

The Architect: Development of a Learning Personalization Agent

Mallika Manam*, Shreeya Channappa Yogesh*, Bhavna Mohan*, Goziechukwu Inekwe*, Yashika Balaji*

Otto von Guericke Universität Magdeburg, Germany

Abstract—The education industry has in recent years experienced significant influence from generative artificial intelligence (GAI) and related technology. These AI models can engage in interactions that closely resemble those of a college student. They excel in sentence completion, text generation, and, more lately, math tasks. Given its ever increasing capability and its widely widespread use by students while solving course projects and assignments, the potential for using these models as mediation agents becomes a viable option. This technology has the potential to revolutionize the way students learn and interact with course materials, providing personalized support and feedback. As GAI continues to evolve, educators will need to adapt their teaching methods to effectively incorporate these advanced AI models into the learning process. A learning personalization agent, for example, can assist students by providing real-time recommendations for additional resources and practice activities based on their individual learning needs and progress. Thus, in this paper, we propose a personalized learning agent tailored for the SQL-based e-learning environment. The system will evaluate student interactions within the learning platform and deliver personalized instructions and learning strategies or improvements based on their performance and areas of struggle. This personalized approach aims to enhance student engagement and learning outcomes by catering to their specific needs and learning styles. To achieve this goal, we have selected a BERT model and employed information retrieval and sentence embedding strategies to teach the BERT model to understand structured query language (SQL) tasks and their respective error descriptions. Our proposed method has attained a precision of 56% and a recall of 62%, showing promising results in accurately identifying and addressing student errors in SQL tasks. By continuously refining and updating the BERT model with more data and feedback, we aim to further improve its performance and provide even more effective personalized learning experiences for students.

Index Terms—Personalization agents, Sentence transformers, Sentence embeddings.

I. INTRODUCTION

Most traditional and hybrid pedagogies are designed on the assumption that every student has already acquired the necessary foundational knowledge and skills to be successful in the classroom [1]. However, this is not always the case, as some students may have gaps in their understanding that need to be addressed. These gaps may require additional support and targeted interventions to ensure that all students have equal opportunities for success. Consensus from state-of-the-art literature indicates that students who employ a structured form of learning engagement are more likely to acquire more knowledge compared to those who employ a trial-and-error form of engagement [2], [3]. A student that employs a structured form of problem engagement first understands the interaction medium, the instruction, and the objective of the respective exercises before engaging with any of them. Thus,

a structured form of interaction leads to a more efficient and effective learning process. This, however, requires a level of patience and dedication that may be challenging for some learners. Thus, most learners emphasize the trial-and-error form of learning engagement which has resulted in a recent decline in study motivation among students (as shown in figure 1).

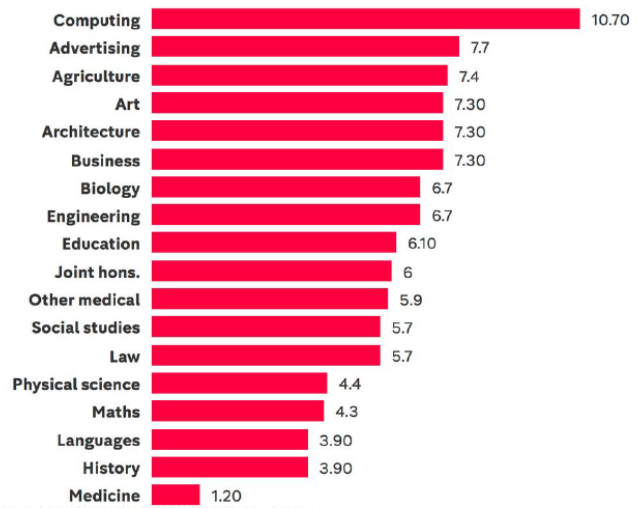


Fig. 1: Percentage of first-year dropouts by subject [4]

However, state-of-the-art research also indicates that small sized traditional lecture scenarios are effective compared to a similar sized full online learning scenario, as most of the online learners subscribe to the trial-and-error form of learning engagement rather than a comprehensive understanding of lecture artifacts. Therefore, incorporating elements of traditional lectures into online learning environments may improve overall effectiveness and engagement for learners.

In this paper, we seek to achieve this by employing pedagogical artificial intelligence agents, which can be trained to mediate interactions between students and online course content, providing personalized guidance and feedback to enhance the learning experience. This approach has been shown to significantly improve learning outcomes and student satisfaction due to current advancements in artificial intelligence technology. We investigated the following research questions:

- What strategies exist for the creation of a motivation-incentive learning personalization system that has significant potential for enhancing learning motivation and engagement?

- What optimum strategies can be utilized to elicit and optimize a student’s learning pathway?

In the next section, we discuss related studies

II. RELATED STUDIES

Interactive virtual entities specifically designed to support learning experiences are referred to as Pedagogical agents. Pedagogical agents were found to enhance learning [5]. They have also immensely contributed in the improvement of the efficiency of educational processes, promotion of global learning, personalization of learning, creation of more intelligent content, and the optimization of educational management in terms of effectiveness and efficiency [6].

Meta Analytic review on embodied pedagogical agent design and testing format [7] states that instead of relying on the outcome labels like transfer, retention and recognition the meta analytic review evaluates the impact that pedagogical agents have on learning outcomes like test formats, test-taking strategy. In case the testing format and testing strategies are similar, excluding studies based on outcome labels could be problematic. The meta analysis study was conducted from 1988 to 2020 with a total of 42 studies and 48 experiments including 4817 participants.

The result of the study builds on the earlier findings that pedagogical agents enhance the learning outcome depending on the agent design and the type of information being learned. The study also introduces a new approach to evaluating the impact of embodied agents and underscores the importance of considering testing formats and strategies in addition to moderating variables related to agent design and content information in future research.

BPR (Bayesian Personalised Ranking from Implicit Feedback) [8] highlights item recommendation with implicit feedback and puts forward a generic optimisation criterion “BPR-Opt” for personalised ranking. After Bayesian analysis, the MAP(Maximum Posterior Estimator) is acquired which is then used to optimise models for rankings. Algorithm LearnBPR based on stochastic gradient descent with bootstrap sampling is introduced in the paper ignored to maximise BPR-Opt. Two state-of-the-art recommender models were used in n experiments which conclude that BPR-Opt surpasses standard learning techniques, focusing on the importance of optimisation of models for the correct criterion. The paper also draws attention on the frequency of implicit feedback in real world scenarios and states that BPR optimisation method is effective for personalised ranking tasks. This in turn offers an empirical and theoretical support for its dominance in prediction quality when compared to other criteria.

PALR (Personalisation Aware LLMs for Recommendation) [9] introduces PALR framework which combines LLMs with user behaviour enhancing personal recommendations. Although, general purpose LLMs point out breakthroughs in natural language processing but their prospects in recommender systems has been obscured.

By using a 7 billion parameters LLM that has been fine-tuned for ranking, integrating user/item interactions for candi-

date retrieval, and creating recommended items based on past user behaviours, PALR solves this void.

In contrast to other methods, PALR places a strong emphasis on integrating rich item side parametric knowledge with the reasoning skills of LLMs. The system achieves competitive performance in sequential recommendation tasks compared to state-of-the-art models by breaking down the recommendation job into three parts: item ranking, candidate retrieval, and user profile generation. The benefits of LLMs in recommendation scenarios are also highlighted in the research, including inductive learning, simple signal integration, knowledge transfer across domains, and the capacity to produce explanations that are understandable to humans. In their recommendations for future study, the authors stress the need to optimise LLM performance and latency in recommendation jobs while preserving accuracy and personalisation.

The paper Large Language Models are Zero-Shot Rankers for Recommender Systems [10] investigates the use of large language models (LLMs), such as GPT-4, as zero-shot rankers in recommender systems. The recommendation problem can be defined as a conditional ranking job, where objects retrieved by candidate generation models are candidates and sequential interaction histories are conditions. The study uses a particular prompting strategy, meticulously creating templates with candidate objects, past encounters, and scoring guidelines. Long-term tests on two widely used recommender system datasets show that LLMs can sometimes beat traditional recommendation models in zero-shot ranking scenarios. But understanding the chronological sequence of past exchanges and overcoming prejudices like popularity and position bias are difficult. To overcome these obstacles, specially crafted prompts and bootstrapping techniques are suggested, showcasing LLMs potential as formidable ranking models in recommender systems. In addition to offering insightful information for the efficient application of LLMs in customised ranking jobs, the study attempts to comprehend the components impacting LLMs’ recommending skills.

III. BACKGROUND

The intelligent agent is designed to initiate a conversation with the user(see figure 2), and on getting a response from the user, a sequence of turns of dialogue continues between the agent and the user. The intelligent agent serves as an intervention agent and is responsible for recommending specific actions which detail what the user should do to solve the user’s specific challenges and meet the user’s objectives.

The specific purposes of the conversation between the user and the agent is to lead to lesser errors, increase the user’s engagement with learning materials, improve the motivation of the student to respond to further chats with the intelligent (intervention) agent and in the end aid the user to obtain better grades through its tailored recommendations for the specific challenges and objectives of the user. Overall, the system behaves in such a manner that leads to enhanced learning motivation and engagement in the user(see figure 3).

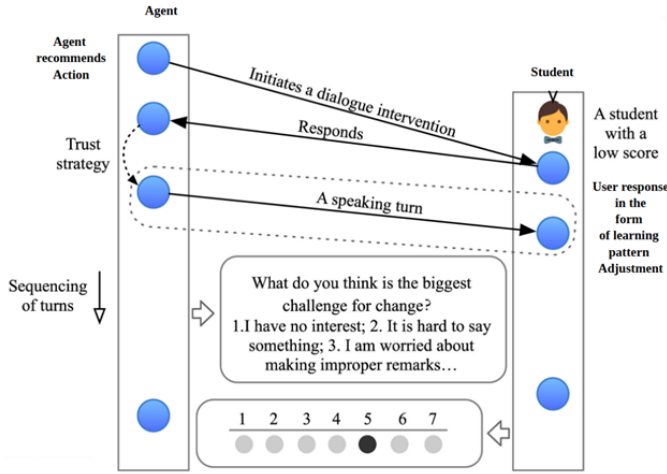


Fig. 2: System of Dialogue between User and Intelligent Agent[11]

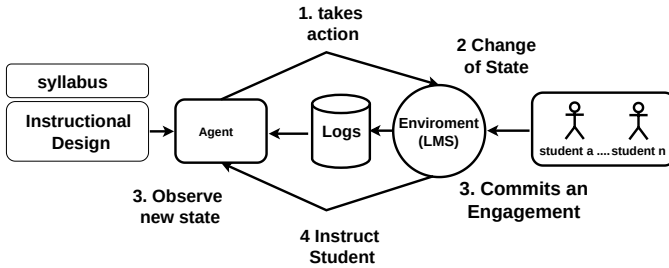


Fig. 3: System Behaviour

IV. SYSTEM ARCHITECTURE

The system architecture detailed in this section outlines the implementation flow of a learning recommendation personalization agent is shown in figure 4. At its core, the architecture underscores a user-centric design, aiming to cater the unique learning needs of individuals by extracting the information encapsulated within their submitted task descriptions and encountered errors. By leveraging this data, the personalization agent tailors learning paths, ensuring that users receive recommendations for their specific challenges and objectives. This user-focused approach holds significance in optimizing the overall learning experience within e-learning platforms, fostering a more adaptive and effective educational journey for individuals with diverse learning requirements.

A. Retrieving User Submissions and Preprocessing

The first step involves retrieving user submissions from the e-learning website, capturing essential information about their task details and error that they encountered. The extracted data undergoes a extensive preprocessing phase, including data cleaning techniques and data integration to create relevant datasets based on the requirements. This step ensures that the dataset is refined and ready for subsequent stages in the recommendation process. High-quality data is imperative to yield optimal recommendations.

Additionally, predefined thresholds for errors are established, serving as triggers for the agent to furnish recommendations when users surpass the designated threshold value for a particular error class. Upon reaching the threshold for a specific error class, the corresponding error class counter is set to 1, signaling that the agent should provide users with recommendations, guiding them along the correct path to rectify potential errors.

Optimizing system efficiency, certain words in task descriptions are substituted with more suitable alternatives. The learning contents has also been preprocessed by replacing the unwanted contents like headers, footers, newline characters etc with blank space. This replacement signifies that these components are deemed non-essential in the context of recommendations, contributing to an overall refinement of the system's functionality.

B. Creating Word Embeddings

Once the data has been preprocessed, we proceed to generate word embeddings, representing the encoded vector form of the words employed in the task description, error description, and learning content. The system generates word embeddings using Sentence Transformer(multi-qa-MiniLM-L6-cos-v1) model for the learning slide contents, utilized for potential recommendation lookups. Similarly, word embeddings are also generated for the enriched query, formed by combining the task description and the error description corresponding to the error class encountered by the user during solution submissions. This approach ensures a comprehensive representation of the textual elements involved, enabling a robust foundation for subsequent recommendation processes.

C. Recommender System

The recommendation process is executed using Dot Product Similarity measure in the Sentence Transformer model when a match is identified between the embeddings of the combination of tasks and error descriptions and the learning content slides. This model leverages its capabilities to recommend the top three slides from the learning content based on the alignment between embeddings of the query and embeddings of the slides. This approach ensures that users receive the most relevant and personalized recommendations, empowering them to enhance their learning experience. The utilization of advanced techniques such as Sentence Transformer enhances the accuracy and relevance of recommendations, contributing to an effective and personalized learning environment within the e-learning platform.

Provided here overall implementation workflow such as the steps involved from data processing till the action suggestion to the student or user(see figure 5).

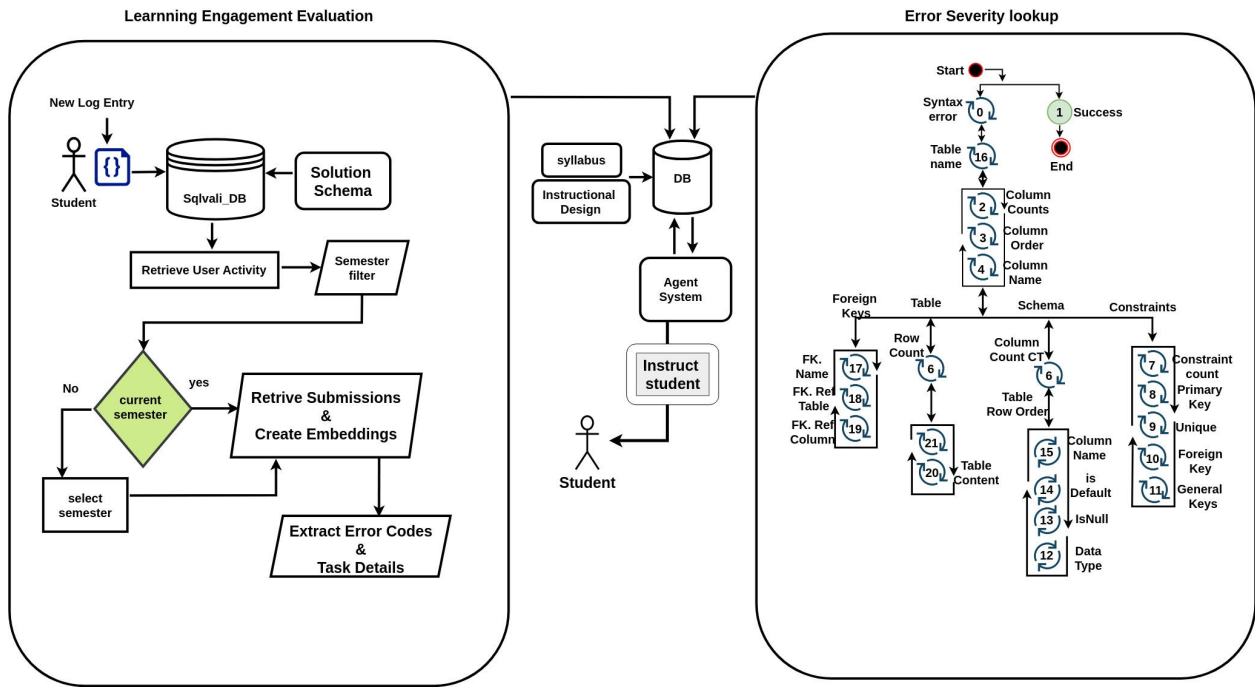


Fig. 4: System Schematic of the learning recommendation agent

```
def preprocessing(Q):
    # 1. Select student records whose attempts
    # crossed the threshold
    filtered_student_records = []
    threshold = 5
    if attempts >= threshold then
        update student record to 1
    end if
    columns_with_1 = []
    df['Severe Error'] = columns_with_1

    # 2. Process slides
    result = remove_slide_header_and_footer(
        text)
    result = replace_newline_with_space(result)
    return result

# Generate sentence embeddings
query1 = get_task_description(tid)
query2 = get_error_description(error_class)
query = concatenate_queries(query1, query2)
def get_encodings(x):
    model = load_model('multi-qa-MiniLM-L6-cos
-v1')
    embeddings = encode_text_with_model(x,
    model)
    return embeddings

# Generate suggestions of slides with high
similarity
def recommend_best_slides(keyword_encodings):
    recommendations = ''
    data['similarity_score'] =
    calculate_similarity_with_keyword_encodings
```

```
(data['encodings'], keyword_encodings)
df_results = sort_data_by_similarity_score(
data)
recommendations =
concatenate_recommendations(df_results
)

def generate_instruction(error_class,
recommendation):
    action = "Learn"
    instruction_synonyms = {
        "Learn": ["Learn", "Study", "Review",
        "Access", "Read", "Go to", "
        Navigate to"]
    }
    instruction_synonym =
    select_random_instruction_synonym(
    instruction_synonyms.get(action, []))
    )

    instruction_text =
    construct_instruction_text(error_class
    , recommendation)

    return instruction_text
```

Listing 1: Pseudocode 1: Algorithm for Data Preprocessing, Action/Suggestion Generation

V. EVALUATION AND DISCUSSION

Results include suggested action by the personalization agent to the errors made by the students. Results are presented in a dataset in figure .

Agent takes into account the SQL questions or tasks, error codes(all possible SQL errors are encoded to numbers), and

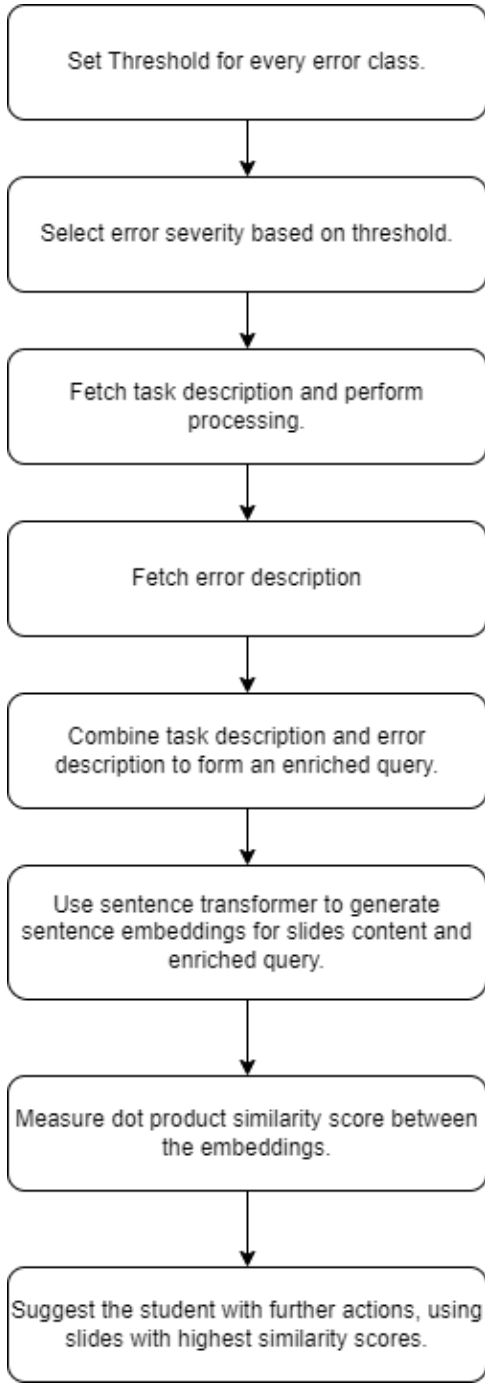


Fig. 5: Implementation workflow

the error descriptions related to the error codes. Based on this information, the agent provides the top 3 slides for the students to go through.

The evaluation results consist of precision, recall, F1 score based on true positives, false positives, and false negatives.

The precision metric measures the proportion of correctly suggested slides over all the suggested slides. It calculates the ratio of true positives to the total positives[12]. The result of the precision metric ranges from 0 to 1. A precision of 0

indicates that the slides of suggested action are less precise. Fig shows the formula of precision.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (1)$$

The Recall metric is formalized as a ratio of true positives to the sum of true positives and false negatives[12]. It defines the ratio of correctly suggested slides to the total slides.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (2)$$

F1 score[3] is formulated as the harmonic mean of precision and recall[12]. It balances precision and recall and provides a single metric. A Higher F1-score tends to provide better balance.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

For evaluation, We have considered Precision, Recall, and F1-measure for the top 3, top 2, and top 1 slide suggestions to the students. Below are the results I which show that performance is optimal for the top 3 predictions compared to the rest. This means that if the prediction in top 1 is not present in the actual truth, then results would often be poor. In terms of suggesting to the students, it is optimal to have more than 1 suggestion for their reference.

Accuracy is not taken into consideration for evaluation as there would be high-class imbalance. Our main focus is identifying correctly suggested slides to the students rather than giving any suggestions.

Metrics	Evaluation and baseline comparison		
	top 3	top 2	top 1
Precision(%)	55.8	56.3	45
Recall(%)	62.5	41.3	16.67
F1-Score(%)	59	47.6	24.3

TABLE I: Evaluation matrix for comparison between baseline and implemented system.

Our evaluation baseline is the MiniLM model(multi-qa-MiniLM-L6-cos-v1) which is trained for multiple question-answering tasks. We have used dot product similarity to compute the similarity between the sentence embeddings of task, error description, and slide. The higher the similarity, the maximum the likelihood of suggesting slides for corresponding tasks and error descriptions.

We compared with 2 other sentence transformers as mentioned in table II. We observed that the evaluation performance is less than 20% for paraphrase-MiniLM-L6-v2 whereas bert-base-nli-mean-tokens accounts for less than 5%. Processing time of the bert model(bert-base-nli-mean-tokens) is almost 50 minutes. This shows that the multi-qa-MiniLM-L6-cos-v1 performs better and faster compared to the other 2 sentence transformers.

Models Comparison		paraphrase-MiniLM-L6-v2	bert-base-nli-mean-tokens
Processing Time(min)		3:34	49:32
Precision(%)	Top 3	17.5	2.5
	Top 2	21.2	2.5
	Top 1	17.5	0
Recall(%)	Top 3	19.2	2.5
	Top 2	14.97	1.67
	Top 1	6.7	0
F1-Score(%)	Top 3	18.3	2.5
	Top 2	17.6	2
	Top 1	9.6	NaN

TABLE II: Evaluation matrix for comparison between other models and implemented system.

VI. CONCLUSION

This work develops a learning personalization agent customized for individual learning interactions in an SQL-based e-learning environment. The learning personalization system is designed to extract information encapsulated within the user's submitted task descriptions and encountered errors, and leverage this data to tailor user-specific learning paths in the form of providing recommendations for the user's specific challenges and objectives. This agent hence takes into account a student's SQL questions or tasks, error codes, and the error descriptions related to the error codes, and using a MiniLM model on the acquired information, provides the top 3 lecture slides for the student to study that will enable the student understand the concepts related to the task and adequately solve the failed task. The proposed methodology in this work attained a precision of 56% and a recall of 62%, thus indicating a noteworthy advancement in our approach.

This user-focused and interactive approach, holds significance in optimizing the overall learning experience of the user within e-learning platforms, fostering a more adaptive and effective educational journey for individuals with diverse learning requirements.

VII. FUTURE WORK

The results in the paper are reliant on the combination of the task and the error description which forms the enriched query. As part of future work, we plan to develop a more comprehensive error description to enhance the results. We also plan to fine-tune the sentence transformer for domain-specific data, to get better embeddings.

REFERENCES

- Scida, E. E. and Saury, R. E., "Hybrid courses and their impact on student and classroom performance: A case study at the university of virginia," *CALICO journal*, pp. 517–531, 2006.
- Davies, C. H., "Student engagement with simulations: a case study," *Computers & Education*, vol. 39, no. 3, pp. 271–282, 2002.
- Damşa, C. I. and Nerland, M., "Student learning through participation in inquiry activities: Two case studies in teacher and computer engineering education," *Vocations and Learning*, vol. 9, no. 3, pp. 275–294, 2016.
- Lee, G., "Which universities have the highest first year dropout rates?" August 2017.
- Krämer, N. C. and Bente, G., "Personalizing e-learning. the social effects of pedagogical agents," *Educational Psychology Review*, vol. 22, pp. 71–87, 2010.
- Iqbal, H. M., Parra-Saldivar, R., Zavala-Yoe, R., and Ramirez-Mendoza, R. A., "Smart educational tools and learning management systems: supportive framework," *International journal on interactive design and manufacturing (IJIDeM)*, vol. 14, pp. 1179–1193, 2020.

- Davis, R. O., Park, T., and Vincent, J., "A meta-analytic review on embodied pedagogical agent design and testing formats," *Journal of Educational Computing Research*, vol. 61, no. 1, pp. 30–67, 2023.
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L., "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.
- Chen, Z., "Palr: Personalization aware llms for recommendation," *arXiv preprint arXiv:2305.07622*, 2023.
- Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., and Zhao, W. X., "Large language models are zero-shot rankers for recommender systems," *arXiv preprint arXiv:2305.08845*, 2023.
- Xie, T., Liu, R., Chen, Y., and Liu, G., "Moca: A motivational online conversational agent for improving student engagement in collaborative learning," *IEEE Transactions on Learning Technologies*, vol. 14, no. 5, pp. 653–664, 2021.
- Dalianis, H. and Dalianis, H., "Evaluation metrics and evaluation," *Clinical Text Mining: secondary use of electronic patient records*, pp. 45–53, 2018.