Expert Agent Guided Learning with Transformers and Knowledge Graphs

Chukwuka Victor Obionwu[1], Bhavya Baburaj Chovatta Valappil[1], Minu Genty[1], Maria Jomy[1], Visakh Padmanabhan[1], Aishwarya Suresh[1], Sumat Singh Bedi[1], David Broneske[2][a] and Gunter Saake[1][b]

[1] *University of Magdeburg, Magdeburg, Germany*

[2] *German Centre for Higher Education Research and Science Studies, Hannover, Germany*

Keywords: Information Retrieval, Generative AI, Knowledge Graphs, Intent Classification, Conversational Agents, Instructional Feedback.

Abstract: The interaction between students and instructors can be likened to an interaction with a conversational agent model that understands the context of the interaction and the questions the student poses. Large language models have exhibited remarkable aptitude for facilitating learning and educational procedures. However, they occasionally exhibit hallucinations, which can result in the spread of inaccurate or false information. This issue is problematic and requires attention in order to ensure the general reliability of the information system. Knowledge graphs provide a methodical technique for describing entities and their interconnections. This facilitates a comprehensive and interconnected understanding of the knowledge in a specific field. Therefore, in order to make the interactions with our conversational agent more human-like and to deal with hallucinations, we employ a retrieval-focused generation strategy that utilizes existing knowledge and creates responses based on contextually relevant information. Our system relies on a knowledge graph, an intent classifier, and a response generator that compares and evaluates question embeddings to ensure accurate and contextually appropriate replies. We further evaluate our implementation based on relevant metrics and compare it to state-of-the-art task-specific retrieve-and-extract architectures. For language generation tasks, we find that the RCG models generate more specific, diverse, and factual information than state-of-the-art baseline models.

1 INTRODUCTION

The success of a pedagogical intervention is generally determined in the short term by the recipients' performance in tests and exams and, in the long term, by the level of mastery they demonstrate with respect to the application of acquired knowledge in a related task. In a programming language course, evaluating knowledge acquisition as a consequence of a pedagogical strategy is a significantly more intricate task (Ismail et al., 2010). It requires assessing not only the correctness of the code but also the efficiency and scalability of the solution. Furthermore, an individual's ability to critically analyze a problem, carry out well-defined procedures, and produce a solution or a set of potential propositions serves as further indicators of knowledge in programming. Also important to consider is the overall impact of the intervention on the students' motivation and engagement in the learning process (Buckley and Doyle, 2016).

Given the difficulty of this challenge and the importance of programming skills in today's world, it is crucial to continuously innovate and improve teaching strategies. In our particular context, we have found that incorporating automatic instructional feedback into a student's online task engagement significantly increased the student's knowledge acquisition and the overall learning process. 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

[a] https://orcid.org/0000-0002-9580-740X

[b] https://orcid.org/0000-0001-9576-8474

Obionwu, C., Valappil, B., Genty, M., Jomy, M., Padmanabhan, V., Suresh, A., Bedi, S., Broneske, D. and Saake, G.
Expert Agent Guided Learning with Transformers and Knowledge Graphs.
DOI: 10.5220/0012860700003756

In *Proceedings of the 13th International Conference on Data Science, Technology and Applications (DATA 2024)*, pages 180-189
ISBN: 978-989-758-707-8; ISSN: 2184-285X

of instructional feedback and also make it automatic for the large population of students using our learning management system, we employed 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. The capacity of an artificial intelligence agent to interact with or give responses resembling those of humans is less likely to result in misinterpretation or user confusion. Replying with clarity facilitates interactions with reduced ambiguity, ultimately resulting in more fruitful talks. Trust is crucial in determining the reliance on AI technology and the acceptance of users. Employing 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. Essentially, a knowledge graph is a meticulously structured framework that consists of a knowledge model. This model includes a network of interrelated explanations that cover concepts, entities, and their relationships. The end result is textbook-backed, factual instructional feedback that is easily accessible to students. Thus, our approach is centered around information retrieval. A Bidirectional Encoder Representations from Transformers (BERT) model is used to decode the intent of a student's query (Devlin et al., 2018). Based on the decoded intent, the system then retrieves relevant information from the knowledge graph. Thus, our approach ensures that students receive reliable and comprehensive answers to their questions, promoting a deeper understanding of the subject matter. Additionally, we adapt and update the knowledge graph regularly, ensuring that the information provided remains up-to-date and relevant. This article outlines the approach we employed towards integrating factual expert knowledge into our learning management system to provide human-like and automated instructional feedback. In comparison to other endeavors concerning the utilization of conversational agents and associated structures, in our approach:

- We demonstrate a strategy for delivering instructional feedback or responses that are contextually appropriate and resemble human-like interactions
- We present a methodology for identifying important entities and relationships in unstructured course material.

The rest of the paper is structured as follows: Section 2 describes introduces the retrieval-based conversational agent in Section 3, we give an overview of our strategy. The implementation of our system is described in Section 5, and in Section II, insight on the background knowledge required for our implementation of our system is described. In Section 6 we evaluate and discuss the performance of our system. In Section VI, we discuss related work and show limitations in Section VII. In Section VIII, we summarize our contributions and indicate directions for future efforts.

2 RETRIEVAL-BASED CONVERSATIONAL AGENTS

Retrieval-based strategies convert non-linguistic structured input queries into natural language representations (Kusal et al., 2022). A retrieval-based conversational agent, which consists of an offline and an online component, chooses the most appropriate response for a user's input using a predefined response repository and ranking model, according to Manzoor et al. (Manzoor and Jannach, 2022). 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 an input text. 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.

According to Ji et al. (Ji et al., 2014), a typical architecture for retrieval-based conversational agents is one in which the conversational agents first use a search engine to find a lot of possible responses and then use a text similarity model to figure out how similar the message and the possible responses are. Several studies investigate how to choose responses that are appropriate for a specific message. The article (Agarwal and Wadhwa, 2020) emphasizes that taking only the most recent input into account when generating responses leads to boring interactions. Thus, it is important for conversational agents to be able to give responses that take into account both past and present conversations. This strategy allows the conversational agent to select the most appropriate response based on the similarity score. Furthermore, by considering both intent and entity identification, the agent can also provide more accurate and contextually relevant

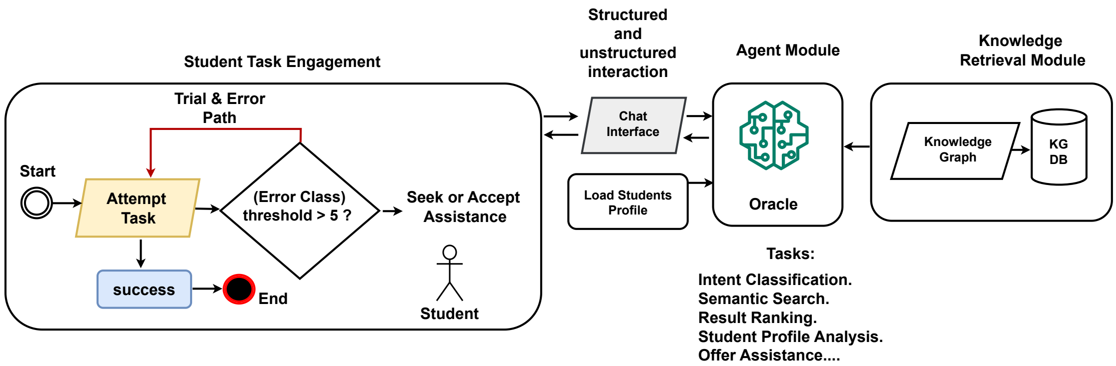

Figure 1: Expert Agent Guided Learning Strategy.

feedback to the user. Additionally, a retrieval-based setup enables an agent to handle a wide range of input queries effectively. In the next section, we give an overview of our expert agent guided learning strategy.

3 EXPERT AGENT GUIDED LEARNING STRATEGY

The primary objective of our learning management system, sqlvalidator (Obionwu et al., 2021a; Obionwu et al., 2021b), is to provide a conducive atmosphere for students to enhance their proficiency in utilizing structured query language. SQLValidator is an online platform that allows users to practice and learn SQL in an interactive manner. The SQLValidator allows users to complete exercises for the database concept course (Saake et al., 2018) and other courses that require the learning of structured query language. 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. Additionally, it incorporates a self-assessment feature that allows students to assess their proficiency in SQL queries outside the main courses, participate in course projects, and provide recommendations during online exercises. Figure 1 shows an overview of the expert agent-guided learning strategy. The goal of the expert agent-guided learning strategy is to provide real-time feedback and suggestions to assist students in their learning engagements. As shown in figure 1, there are two main modules: the knowledge retrieval module and the agent module. The data retrieval module is responsible for retrieving relevant information from the course materials and other approved learning resources, while the agent module uses this information, interaction data, and engagement preferences extracted from the student profile to provide students with personalized recommendations and feedback. This interactive approach enhances student learning by offering tailored support and guidance throughout their learning engagement sessions based on individual needs and preferences. Trial-and-error is a form of learning engagement (Obionwu et al., 2022); thus, we do not force students to engage with the agent. However, the oracle agent continues to monitor the students' progress. 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. According to the responses, the agent provides guidance to the student. The student can also initiate an engagement with the Oracle agent. We categorize this type of conversation as an unstructured interaction, and it follows a question-and-answer format. Some of the tasks performed by the Oracle agent include intent classification, semantic search, result ranking, student profile analysis, offering assistance, etc. These tasks are all focused on the provision of different forms of feedback that serve to instruct the student on how their engagement can be improved. This feedback is classified as instructional feedback. In the next section, we will describe instructional feedback for student success. Specifically, we will focus on the benefits of utilizing digital instructional feedback in educational settings.

4 INSTRUCTIONAL FEEDBACK

According to the authors (Smith and Lipnevich, 2018), instructional feedback is the delivery of information, guidance, and assessment to learners to

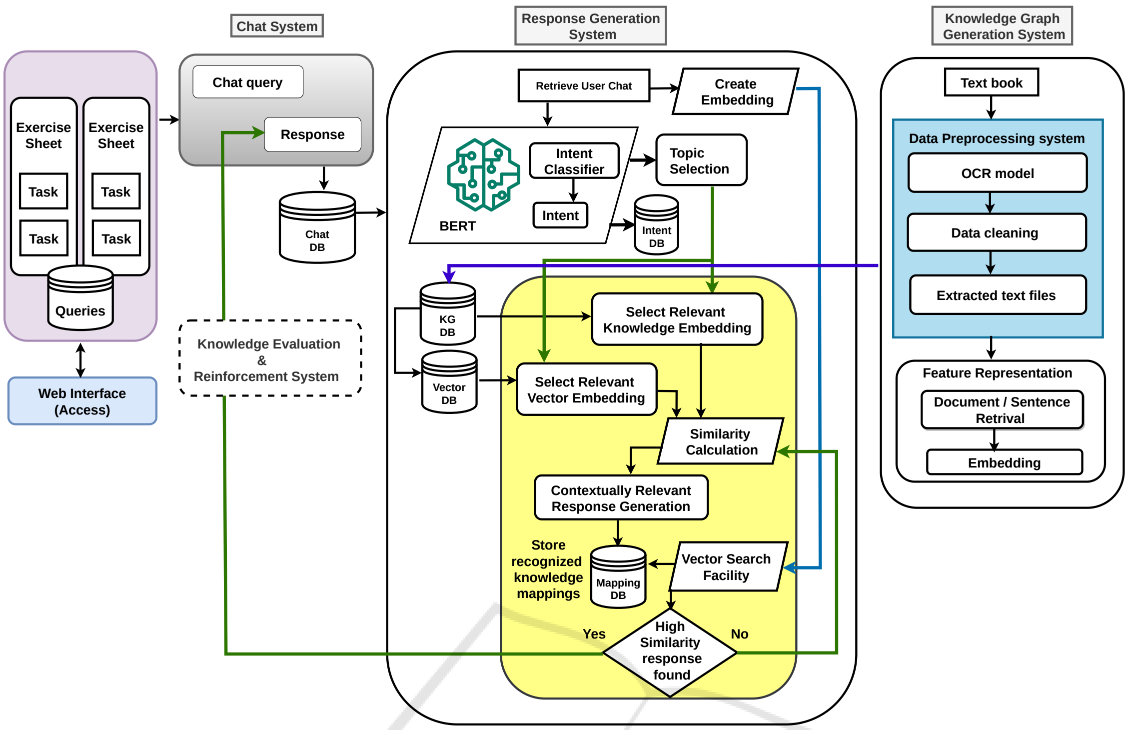

Figure 2: Oracle System.

help them understand their performance and make improvements. It has a vital function in the learning process by way of promoting self-regulation, motivating learners, and enhancing their understanding of the subject matter. Instructional feedback can be provided in several formats, including written comments, verbal discussions, tests, or assessments. Traditional forms of instructional feedback refer to established methods that educators have historically used to evaluate, guide, and support students' learning. These methods, which typically involve direct communication between teachers and students, have been part of educational practices for many years. While newer digital methods of delivering instructional feedback have gained popularity, traditional approaches continue to play a crucial role in effective teaching and learning practices.

4.1 Digital Instructional Feedback

Digital instructional feedback encompasses the utilization of technology-based approaches to deliver guidance, evaluation, and assistance within educational environments. This style of feedback utilizes digital tools and platforms to improve the teaching and learning process, providing a dynamic and frequently customized method of contact between teachers and students. Digital instructional feedback encompasses a range of formats, including online quizzes, interactive exercises, and virtual simulations (Yarbro et al., 2016). These tools and platforms enable educators to promptly deliver feedback and monitor students' progress in real time, facilitating focused interventions and individualized education. Furthermore, digital instructional feedback facilitates active participation and introspection among learners, cultivating a greater sense of autonomy and self-guided learning (Gaytan and McEwen, 2007). Furthermore, digital instructional feedback can help promote the accessibility and inclusivity of education by providing accommodations for learners with various requirements. For instance, students who have visual impairments can gain advantages from the inclusion of audio descriptions or compatibility with screen readers in online quizzes and exercises. In the same way, those with hearing disabilities can make use of closed captioning or transcripts when engaging in virtual simulations (Won et al., 2019). Digital instructional feedback fosters inclusivity and equity in the learning environment by accommodating diverse learning styles and demands.

5 IMPLEMENTATION

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. A strategy we used to achieve this is optical character recognition and feature representation, which are discussed in the next subsection.

5.1 Knowledge Graph Generation System

Optical Character Recognition (OCR) technology is primarily used to convert printed or handwritten documents into a digital version, making them easier to use, save, and access. An optical character recognition (OCR) system can be categorized into two distinct groups: printed character recognition and handwritten character recognition. Printed character recognition involves identifying and converting printed text, while handwritten character recognition focuses on converting handwritten text into digital format. Each type of OCR system uses different algorithms and techniques to accurately interpret and convert the characters. The latter task is particularly challenging due to the absence of uniformity in handwritten characters. Printed letters have a regular and measured size, which makes them easier to recognize compared to other types of letters (Islam et al., 2017). In this work, we employ the printed character recognition strategy.

The stages of optical character recognition are as follows:

- **Pre-Processing:** Once images have been obtained, they are subjected to a number of preprocessing steps in order to enhance their quality. The images are consequently better suited for future applications. Following this phase, techniques such as skew reduction, thinning, and noise removal are utilized.
- **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:** After extracting the features, a classification algorithm is utilized to identify and categorize the characters according to their distinct qualities. This stage is essential for effectively identifying and differentiating various characters.
- **Post-Processing:** The features extracted from the segmented images play a significant role in the character identification process. After the classification phase, other approaches, such as error correction and verification, are used to improve the accuracy of character recognition. These strategies help to minimize any misinterpretations or errors that may have occurred in the previous steps.

Algorithm 1: 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 $data\_dict$

As depicted in Algorithm 1, the data stored in the file *keyword.xlsx* serves as a point of reference for extracting information from the textbook. This data is imported into a data Frame called *keyword_mapping*, which includes columns such as subject, subtopic, and page_numbers. The function *conv_pdf_to_image* is used to translate the appropriate pages from the textbook into images by analyzing each row in *keyword_mapping* and utilizing the page numbers. Following the conversion process, the text is retrieved utilizing the *pytesseract.image_to_string* function and is subsequently repeated for each subtopic, resulting in a thorough extraction of information. The retrieved content is then organized into a nested dictionary called *data_dict*. The top-level keys serve as representations of several themes. Second-level keys serve as a representation of subtopics. The subtopic item contains specific page numbers and selected excerpts. Algorithm 1 shows this explanation. The last stage is feature representation, which seeks to extract significant and useful features from the data, thereby eliminating the necessity for manual feature engineering. Feature learning techniques can enhance performance in di-

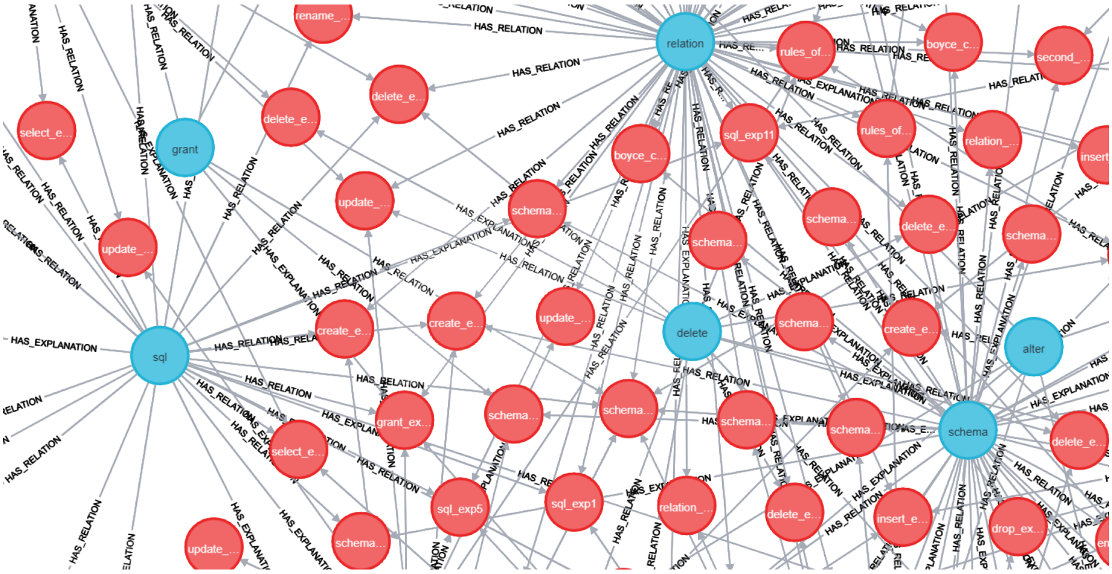

Figure 3: Section of the Derived Knowledge Graph.

verse domains by acquiring representations directly from the data.

As indicated in Algorithm 2, the file *subject.txt* contains the text that describes the given topic. This text is utilized to generate the knowledge graph. Figure 3 depicts a segment of the knowledge graph that was created. The current state of the system includes 196 nodes, 30 labels, 166 explanations, 277 linkages, 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.

5.2 Response Generation System

Once a learner has started using our system, we generate embedding for their queries. The embeddings are compared to the embeddings stored in the vector database. If a response with a high degree of similarity is detected, it is forwarded to the learner as shown in Figure 4. If there are no embedding 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,

Algorithm 2: Feature Representation.

Require: text file $< topic.txt >$
foreach $< topic.txt >$ **do**
  initialize dictionary $node\_data$
  $node\_data \leftarrow \{\text{'}label\text{'} :< topic >\}$
  $file\_lines \leftarrow read\_lines(topic.txt)$
  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

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. The responses are kept in the mapping database and utilized 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 node in the knowledge

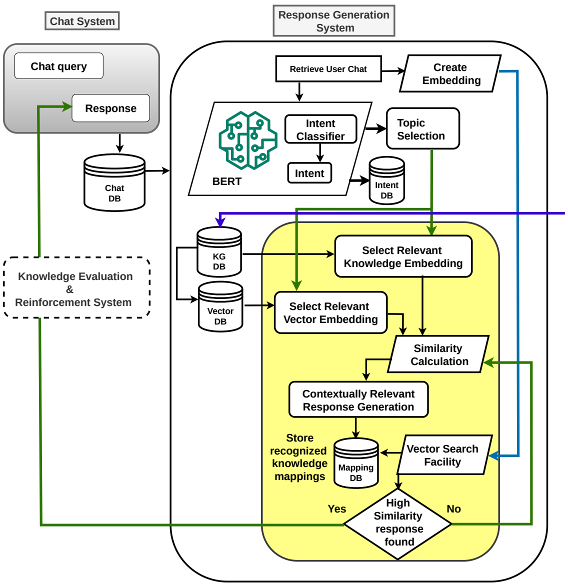

Figure 4: Response Generator Sub-system.

graph, then it is pre-processed 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 3: 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 4: 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** $embedding$ **do**
    $sim \leftarrow calculate\_sim(user\_question\_embedding,$
    $emb)$
    $similarity$ append sim
end while
$response \leftarrow$
$explanation[max\_sim\_index\_embedding]$
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.

6 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.

6.1 System and Model Evaluation

The accuracy metric (Dalianis and Dalianis, 2018) calculates the frequency with which a correct classification is provided for an intent. The accuracy is calculated 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. it must be noted that the prediction values can't be lower than zero, a situation where there are no true predictions or if no predicted cases were marked as true. Figure 5. "FP" designates a false prediction.

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

Figure 5: Formula for precision.

The recall metric (Dalianis and Dalianis, 2018) allows for the calculation of the proportion of accurate

Table 1: 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

predictions out of the total number of accurate forecasts. The formula is illustrated in figure 7. 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.

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 6: Knowledge Graph Evaluation.

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

Figure 7: 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 8.

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

Figure 8: Formula for F-measure metric.

The baseline for our evaluation is the BERT model in its basic form. The baseline BERT model is the simplest model built 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 is unable to differentiate between the SQL keywords ”create” and a task description as ”create a database.”

The results in Table 1 highlights the importance of domain-specific training for conversational agents to effectively answer questions within a particular subject area. It suggests that a conversational agent's performance can be significantly enhanced through targeted training in the specific domain it is intended to operate in.

Figure 6 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 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.

6.2 User Evaluation

Our platform's development primarily focuses on education, so our participants consist of students. Currently, these students are registered in our database courses. As the conversational agent was incorporated into our educational platform, we requested the registered students to evaluate the system by the conclusion of the semester. The survey responses were gathered through a Google form that was connected to our educational platform. The survey questionnaires were designed to get input regarding users' level of satisfaction. 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 11, 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 12. 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 9.

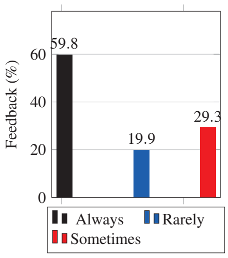

Figure 9: Ease of Use.

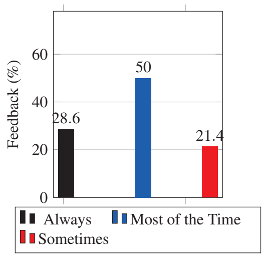

Figure 10: System Interactivity.

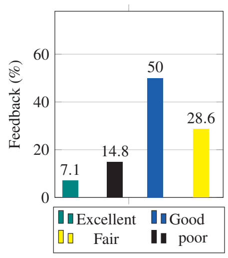

Figure 11: Technical Correctness.

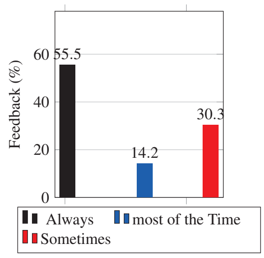

Figure 12: Usability.

7 RELATED STUDIES

Several forms of conversational agents have been deployed in the commercial and industrial sectors. While this is mostly not the case in education, the recent adoption of the blended instructional strategy has opened up avenues for the integration of conversational agents. The agents are agencies for the integration of seemless domain-specific automatic instructional feedback. The following section summarizes the most recent implementations of conversational agents in education and related sectors. Wambsganss et.al (Wambsganss et al., 2021) in their work developed a digital assistant that can provide students with formative comments on their essays. To evaluate students' impressions of their conversational agent, a survey-based assessment was undertaken. Participants were asked to write an argumentative paragraph and use our conversational agent to receive customized feedback. A further effort by Tellols et al. (Tellols et al., 2020) enhanced conversational agents with machine learning technology, thus giving them sentient skills. They showcased their methodology by integrating a virtual instructor into children's educational software. During their assessment, they conducted a comparison between two conversational agents. Based on their findings, the conversational agent, which was augmented with machine learning capabilities, exhibited superior performance and user satisfaction. Our conversational agent utilizes machine learning algorithms to provide automatic instructional feedback to students during their structured language query learning engagement. By leveraging machine learning algorithms, our conversational agent is able to analyze and understand students' language queries, allowing it to provide personalized and targeted instructional feedback. This approach enhances the agent's effectiveness in assisting students in their language learning journey, making it a valuable tool for both educators and learners.

8 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.

9 CONCLUSION

In this research, we discussed the implementation of a domain-specific conversational agent that leverages knowledge graphs and generative AI to enhance learning experiences. By utilizing a retrieval-focused generation strategy, the system aims to provide accurate and contextually relevant responses to students queries. Our study demonstrates that training the model with domain-specific use cases significantly improved its performance compared to general models that lack specialization in the structured query language domain. The future trajectory of our research includes incorporating further learning capabilities, context comprehension, and knowledge evaluation to enhance students' knowledge acquisition. The system's effectiveness is evaluated based on relevant metrics and compared with state-of-the-art models. Additionally, we plan to automate knowledge graph creation, update the system with the latest advancements in natural language processing, and explore potential applications in other educational domains. Our ultimate goal is to provide a comprehensive and user-friendly tool for students to improve their understanding and proficiency in structured query language.

REFERENCES

Agarwal, R. and Wadhwa, M. (2020). Review of state-of-the-art design techniques for chatbots. *SN Computer Science*, 1(5):246.

Babu, C. S. and Akshara, P. (2024). 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.

Buckley, P. and Doyle, E. (2016). Gamification and student motivation. *Interactive learning environments*, 24(6):1162–1175.

Dalianis, H. and Dalianis, H. (2018). Evaluation metrics and evaluation. *Clinical Text Mining: secondary use of electronic patient records*, pages 45–53.

De Lorenzis, F., Pratticò, F. G., Repetto, M., Pons, E., and Lamberti, F. (2023). Immersive virtual reality for procedural training: Comparing traditional and learning by teaching approaches. *Computers in Industry*, 144:103785.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gaytan, J. and McEwen, B. C. (2007). Effective online instructional and assessment strategies. *The American journal of distance education*, 21(3):117–132.

Islam, N., Islam, Z., and Noor, N. (2017). A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*.

Ismail, M. N., Ngah, N. A., and Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *The Turkish Online Journal of Educational Technology*, 9(2):125–131.

Ji, Z., Lu, Z., and Li, H. (2014). An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Kusal, S., Patil, S., Choudrie, J., Kotecha, K., Mishra, S., and Abraham, A. (2022). Ai-based conversational agents: A scoping review from technologies to future directions. *IEEE Access*.

Manzoor, A. and Jannach, D. (2022). Towards retrieval-based conversational recommendation. *Information Systems*, 109:102083.

Merdivan, E., Singh, D., Hanke, S., Kropf, J., Holzinger, A., and Geist, M. (2020). Human annotated dialogues dataset for natural conversational agents. *Applied Sciences*, 10(3):762.

Obionwu, C. V., Harnisch, C., Kalu, K., Broneske, D., and Saake, G. (2022). 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.

Obionwu, V., Broneske, D., Hawlitschek, A., Köppen, V., and Saake, G. (2021a). Sqlvalidator–an online student playground to learn sql. *Datenbank-Spektrum*, 21:73–81.

Obionwu, V., Toulouse, V., Broneske, D., and Saake, G. (2021b). 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.

Saake, G., Sattler, K.-U., and Heuer, A. (2018). *Datenbanken: Konzepte und Sprachen*. MITP-Verlags GmbH & Co. KG.

Sengupta, S., Krone, J., and Mansour, S. (2021). On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise. *arXiv preprint arXiv:2104.07149*.

Smith, J. and Lipnevich, A. A. (2018). Instructional feedback: Analysis, synthesis, and extrapolation. In *The Cambridge handbook of instructional feedback*, pages 591–603. Cambridge University Press.

Tellols, D., Lopez-Sanchez, M., Rodríguez, I., Almajano, P., and Puig, A. (2020). Enhancing sentient embodied conversational agents with machine learning. *Pattern Recognition Letters*, 129:317–323.

Wambsganss, T., Guggisberg, S., and Söllner, M. (2021). Arguebot: A conversational agent for adaptive argumentation feedback. In *Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues*, pages 267–282. Springer.

Won, N., Liu, K., and Bukko, D. (2019). Developing instructional skills: Perspectives of feedback in student teaching. *Networks: An Online Journal for Teacher Research*, 21(2):8.

Xu, P. and Sarikaya, R. (2014). 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.

Yarbro, J., McKnight, K., Elliott, S., Kurz, A., and Wardlow, L. (2016). Digital instructional strategies and their role in classroom learning. *Journal of Research on Technology in Education*, 48(4):274–289.