framework for online social network*. PhD thesis, Ph. D. dissertation, Department of Computer Science, Univ. Pisa, Pacinotti, PI, 2015. (cited on Page xix, 13, and 14)

Barbara Guidi, Andrea Michienzi, Laura Ricci, and Vincenzo Ambriola. Analysing dunbar circles in facebook groups. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2021. (cited on Page 13)

Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational visual media*, 8(3):331–368, 2022. (cited on Page 118)

Maryam Habibi and Andrei Popescu-Belis. Keyword extraction and clustering for document recommendation in conversations. *IEEE/ACM Transactions on audio, speech, and language processing*, 23(4):746–759, 2015. (cited on Page 84)

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008. (cited on Page 41)

Frank Harary. The determinant of the adjacency matrix of a graph. *Siam Review*, 4 (3):202–210, 1962. (cited on Page 11)

John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007. (cited on Page 79)

Claudia Hauff and Andreas Nürnberger. Utilizing scale-free networks to support the search for scientific publications. In *Proceedings of the 6th Dutch-Belgian Information Retrieval Workshop (DIR 2006), Delft*, pages 57–64. Citeseer, 2006. (cited on Page 24 and 25)

Michael Heilman. *Automatic factual question generation from text.* PhD thesis, Carnegie Mellon University, 2011. (cited on Page 117)

Oliver Hein, Michael Schwind, and Wolfgang König. Scale-free networks. *Wirtschaftsinformatik*, 48(4):267–275, 2006. (cited on Page 24 and 25)

Russell A Hill and Robin IM Dunbar. Social network size in humans. *Human nature*, 14(1):53–72, 2003. (cited on Page 13)

Pamela Hinds and Cathleen McGrath. Structures that work: social structure, work structure and coordination ease in geographically distributed teams. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 343–352, 2006. (cited on Page 15)

Paul W Holland and Samuel Leinhardt. The statistical analysis of local structure in social networks, 1974. (cited on Page 21)

Yan hong Huang and Changfeng Ning. Enhancing critical thinking in chinese students in physical education through collaborative learning and visualization. *Thinking Skills and Creativity*, 2021. (cited on Page 29)

Shwu-Min Horng and Chih-Lun Wu. How behaviors on social network sites and online social capital influence social commerce intentions. *Information & Management*, 57(2):103176, 2020. (cited on Page 15)

Hong Huang, Jie Tang, Lu Liu, JarDer Luo, and Xiaoming Fu. Triadic closure pattern analysis and prediction in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3374–3389, 2015. (cited on Page xix, 21, and 22)

Woei Hung, David H. Jonassen, et al. Problem-based learning. *Handbook of research on educational communications and technology*, 3(1):485–506, 2008. (cited on Page 60)

Dawn Iacobucci. Everything you always wanted to know about sem (structural equations modeling) but were afraid to ask. *Journal of Consumer Psychology*, 19 (4):673–680, 2009. (cited on Page 18)

Noman Islam, Zeeshan Islam, and Nazia Noor. A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*, 2017. (cited on Page 107)

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. (cited on Page 83)

Patrick Jermann, Amy Soller, and Martin Muehlenbrock. From mirroring to guiding: A review of the state of art technology for supporting collaborative learning. In *European Perspectives on Computer-Supported Collaborative Learning*, number CONF, pages 324–331, 2001. (cited on Page 28)

Scott D Johnson, Steven R Aragon, Najmuddin Shaik, and Nilda Palma-Rivas. Comparative analysis of online vs. face-to-face instruction. 1999. (cited on Page 79 and 80)

Rushed Kanawati. Multiplex network mining: A brief survey. *IEEE Intell. Informatics Bull.*, 16(1):24–27, 2015. (cited on Page 34)

Navdeep Kaur and Parminder Singh. Conventional and contemporary approaches used in text to speech synthesis: A review. *Artificial Intelligence Review*, 56(7): 5837–5880, 2023. (cited on Page 100)

David P Kennedy, Grace L Jackson, Harold D Green, Thomas N Bradbury, and Benjamin R Karney. The analysis of duocentric social networks: A primer. *Journal of Marriage and Family*, 77(1):295–311, 2015. (cited on Page 11 and 17)

David A Kenny. Models of non-independence in dyadic research. *Journal of Social and Personal Relationships*, 13(2):279–294, 1996. (cited on Page 18, 19, and 20)

David A Kenny. Terminology and basics of sem. *Terminology*, 2011. (cited on Page 18)

David A Kenny and Deborah A Kashy. The design and analysis of data from dyads and groups. *xxxx*, 2014. (cited on Page 18)

David A Kenny, Deborah A Kashy, and William L Cook. *Dyadic data analysis*. Guilford Publications, 2020. (cited on Page 18)

Arijit Khan. Knowledge graphs querying. *arXiv preprint arXiv:2305.14485*, 2023. (cited on Page 106)

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022. (cited on Page 118)

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. Improving neural question generation using answer separation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6602–6609, 2019. (cited on Page 117)

Paul E King, Paul Schrodt, and Jessica J Weisel. The instructional feedback orientation scale: Conceptualizing and validating a new measure for assessing perceptions of instructional feedback. *Communication Education*, 58(2):235–261, 2009. (cited on Page 1)

Femke Kirschner, Fred Paas, and Paul A Kirschner. A cognitive load approach to collaborative learning: United brains for complex tasks. *Educational psychology review*, 21:31–42, 2009. (cited on Page 30)

Avraham N Kluger and Angelo DeNisi. The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological bulletin*, 119(2):254, 1996. (cited on Page 78 and 79)

Mark L Knapp and John A Daly. *Handbook of interpersonal communication*. Sage, 2002. (cited on Page 17)

Graph Knowledge Graph. What is a knowledge graph?, May 2023. URL https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/. (cited on Page 105)

Meera Komarraju and Steven J Karau. The relationship between the big five personality traits and academic motivation. *Personality and individual differences*, 39(3):557–567, 2005. (cited on Page 32)

Meera Komarraju, Steven J Karau, Ronald R Schmeck, and Alen Avdic. The big five personality traits, learning styles, and academic achievement. *Personality and individual differences*, 51(4):472–477, 2011. (cited on Page 32, 33, and 34)

Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *science*, 311(5757):88–90, 2006. (cited on Page 22)

Pavel N Krivitsky, M Bojanowski, and M Morris. Inference for exponential-family random graph models from egocentrically-sampled data with alter–alter relations. In *National Institute for Applied Statistics Research Australia, University of Wollongong Working Paper*. xxx, 2019. (cited on Page 15)

Sheetal Kusal, Shruti Patil, Jyoti Choudrie, Ketan Kotecha, Sashikala Mishra, and Ajith Abraham. Ai-based conversational agents: A scoping review from technologies to future directions. *IEEE Access*, 2022. (cited on Page 103 and 106)

Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 2005. (cited on Page 130)

Thomas Ledermann and David A Kenny. The common fate model for dyadic data: variations of a theoretically important but underutilized model. *Journal of Family Psychology*, 26(1):140, 2012. (cited on Page xix and 20)

Georgios Liapis, Katerina Zacharia, Kejsi Rrasa, Aikaterini Liapi, and Ioannis Vlahavas. Modelling core personality traits behaviours in a gamified escape room environment. In *European Conference on Games Based Learning*, volume 16, pages 723–731, 2022. (cited on Page xix and 32)

Cristina Liébana-Presa, Elena Andina-Díaz, María-Mercedes Reguera-García, Iván Fulgueiras-Carril, David Bermejo-Martínez, and Elena Fernández-Martínez. Social network analysis and resilience in university students: an approach from cohesiveness. *International journal of environmental research and public health*, 15(10): 2119, 2018. (cited on Page 14)

Rochelle Lieber. *Introducing morphology*. Cambridge University Press, 2021. (cited on Page 99)

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. (cited on Page 130)

Jianfeng Lin, Lei Zhang, Ming He, Hefu Zhang, Guiquan Liu, Xiuyuan Chen, and Zhongming Chen. Multi-path relationship preserved social network embedding. *IEEE Access*, 7:26507–26518, 2019. (cited on Page 10)

Kevin Lin, Sumant Guha, Joe Spaniac, and Andy Zheng. Nifty web apps: Build a web app for any text-based programming assignment. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1236–1237, 2021a. (cited on Page 75)

Zihang Lin, Yuwei Zhang, Qingyuan Gong, Yang Chen, Atte Oksanen, and Aaron Yi Ding. Structural hole theory in social network analysis: A review. *IEEE Transactions on Computational Social Systems*, 2021b. (cited on Page 15)

Elizabeth A Linnenbrink and Paul R Pintrich. Role of affect in cognitive processing in academic contexts. *Motivation, emotion, and cognition: Integrative perspectives on intellectual functioning and development*, pages 57–87, 2004. (cited on Page 102)

Anastasiya A Lipnevich and Jeffrey K Smith. *The Cambridge handbook of instructional feedback*. Cambridge University Press, 2018. (cited on Page 1)

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. Learning to generate questions by learningwhat not to generate. In *The world wide web conference*, pages 1106–1118, 2019. (cited on Page 117)

Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. Learning to predict reciprocity and triadic closure in social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(2):1–25, 2013. (cited on Page 14)

Hugh Louch. Personal network integration: transitivity and homophily in strong-tie relations. *Social networks*, 22(1):45–64, 2000. (cited on Page 22 and 25)

Thomas F Lyons. Role clarity, need for clarity, satisfaction, tension, and withdrawal. *Organizational behavior and human performance*, 6(1):99–110, 1971. (cited on Page 60)

Longxuan Ma, Mingda Li, Wei-Nan Zhang, Jiapeng Li, and Ting Liu. Unstructured text enhanced open-domain dialogue system: A systematic survey. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–44, 2021. (cited on Page 98)

Alexander V Mantzaris and Desmond J Higham. Infering and calibrating triadic closure in a dynamic network. In *Temporal networks*, pages 265–282. Springer, 2013. (cited on Page 22 and 25)

Amelia Manuti, Serafina Pastore, Anna Fausta Scardigno, Maria Luisa Giancaspro, and Daniele Morciano. Formal and informal learning in the workplace: A research review. *International journal of training and development*, 19(1):1–17, 2015. (cited on Page 29)

Ahtsham Manzoor and Dietmar Jannach. Towards retrieval-based conversational recommendation. *Information Systems*, 109:102083, 2022. (cited on Page 106)

Zeev Maoz. Preferential attachment, homophily, and the structure of international networks, 1816–2003. *Conflict Management and Peace Science*, 29(3):341–369, 2012. (cited on Page 24)

Olivera Marjanovic. Learning and teaching in a synchronous collaborative environment. *Journal of Computer Assisted Learning*, 15(2):129–138, 1999. (cited on Page 31)

Jay Martin. *The education of John Dewey: A biography.* Columbia University Press, 2003. (cited on Page 28)

B Mazur, Karl Rubin, and Alice Silverberg. Twisting commutative algebraic groups. *Journal of Algebra*, 314(1):419–438, 2007. (cited on Page 21)

Barry Mazur. When is one thing equal to some other thing. *Proof and other dilemmas: Mathematics and philosophy*, pages 221–242, 2007. (cited on Page 21)

Christopher McCarty, Peter D Killworth, and James Rennell. Impact of methods for reducing respondent burden on personal network structural measures. *Social networks*, 29(2):300–315, 2007. (cited on Page 15)

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001. (cited on Page 14)

Michael McTear. Rule-based dialogue systems: Architecture, methods, and tools. In *Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots*, pages 43–70. Springer, 2021. (cited on Page xxi and 103)

Michael McTear. *Conversational ai: Dialogue systems, conversational agents, and chatbots.* Springer Nature, 2022. (cited on Page 104)

Shoeb Ahmed Memon, Bonaventura HW Hadikusumo, and Riza Yosia Sunindijo. Using social interaction theory to promote successful relational contracting between clients and contractors in construction. *Journal of Management in Engineering*, 31(6):04014095, 2015. (cited on Page 9)

Erinc Merdivan, Deepika Singh, Sten Hanke, Johannes Kropf, Andreas Holzinger, and Matthieu Geist. Human annotated dialogues dataset for natural conversational agents. *Applied Sciences*, 10(3):762, 2020. (cited on Page 116)

Larry K Michaelsen and Michael Sweet. Team-based learning. *New directions for teaching and learning*, 128(128):41–51, 2011. (cited on Page 27 and 52)

Larry K Michaelsen, Arletta Bauman Knight, and L Dee Fink. Team-based learning: A transformative use of small groups in college teaching. 2004. (cited on Page 27 and 52)

Antonija Mitrovic. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):173–197, 2003. (cited on Page 2)

James Moody. Matrix methods for calculating the triad census. *Social Networks*, 20 (4):291–299, 1998. (cited on Page 21)

Richard L Moreland. Are dyads really groups? *Small Group Research*, 41(2):251–267, 2010. (cited on Page 17)

Lailil Muflikhah and Baharum Baharudin. Document clustering using concept space and cosine similarity measurement. In *2009 International conference on computer technology and development*, volume 1, pages 58–62. IEEE, 2009. (cited on Page 94)

Jerome L Myers, Arnold D Well, and Robert F Lorch Jr. *Research design and statistical analysis*. Routledge, 2013. (cited on Page 7)

Shivangi Nagdewani and Ashika Jain. A review on methods for speech-to-text and text-to-speech conversion. *International Research Journal of Engineering and Technology (IRJET)*, 7(05), 2020. (cited on Page 99)

Susanne Narciss. Feedback in instructional contexts. *Encyclopedia of the Learning Sciences, Volume F (6)*, pages 1285–1289, 2012. (cited on Page 1)

Piia Näykki, Jaana Isohätälä, Sanna Järvelä, Johanna Pöysä-Tarhonen, and Päivi Häkkinen. Facilitating socio-cognitive and socio-emotional monitoring in collaborative learning with a regulation macro script–an exploratory study. *International Journal of Computer-Supported Collaborative Learning*, 12(3):251–279, 2017. (cited on Page 60 and 62)

Mark Ed Newman, Albert-László Ed Barabási, and Duncan J Watts. *The structure and dynamics of networks.* Princeton university press, 2006. (cited on Page 25)

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004. (cited on Page 34)

Huy A Nguyen, Shravya Bhat, Steven Moore, Norman Bier, and John Stamper. Towards generalized methods for automatic question generation in educational domains. In *European conference on technology enhanced learning*, pages 272–284. Springer, 2022. (cited on Page 118)

Chukwuka Victor Obionwu, David Broneske, and Gunter Saake. Microblogs-a means for simulating informal learning beyond classrooms. In *Proceedings of the 14th International Conference on Education Technology and Computers*, pages 219–225, 2022a. (cited on Page 3)

Chukwuka Victor Obionwu, Christian Harnisch, Kalu Kalu, David Broneske, and Gunter Saake. An intervention strategy for mitigating the prevalence of syntax errors during task exercise engagements. In *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–6. IEEE, 2022b. (cited on Page 2 and 67)

Chukwuka Victor Obionwu, SM Laique Abbas, Visakh Padmanabhan, Taruna Tiwari, David Broneske, and Gunter Saake. Optical image recognition strategy for keyword extraction and page ranking for slide recommendation system. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6. IEEE, 2023a. (cited on Page 5 and 67)

Chukwuka Victor Obionwu, Kalu Oji Kalu, Paul Blockhaus, David Broneske, and Gunter Saake. A strategy for retrospective evaluation of students sql learning engagements. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–7. IEEE, 2023b. (cited on Page 2, 27, and 67)

Chukwuka Victor Obionwu, Rahul Kumar, Suhas Shantharam, David Broneske, and Gunter Saake. Semantic relatedness: A strategy for plagiarism detection in sql assignments. In *2023 6th World Conference on Computing and Communication Technologies (WCCCT)*, pages 158–165. IEEE, 2023c. (cited on Page 2)

Chukwuka Victor Obionwu, Tiwari Taruna, Baburaj Chovatta Valappil Bhavya, Raikar Nishanth, Singh Walia Damanpreet, Laique Abbas S.M, Okafor chukwuemeka, Broneske David, and Saake Gunter. A domain specific students' assistance system for the provision of instructional feedback. In *2023 22nd IEEE international conference on machine learning and applications (ICMLA)*, pages xxx–xxx. IEEE, 2023d. (cited on Page 5 and 67)

Chukwuka Victor Obionwu, Damanpreet Singh Walia, Taruna Tiwari, Tathagatha Ghosh, David Broneske, and Gunter Saake. Towards a strategy for developing a project partner recommendation system for university course projects. In *2023 6th World Conference on Computing and Communication Technologies (WCCCT)*, pages 144–151. IEEE, 2023e. (cited on Page 4 and 27)

Victor Obionwu, David Broneske, Anja Hawlitschek, Veit Köppen, and Gunter Saake. Sqlvalidator–an online student playground to learn sql. *Datenbank-Spektrum*, 21: 73–81, 2021a. (cited on Page 2 and 69)

Victor Obionwu, Vincent Toulouse, David Broneske, and Gunter Saake. Automatic instructional feedback, and a lecture hub system: A strategy towards nurturing the acquisition of a structured engagement behavior. In *International Conference on Data Management Technologies and Applications*, pages 219–242. Springer, 2021b. (cited on Page 5)

Victor Obionwu, David Broneske, and Gunter Saake. A collaborative learning environment using blogs in a learning management system. In *International Conference on Computer Science and Education in Computer Science*, pages 213–232. Springer, 2022c. (cited on Page 3 and 27)

Victor Obionwu, David Broneske, and Gunter Saake. Topic maps as a tool for facilitating collaborative work pedagogy in knowledge management systems. In *Proceedings of the 12th International Conference on Information Communication and Management*, pages 50–60, 2022d. (cited on Page 3)

Victor Obionwu, Andreas Nurnberger, Anja Hawlitschek, and Gunter Saake. Learner centered network models: A survey. *International Journal on Integrating Technology in Education*, 2022e. URL https://api.semanticscholar.org/CorpusID:250574447. (cited on Page 4 and 10)

Victor Obionwu, Andreas Nürnberger, and Gunter Saake. A skill sharing platform for team collaboration and knowledge exchange. In *WEBIST*, pages 365–372, 2022f. (cited on Page 27)

Victor Obionwu, Vincent Toulouse, David Broneske, and Gunter Saake. Slide-recommendation system: A strategy for integrating instructional feedback into online exercise sessions. In *A Springer Nature Computer Science*, pages 541–548, 01 2022g. doi: 10.5220/0011351000003269. (cited on Page 4 and 67)

Victor Obionwu., Vincent Toulouse., David Broneske., and Gunter Saake. Slide-recommendation system: A strategy for integrating instructional feedback into online exercise sessions. In *Proceedings of the 11th International Conference on Data Science, Technology and Applications - DATA,*, pages 541–548. INSTICC, SciTePress, 2022. ISBN 978-989-758-583-8. doi: 10.5220/0011351000003269. (cited on Page 72)

Victor Obionwu, Vincent Toulouse, David Broneske, and Gunter Saake. Slide-recommendation system: A strategy for integrating instructional feedback into online exercise sessions. pages 541–548, 01 2022a. doi: 10.5220/0011351000003269. (cited on Page xx, xxi, xxiv, xxv, 82, 83, 87, 88, 89, 90, 91, 92, 93, 94, 95, and 96)

Victor Obionwu, Vincent Toulouse, David Broneske, and Gunter Saake. Slide-recommendation system: A strategy for integrating instructional feedback into online exercise sessions. In *Proceedings of the 11th International Conference on Data Science, Technology and Applications*, pages 542–549, 2022b. (cited on Page 5)

Victor Obionwu, David Broneske, and Gunter Saake. Leveraging educational blogging to assess the impact of collaboration on knowledge creation. *International Journal of Information and Education Technology*, 13(5), 2023f. (cited on Page 3, 27, and 56)

Victor Obionwu, Vincent Toulouse, David Broneske, and Gunter Saake. Automatic instructional feedback, and a lecture hub system: A strategy towards nurturing the acquisition of a structured engagement behavior. In Alfredo Cuzzocrea, Oleg Gusikhin, Slimane Hammoudi, and Christoph Quix, editors, *Data Management Technologies and Applications*, pages 219–242, Cham, 2023g. Springer Nature Switzerland. (cited on Page 4 and 67)

Victor Obionwu, Ilapavuluri Devi, Prasad, David Broneske, and Gunter Saake. A study partner recommender system using a community detection algorithm. In *A Springer Nature Computer Science*, pages 000–000, Cham, 2024. Springer Nature Switzerland. (cited on Page 4 and 27)

AE Oke, SO Olatunji, AO Awodele, JA Akinola, and M Kuma-Agbenyo. Importance of team roles composition to success of construction projects. *International Journal of Construction Project Management*, 8(2):141, 2016. (cited on Page 60)

Joseph A Olsen and David A Kenny. Structural equation modeling with interchangeable dyads. *Psychological methods*, 11(2):127, 2006. (cited on Page 17)

Isaiah Mulang' Onando. *Knowledge Context for Entity and Relation Linking*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2021. (cited on Page 106)

Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social networks*, 35(2):159–167, 2013. (cited on Page 22 and 25)

Sumit Pandey and Srishti Sharma. A comparative study of retrieval-based and generative-based chatbots using deep learning and machine learning. *Healthcare Analytics*, 3:100198, 2023. (cited on Page 103)

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. ACL, 2002. (cited on Page 129)

Glenn M Parker. What makes a team effective or ineffective. *Organisational Development, Jossey-Bass Reader, San Francisco*, pages 656–680, 2006. (cited on Page 60)

Andrea Passarella, Robin IM Dunbar, Marco Conti, and Fabio Pezzoni. Ego network models for future internet social networking environments. *Computer Communications*, 35(18):2201–2217, 2012. (cited on Page 15)

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (cited on Page 36, 39, and 40)

Svetozar Pejovich. The effects of the interaction of formal and informal institutions on social stability and economic development. *Institutions, Globalisation and Empowerment*, 56, 2006. (cited on Page 29)

Brea L Perry, Bernice A Pescosolido, and Stephen P Borgatti. *Egocentric network analysis: Foundations, methods, and models*, volume 44. Cambridge university press, 2018. (cited on Page 15, 16, and 17)

Larry L Peterson and Bruce S Davie. *Computer networks: a systems approach*. Elsevier, 2007. (cited on Page 9)

Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20*, pages 284–293. Springer, 2005. (cited on Page 34)

Arthur E Poropat. A meta-analysis of the five-factor model of personality and academic performance. *Psychological bulletin*, 135(2):322, 2009. (cited on Page 33)

L Portney and M Watkins. Foundations of clinical research: Applications to practice, norwalk, appleton and lange, 2000. *Foot & Ankle Internafional*, 24:557–586, 2020. (cited on Page 1)

Alvin Prasad, Kaylash Chaudhary, and Bibhya Sharma. Programming skills: Visualization, interaction, home language and problem solving. *Education and Information Technologies*, 27(3):3197–3223, 2022. (cited on Page 1)

Clara Puga, Uli Niemann, Vishnu Unnikrishnan, Miro Schleicher, Winfried Schlee, and Myra Spiliopoulou. Discovery of patient phenotypes through multi-layer network analysis on the example of tinnitus. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2021. (cited on Page 40)

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. (cited on Page 140)

Elie Raad, Richard Chbeir, and Albert Dipanda. User profile matching in social networks. In *2010 13th International Conference on Network-Based Information Systems*, pages 297–304. IEEE, 2010. (cited on Page 11)

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. (cited on Page 117)

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435. (cited on Page 118)

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007. (cited on Page 34)

Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003. (cited on Page 84)

Automatic Speech Recognition—ASR. Automatic speech recognition—asr. 2013. (cited on Page 99)

Gregorio Rius-Sorolla, Julien Maheut, Sofia Estelles-Miguel, and Jose P Garcia-Sabater. Collaborative distributed planning with asymmetric information. a technological driver for sustainable development. *Sustainability*, 13(12):6628, 2021. (cited on Page 28)

Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105 (4):1118–1123, 2008. (cited on Page 34)

Willibald Ruch, Fabian Gander, Tracey Platt, and Jennifer Hofmann. Team roles: Their relationships to character strengths and job satisfaction. *The Journal of Positive Psychology*, 13(2):190–199, 2018. (cited on Page 60)

Gunter Saake. *Datenbanken: Konzepte und Sprachen*. mitp Verlags, Frechen, 6. auflage edition, 2018. ISBN 9783958457782. (cited on Page 67 and 68)

Juan Pablo Saenz and Luigi De Russis. On how novices approach programming exercises before and during coding. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–6, 2022. (cited on Page 77)

Rebecca L Sandefur and Edward O Laumann. A paradigm for social capital. *Rationality and society*, 10(4):481–501, 1998. (cited on Page 23)

Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, pages 27–43, 2022. (cited on Page 117)

Tammy Schellens and Martin Valcke. Collaborative learning in asynchronous discussion groups: What about the impact on cognitive processing? *Computers in Human behavior*, 21(6):957–975, 2005. (cited on Page 31)

Manfred R Schroeder. *Computer speech: recognition, compression, synthesis*, volume 35. Springer Science & Business Media, 2004. (cited on Page 99)

Sailik Sengupta, Jason Krone, and Saab Mansour. On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise. *arXiv preprint arXiv:2104.07149*, 2021. (cited on Page 106)

Barbara Senior. Team roles and team performance: is there 'really'a link? *Journal of occupational and organizational psychology*, 70(3):241–258, 1997. (cited on Page 60)

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*, 2017. (cited on Page 103)

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*, 2016. (cited on Page 117)

Jordana MY Shane. Formal and informal structures for collaboration on a campus-wide information literacy program. *Resource Sharing & Information Networks*, 17 (1-2):85–110, 2005. (cited on Page 29)

Grigori Sidorov, Alexander Gelbukh, Helena Gómez-Adorno, and David Pinto. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, 18(3):491–504, 2014. (cited on Page 83)

Anne C Sinclair, Samantha A Gesel, Lauren M LeJeune, and Christopher J Lemons. A review of the evidence for real-time performance feedback to improve instructional practice. *The Journal of Special Education*, 54(2):90–100, 2020. (cited on Page 1)

Jitendra Singh, Erica Evans, Amber Reed, Lisa Karch, Karen Qualey, Lovely Singh, and Hannah Wiersma. Online, hybrid, and face-to-face learning through the eyes of faculty, students, administrators, and instructional designers: Lessons learned and directions for the post-vaccine and post-pandemic/covid-19 world. *Journal of Educational Technology Systems*, 50(3):301–326, 2022. (cited on Page 81)

Michael Sintek and Stefan Decker. Triple—a query, inference, and transformation language for the semantic web. In *International semantic web conference*, pages 364–378. Springer, 2002. (cited on Page 105)

Kowta Sita Nirmala Kumaraswamy and CM Chitale. Collaborative knowledge sharing strategy to enhance organizational learning. *Journal of Management Development*, 31(3):308–322, 2012. (cited on Page 28)

Jeffrey Smith and Anastasiya A Lipnevich. Instructional feedback: Analysis, synthesis, and extrapolation. In *The Cambridge handbook of instructional feedback*, pages 591–603. Cambridge University Press, 2018. (cited on Page 77)

Mark Snyder and William Ickes. Personality and social behavior. *Handbook of social psychology*, 2(3):883–947, 1985. (cited on Page 31)

Hyo-Jeong So, Lay Hoon Seah, and Hwee Leng Toh-Heng. Designing collaborative knowledge building environments accessible to all learners: Impacts and design challenges. *Computers & Education*, 54(2):479–490, 2010. (cited on Page 28)

Tingting Song, Qian Tang, and Jinghua Huang. Triadic closure, homophily, and reciprocation: an empirical investigation of social ties between content providers. *Information Systems Research*, 30(3):912–926, 2019. (cited on Page 22 and 25)

Peter Spirtes. Latent structure and causal variables. *xxx*, 2001. (cited on Page 20)

K Sreelakshmi, PC Rafeeque, S Sreetha, and ES Gayathri. Deep bi-directional lstm network for query intent detection. *Procedia computer science*, 143:939–946, 2018. (cited on Page xxi and 105)

Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, page 64, 2000. (cited on Page 94)

Robert Sugden. Experiments as exhibits and experiments as tests. *Journal of Economic Methodology*, 12(2):291–302, 2005. (cited on Page 7)

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3930–3939, 2018. (cited on Page 117)

Eric Sundstrom, Kenneth P De Meuse, and David Futrell. Work teams: Applications and effectiveness. *American psychologist*, 45(2):120, 1990. (cited on Page 27)

Gerald J Sussman. A computational model of skill acquisition. *xxx*, xxx(xxx):xxx, 1973. (cited on Page 2)

Alistair Sutcliffe, Robin Dunbar, Jens Binder, and Holly Arrow. Relationships and the social brain: integrating psychological and evolutionary perspectives. *British journal of psychology*, 103(2):149–168, 2012. (cited on Page 12, 13, and 14)

Daniel D Suthers, Ravi Vatrapu, Richard Medina, Samuel Joseph, and Nathan Dwyer. Beyond threaded discussion: Representational guidance in asynchronous collaborative learning environments. *Computers & Education*, 50(4):1103–1127, 2008. (cited on Page 31)

Shazia Tabassum, Fabiola SF Pereira, Sofia Fernandes, and João Gama. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1256, 2018. (cited on Page 13 and 16)

Chatbot Team. Hybrid Chatbot: Everything You Need To Know (2023 Complete Guide) — linkedin.com. https://www.linkedin.com/pulse/hybrid-chatbot-everything-you-need-know-2023-complete-guide/. [Accessed 14-02-2024]. (cited on Page 104)

Dimitrios Theofanidis and Antigoni Fountouki. Limitations and delimitations in the research process. *Perioperative Nursing-Quarterly scientific, online official journal of GORNA*, 7(3 September-December 2018):155–163, 2018. (cited on Page 6)

Peter Tomlinson. Psychological theory and pedagogical effectiveness: The learning promotion potential framework. *British journal of educational psychology*, 78(4): 507–526, 2008. (cited on Page 1)

Keith Topping. Peer assessment between students in colleges and universities. *Review of educational Research*, 68(3):249–276, 1998. (cited on Page 78)

Vincent A Traag, Rodrigo Aldecoa, and J-C Delvenne. Detecting communities using asymptotical surprise. *Physical review e*, 92(2):022816, 2015. (cited on Page 35)

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019. (cited on Page xix, 34, 36, 37, 38, 42, and 43)

Danny C.L. Tsai, Anna Y.Q. Huang, Owen H.T. Lu, and Stephen J.H. Yang. Automatic question generation for repeated testing to improve student learning outcome. In Maiga Chang, Nian-Shing Chen, Demetrios G Sampson, and Ahmed Tlili, editors, *Proceedings - IEEE 21st International Conference on Advanced Learning Technologies, ICALT 2021*, Proceedings - IEEE 21st International Conference on Advanced

Learning Technologies, ICALT 2021, pages 339–341. Institute of Electrical and Electronics Engineers Inc., July 2021. doi: 10.1109/ICALT52272.2021.00108. Publisher Copyright: © 2021 IEEE.; 21st IEEE International Conference on Advanced Learning Technologies, ICALT 2021 ; Conference date: 12-07-2021 Through 15-07-2021. (cited on Page 119)

Maksim Tsvetovat and Alexander Kouznetsov. *Social Network Analysis for Startups: Finding connections on the social web.* " O'Reilly Media, Inc.", 2011. (cited on Page xix, 21, and 22)

Jonathan H Turner. *A theory of social interaction.* Stanford University Press, 1988. (cited on Page 9)

Shahadat Uddin, Liaquat Hossain, et al. Dyad and triad census analysis of crisis communication network. *Social Networking*, 2(01):32, 2013. (cited on Page xix and 21)

Fabienne M Van der Kleij, Remco CW Feskens, and Theo JHM Eggen. Effects of feedback in a computer-based learning environment on students' learning outcomes: A meta-analysis. *Review of educational research*, 85(4):475–511, 2015. (cited on Page 78)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. (cited on Page 118)

Giampaolo Viglia, Ghasem Zaefarian, and Aulona Ulqinaku. How to design good experiments in marketing: Types, examples, and methods. *Industrial Marketing Management*, 98:193–206, 2021. (cited on Page 7)

Hei-Chia Wang, Martinus Maslim, and Chia-Hao Kan. A question–answer generation system for an asynchronous distance learning platform. *Education and Information Technologies*, Mar 2023. ISSN 1573-7608. doi: 10.1007/s10639-023-11675-y. URL https://doi.org/10.1007/s10639-023-11675-y. (cited on Page 140)

Idi Warsah, Ruly Morganna, Muhamad Uyun, Hamengkubuwono Hamengkubuwono, and M. Rani Afandi. The impact of collaborative learning on learners' critical thinking skills. *International Journal of Instruction*, 2021. (cited on Page 28)

Stanley Wasserman and Katherine Faust. Social network analysis: Methods and applications. *American Ethnologist*, 24(1):219–220, 1994. (cited on Page 10, 11, 12, and 21)

Harvey S Wiener. Collaborative learning in the classroom: A guide to evaluation. *College English*, 48(1):52–61, 1986. (cited on Page 29)

Jo Williams. Designing neighbourhoods for social interaction: The case of cohousing. *Journal of Urban design*, 10(2):195–227, 2005. (cited on Page 34)

Mark Wilson. Seeking a balance between the statistical and scientific elements in psychometrics. *Psychometrika*, 78:211–236, 2013. (cited on Page 7)

Ruben Winastwan. Semantic textual similarity with bert, Feb 2023. URL https://towardsdatascience.com/semantic-textual-similarity-with-bert-fc800656e7a3. (cited on Page 122)

Erik Woody and Pamela Sadler. Structural equation models for interchangeable dyads: being the same makes a difference. *Psychological methods*, 10(2):139, 2005. (cited on Page xix, 19, and 20)

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–37, 2008. (cited on Page 84)

Yao Xiong, Hyo-Jeong So, and Yancy Toh. Assessing learners' perceived readiness for computer-supported collaborative learning (cscl): A study on initial development and validation. *Journal of Computing in Higher Education*, 27(3):215–239, 2015. (cited on Page 56)

Puyang Xu and Ruhi Sarikaya. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 136–140. IEEE, 2014. (cited on Page 106)

Yang Yang, Yuxiao Dong, and Nitesh V Chawla. Predicting node degree centrality with the node prominence profile. *Scientific reports*, 4(1):1–7, 2014. (cited on Page 11)

Jessica Yarbro, Katherine McKnight, Stephen Elliott, Alexander Kurz, and Liane Wardlow. Digital instructional strategies and their role in classroom learning. *Journal of Research on Technology in Education*, 48(4):274–289, 2016. (cited on Page 80)

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023. (cited on Page 100)

Hao Yin, Austin R Benson, and Johan Ugander. Measuring directed triadic closure with closure coefficients. *Network Science*, 8(4):551–573, 2020. (cited on Page 22 and 25)

Jeongkoo Yoon, Shane R Thye, and Edward J Lawler. Exchange and cohesion in dyads and triads: A test of simmel's hypothesis. *Social science research*, 42(6): 1457–1466, 2013. (cited on Page 56)

LI Yu-Hui and HUANG Fei. Actor-partner interdependence model (apim): A model for dyadic data analysis. *Advances in Psychological Science*, 18(08):1321, 2010. (cited on Page 17)

Akbar Zaheer and Giuseppe Soda. Network evolution: The origins of structural holes. *Administrative Science Quarterly*, 54(1):1–31, 2009. (cited on Page 15)

Bin Zhang, Paul A Pavlou, and Ramayya Krishnan. On direct vs. indirect peer influence in large social networks. *Information Systems Research*, 29(2):292–314, 2018. (cited on Page 22 and 25)

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. User profile preserving social network embedding. In *IJCAI International Joint Conference on Artificial Intelligence*, 2017. (cited on Page 11)

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6*, pages 662–671. Springer, 2018. (cited on Page 117)

Michael J Zyphur and Dean C Pierides. Making quantitative research work: From positivist dogma to actual social scientific inquiry. *Journal of Business Ethics*, 167: 49–62, 2020a. (cited on Page 7)

Michael J Zyphur and Dean C Pierides. Statistics and probability have always been value-laden: An historical ontology of quantitative research methods. *Journal of Business Ethics*, 167:1–18, 2020b. (cited on Page 7)

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

Magdeburg, 21 February 2024