

Knowledge Graphs for xAI Conversational Agents

Nadezhda Videneeva

University of Bern
Bern, Switzerland

Abstract—This article examines the role of knowledge graphs (KGs) in enhancing the explainability of conversational agents within the context of Explainable AI (xAI). It presents a systematic literature review that analyses the existing methodologies for incorporating KGs into xAI conversational agents and identifies promising avenues for future research. Additionally, this paper introduces a framework that integrates KGs with conversational systems to support diverse AI conversational architectures. This framework leverages the capabilities of cutting-edge Large Language Models (LLMs) to autonomously determine when to extract information from databases and to formulate pertinent queries. This strategy for crafting xAI conversational systems enhances their adaptability and applicability across various fields. Moreover, the capability to present the generated query to the user, alongside the agent's response, significantly boosts the system's reliability and mitigates biases inherent in the agent's replies. The article also showcases the implementation of a basic framework setup, demonstrating its efficacy and robustness. This work underscores the importance of KGs in developing user-centric AI conversational systems, promoting transparency and reliability.

Keywords: Conversational Agent, Explainable AI, Knowledge Graph, Large Language Models

I. INTRODUCTION

In the rapidly evolving landscape of artificial intelligence (AI), conversational agents have become instrumental in automating communication, providing customer service, facilitating work across domains and personalizing user experiences. These AI-driven agents, powered by sophisticated algorithms and vast datasets, are increasingly embedded in our daily lives, from virtual assistants on smartphones to chatbots on websites. However, as these systems grow more complex, their decisions become less transparent, raising concerns about trustworthiness and transparency. The field of Explainable AI (xAI) aims to address these concerns by making AI systems more interpretable to humans, ensuring that users can understand, trust, and effectively manage AI technologies.

Explainability in AI, particularly in conversational agents, is paramount for several reasons. First, it fosters trust between the user and the system by making the decision-making process transparent. Users are more likely to rely on and accept recommendations from a conversational agent whose reasoning they can understand. Second, explainability contributes to the correction of biases in AI models, promoting fairness and ethical AI use. Lastly, it enables developers and stakeholders to detect faults and refine conversational systems, leading to improved performance and reliability. This paper focuses on the integration of knowledge graphs (KGs) with xAI to

enhance the explainability of conversational agents, providing a critical advancement in the development of user-centric AI systems.

Knowledge Graphs are graph-based data structures that store interconnected descriptions of entities (such as objects, events, or concepts), their attributes, and the relationships between them ([10]). KGs facilitate the aggregation and integration of information from diverse sources and provide a semantic framework that enhances data interpretation and reasoning capabilities. In the realm of conversational agents, KGs can be leveraged to enrich the agents' understanding of user queries, enabling more accurate, relevant, and explainable responses.

In this article, I conduct a systematic review of how knowledge graphs can be effectively integrated with xAI techniques to enhance the explainability of conversational agents. This article makes two major contributions:

- I conduct a systematic review of the literature on the use of KGs for explainability, interpretability and transparency of conversational agents, outlining current methodologies, applications and limitations.
- I propose an adaptable framework for integrating KGs with conversational agents to improve their interpretability, which supports various conversational AI architectures and configurations of conversational systems with KG. To validate the robustness and applicability of the framework, I also present an implementation of a simple setup that demonstrates how the integration of KG can enhance the explainability, interpretability and transparency of conversational agents.

A notable challenge encountered during literature review was the scarcity of studies addressing all three components —'xAI,' 'CA,' and 'KG'—comprehensively. This gap highlighted the novelty and complexity of the topic, emphasizing the need for comprehensive studies that integrate all these aspects. Attempts to broaden the scope to include 'Knowledge Bases' yielded minimal additional relevant literature, suggesting the focus on 'Knowledge Graphs' as more relevant to the specific objectives of enhancing explainability of conversational agents.

Following this introduction, the paper is organised as follows: Section 2 describes the survey methodology. Section 3 discusses the concept of explainability within the context of conversational agents. Section 4 reviews the state of the art and

related works, highlighting the advancements and gaps in xAI and KGs for conversational agents. Section 5 presents the design and implementation of the proposed framework. Section 6 concludes the paper, summarising the key findings and outlining directions for future research. Finally, the bibliographical references are listed.

II. METHODOLOGY

To ensure a broad and representative collection of scholarly works, I conducted searches across multiple academic databases known for their extensive coverage of computer science and AI research. These included Google Scholar¹, Semantic Scholar², ACM Digital Library³ and IEEE Xplore⁴. The search strategy was built around a core set of terms: 'xAI,' 'Conversational Agents,' and 'Knowledge Graphs.' These were combined with a variety of synonyms and related terms, such as 'explainable', 'interpretable', 'transparent' for xAI and 'chatbots', 'dialogue systems', 'conversational systems' for CAs. The queries were formulated to include diverse grammatical structures and combinations as well, e.g., "explainable conversational agents with knowledge graphs," "knowledge graphs for CA explainability/transparency", "explaining conversational agents decisions via knowledge graphs", "knowledge-graph-based explainable chatbot", etc. The initial eligibility criteria were defined to include studies that address all three components together, with a few papers selected later to further investigate particular aspects of explainable conversational systems with KGs.

The initial search yielded approximately 150 articles. A preliminary screening was conducted by reviewing titles and abstracts for relevance, based on the defined eligibility criteria. This process narrowed the selection to 40 articles deemed potentially relevant. A subsequent, more detailed review of abstracts and, where necessary, full texts, refined this further to 10 articles. These were chosen for in-depth analysis due to their direct relevance to the study's focus.

III. EXPLAINABILITY OF CONVERSATIONAL AGENTS

This section aims to scrutinize what is expected from xAI conversational agents, delving into the reasons why making them explainable is critical for user and which types of explanations are necessary for various purposes. Furthermore, it will unveil the pivotal role of KGs in achieving these objectives, highlighting how KGs serve as a fundamental tool in making conversational agents more explainable, interpretable, and transparent.

A. *How Knowledge Graphs provide explainability, interpretability and transparency*

One of the primary advantages of KGs is their ability to deliver structured and validated knowledge. By organising information into a network of entities and relations—typically

represented as triples of (head entity, relation, tail entity)—KGs provide a clear and logical framework for understanding complex data sets. This structure enables symbolic reasoning, allowing conversational agents to generate interpretable and logically consistent responses.

The possibility of hierarchical organisation of entities within KGs enables systems to generalise effectively and transparently. For instance, hierarchical relations allow a conversational AI assistant to comprehend and respond accurately to queries about broad categories in a predictable manner. This capability was demonstrated in [7], where the use of KGs enabled a tourist AI assistant to correctly interpret and respond to users' queries by understanding hierarchical relationships among cuisines. This was illustrated when an assistant, backed by a KG, could infer that Italian restaurants fall under the broader category of European cuisine, thereby providing meaningful responses to user queries that would otherwise have resulted in a negative or incorrect reply.

Furthermore, KGs enhance system explainability by enabling the tracing of the origin of information. When <https://scholar.google.com/> conversational agent bases its answers on the information stored in a knowledge graph, it enhances reliability, as the origin of the information is explicitly known. Beyond encompassing common knowledge, KGs can incorporate domain-specific knowledge curated by experts in the field. Such KGs can complement large language models (LLMs), which currently form the backbone of conversational systems, by addressing gaps in domain-specific expertise. There are scenarios where constructing a new knowledge graph for each specific use case becomes necessary. For instance, in the study [1], a new KG is developed for each crime investigation case. Analysts create KGs featuring the individuals and organisations involved in a particular case, which then supports the conversational assistant's responses, aiding in unravelling the investigation. The dynamic nature of KGs, which allows for the continuous addition of new knowledge, is another significant advantage. This feature ensures that conversational agents remain up-to-date and relevant, capable of adapting to new information and changing contexts.

Beyond their role in knowledge representation, KGs serve as visual tools that enhance explainability once an answer has been generated. The visualisation of connections and logic paths within KGs allows users to explore the AI's thought process, facilitating a deeper understanding of its decisions. The crime investigation application mentioned in [1] leverages this capability, offering analysts a visual aid to verify the conversational assistant's reasoning and explore alternative hypotheses. Another study [5] uses this benefit of KG to explain the utterances of a Language Tutor chatbot, which simulates a free-style conversation. The GPT-2 model [15] in the XAI Language Tutor does not directly query the knowledge graph before generating a response. Instead, the authors use KG to visualise the connections and reasoning behind the Tutor's output and, therefore, to enhance its educational value.

However, an even more effective approach involves integrating the chatbot and KG in a synergistic manner, not

1 <https://scholar.google.com/>

2 <https://www.semanticscholar.org/>

3 <https://dl.acm.org/>

4 <https://ieeexplore.ieee.org/Xplore/home.jsp>

only to explain the chatbot's responses but also to guide the agent towards more predictable and accurate behaviour. This integration ensures that the chatbot has direct access to the knowledge contained within the KG. Section 4 discusses various methods for organising such a system and explores different aspects of its functionality.

B. Exploring the aims of explainability

Design and implementation of explainability in conversational agents are influenced by the specific aims and contexts in which these systems are deployed. In the realm of recommendation systems ([4, 6]), explanations serve primarily to persuade users towards certain decisions or preferences. These systems leverage explainability to justify their suggestions, thereby enhancing user satisfaction and trust. Here, errors in reasoning are generally not detrimental, and oftentimes a single, well-crafted explanation may satisfy the user's inquiry or guide their decision. In the context of KG, the task can be viewed as identifying any adequate path between entities that have been mentioned by the user or are relevant to their preferences.

Contrastingly, conversational agents deployed in critical decision-making environments ([1]), such as medical diagnostics or criminal investigations, demand a different approach to explainability. In these high-risk scenarios, the agents are not merely recommending but are part of a shared human-machine reasoning process. The explanations provided by these agents must be comprehensive, transparent, and capable of considering multiple reasoning strategies. This thoroughness ensures that human counterparts can make informed decisions, critically evaluate the AI's suggestions, and explore all possible solutions before arriving at a conclusion. In these scenarios, it is crucial not only to identify singular paths but to explore all possible paths in the KG, including neighbouring nodes and relations, to present a comprehensive overview of the knowledge landscape.

Thus, the design and processing of KGs must be carefully aligned with the intended use case.

IV. INTEGRATING KNOWLEDGE GRAPHS INTO CONVERSATIONAL SYSTEMS

Conversational agents powered by Large Language Models (LLMs) have gained widespread popularity in recent years. Various methodologies have been explored to enhance LLMs with knowledge graphs ([10]): pre-training LLMs with KG information, fine-tuning LLMs on KG-based tasks and dynamically incorporating KG data during inference. Pre-training LLMs with KG information integrates structured knowledge directly into the model, thereby improving its foundational understanding and reasoning capabilities. Fine-tuning LLMs on KG-based tasks tailors the model's performance to specific applications and optimises accuracy for tasks that leverage structured knowledge. Dynamically incorporating KG data during inference enables LLMs to access the knowledge base implicitly and utilise the most up-to-date and contextually relevant knowledge. The primary distinction among these methods lies in the timing and manner of knowledge integration. In my research, I have not

encountered studies employing the first two approaches to specifically enhance explainability. This observation is logical, as these approaches do not facilitate direct interaction between the LLM and the KG, but rather incorporate KG information into the LLM's knowledge base. Furthermore, both approaches necessitate retraining the model whenever the KG is updated.

The method of dynamically incorporating KG data during inference includes an important step of finding the appropriate subgraph containing the information necessary to respond to a user's query. This process is crucial for ensuring that conversational agents can provide accurate, relevant and explainable responses. Numerous distinct algorithms have been developed to tackle this task, presenting several challenges. For instance, in study [4] they address two main challenges: the ability to traverse multiple reasoning paths over background knowledge and the capability to arrange selected entities appropriately under current system intents to control response generation. They introduce a model which integrates tree-structured reasoning over knowledge graphs into a Conversational Recommender System. The tree-structured reasoning algorithm in CR-Walker dynamically constructs a reasoning tree over a knowledge graph for each conversation turn. It starts with the user's current interests and performs a multi-hop reasoning process, expanding the tree by exploring various paths through the graph. At each hop, the algorithm evaluates potential paths using a scoring function that considers the dialog context and user preferences, selecting paths that lead towards a recommendation. Another study, [3], suggests employing an attention mechanism to navigate the graph, predicting edges (relations) that lead to other relevant entities. An additional innovative approach, presented in [2], involves training a reinforcement learning agent to navigate the graph from a specified vertex (representing the input query) to the answer vertex. This model's architecture incorporates a complex policy network that integrates multiple components, including a history-dependent feed-forward network, machine reading comprehension, and a bilinear model for action scoring, to efficiently select knowledge for response generation.

Upon identifying the relevant segment of the knowledge graph, the conversational agent must then generate a response for the user. Earlier studies employed Recurrent Neural Networks to generate responses based on identified paths, as seen in works like [2]. More recent research has shifted towards utilising pre-trained LLMs for this purpose, incorporating dialogue context and information from the KG as token sequences to formulate responses. For instance, in [4], the GPT-2 model [15] is utilised. An even more contemporary method involves the use of prompting. The study [6] explores generating explanations conditioned on conversation history through GPT-3 [13] prompt tuning. This approach focuses on designing prompts that facilitate the generation of appropriate explanations for conversational recommendation scenarios, guiding GPT-3 [13] to produce relevant explanations. Several studies are dedicated to developing effective prompting techniques that integrate the structured data of KGs with the input for LLMs. For example, the paper [8] introduces the Graph Neural Prompting (GNP) method. The GNP framework incorporates a Graph Neural Network encoder to embed KG

information, a cross-modality pooling module for identifying relevant node embeddings, a domain projector to reconcile the differences between graph and textual representations, and a self-supervised link prediction objective to enhance the model's comprehension of KG structure.

However, the latest advancements in LLMs have paved the way for these models to process most steps independently, without needing to retrieve relevant information from the knowledge graph beforehand. The authors of [9] propose an architecture for a conversational system wherein the knowledge graph is stored in a database, and the LLM generates queries to this database based on the user's utterance. Subsequently, the conversational system utilises the database's response to formulate a reply to the user's query. This approach appears promising, especially considering the growing emphasis on LLM evolution. Nonetheless, it remains underexplored, as the article [9] is the only publication I have encountered in my research proposing such a methodology.

V. GUIDELINES FOR DESIGNING EXPLAINABLE CONVERSATIONAL SYSTEM WITH KNOWLEDGE GRAPH

Inspired by the methodology outlined in [9], this paper proposes a framework for a conversational system integrated with a knowledge graph.

A. Framework description

The foundational architecture of this system is demonstrated in Fig.1. Here, a user submits their query to the conversational agent, which autonomously determines the necessity of querying the knowledge graph. If the query to the KG is deemed unnecessary, the agent directly responds to the user. Conversely, if required, the agent formulates a query to the KG, assimilates the acquired knowledge, and utilises this information to craft a response to the user.

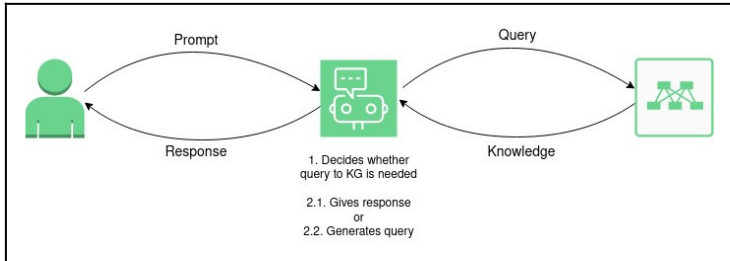


Figure 1. Basic framework.

A major advantage of this framework is its high adaptability, enabling the natural incorporation of various functionalities into any segment of the pipeline. Fig. 2 exemplifies potential enhancements. For instance, if the KG encompasses private information, an access control module can be implemented prior to the query's arrival at the conversational agent. This ensures that users must authenticate themselves and verify their access permissions before querying the KG's contents. Additionally, to augment the system's efficacy, advanced prompting techniques can be employed through a prompt handling module, which refines a user's utterance before it is processed by the agent. Given that Large

Language Models may occasionally generate improper queries to databases, an error handling module could prove invaluable. To improve the agent's accuracy in generating database queries, an innovative approach involves the error handling module providing the agent with prompts highlighting the errors, thereby encouraging the agent to correct the query accordingly. After several iterations, this process significantly increases the likelihood of generating accurate queries. This framework facilitates the integration of numerous other modules designed to enhance different functionalities.

Within this framework, not only can the retrieved knowledge be displayed to the user to ensure interpretability and transparency, but the queries made to the databases can also be made visible. This feature allows users to independently verify the extracted information's relevance and sufficiency for their needs, promoting unbiased decision-making without blind reliance on the agent's responses.

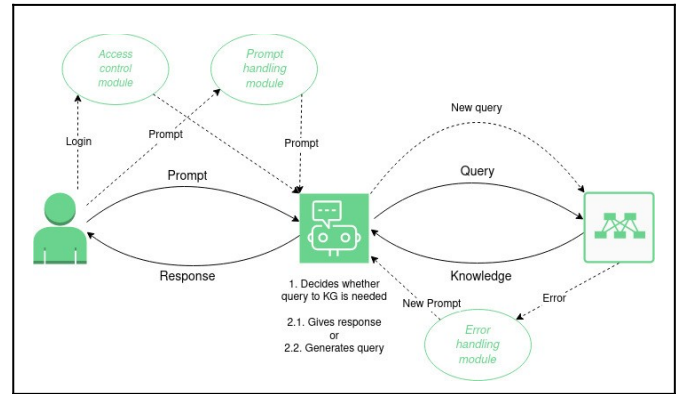


Figure 2. Framework with enhanced functionality.

Moreover, the proposed framework boasts ease of implementation, largely attributed to the agent's responsibility for most operations, such as eliminating the need for identifying relevant subgraphs. This simplicity as well results in the framework's broader applicability.

B. Implementation

To validate the efficacy and robustness of the proposed conversational system framework integrated with a knowledge graph, a preliminary implementation was undertaken⁵. The key components of this experimental setup included:

1. **Language Model (ChatGPT-3.5 [13]):** A state-of-the-art language model developed by OpenAI, known for its ability to understand and generate human-like text based on the input it receives.
2. **Graph Knowledge Base (Neo4j [12]):** A highly scalable and robust graph database platform that excels in storing and querying connected data. Its query language Cypher emphasises human-readable syntax, ensuring that users can structure and interpret complex queries with relative ease. (This database is also used in studies [5, 7, 9])

⁵ <https://drive.google.com/file/d/1FhZcjsewrRC9i77N8L7TxEZISGWUYryL/view?usp=sharing>

3. **Llama Index Module with ReAct Agent and Query Engine (RAG) Tools** [14]: A set of tools designed to facilitate the integration of AI agents with databases, enhancing the capabilities of the agent through structured knowledge.

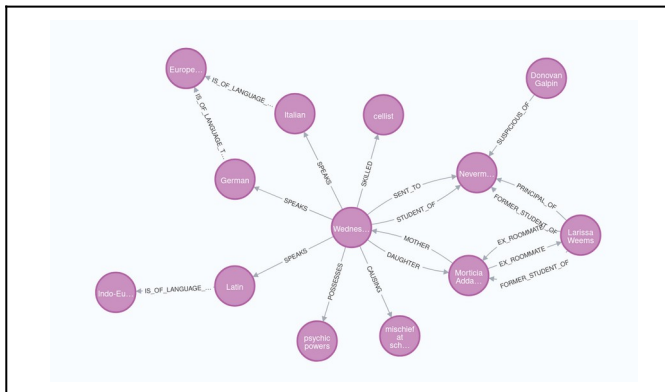


Figure 3. Knowledge Graph used in the experiment.

The experimental process began with the construction of a sample knowledge graph, utilizing the Llama Index library to automatically generate a knowledge graph from the textual content of a Wikipedia article on the "Wednesday" TV series [11]. This process involved segmenting the text into smaller segments, from which triplets were identified and extracted using ChatGPT-3.5 [13]. These triplets, representing subjects, objects, and their connecting predicates, were then integrated into a knowledge graph. Subsequent manual adjustments were made to the graph to enable the demonstration of specific scenarios. The final graph is depicted in Fig. 3.

The agent's loop begins with the reception of a user query, which is first processed by the LLM to extract relevant keywords, entities, and the intent behind the query. Using this processed information, the system attempts to construct and execute a Cypher query against the Neo4j graph database. The agent assesses the results of the Cypher query. If the query successfully returns relevant information, this data is integrated into the LLM's response generation process. In instances where the initial Cypher query fails to obtain relevant information—either due to the specificity of the query surpassing the available data or other factors—the system employs a secondary retrieval strategy. This strategy involves a broader exploration of the graph database, akin to a breadth-first search (BFS), starting from entities identified as related to the user's request. This involves generating a query that selects entities a few steps away from the identified entities, capturing the relationships between every connected pair. Leveraging the data obtained from Neo4j—whether through the initial Cypher query or the fallback BFS-like approach—the conversational agent synthesises this information with its natural language processing capabilities to generate a detailed, informative response to the user's query.

The initial part of the dialogue (Fig. 4) illustrates the agent's behaviour in the absence of relevant information within the knowledge graph. The deletion of actor-related information

during the manual KG refinement was deliberate, to observe this scenario. The agent, despite generating relevant Cypher queries and attempting a BFS-like retrieval, encountered an empty result set. Consequently, the agent communicated its inability to provide the requested information, demonstrating expected and interpretable behaviour. Fig. 5 presents a dialogue instance where the requisite information was successfully derived through Cypher queries, showcasing the agent's capability to formulate non-trivial queries, such as those concerning language types. The graph's response to these queries aligned with the agent's final response to the user, affirming the system's explainability. Another scenario depicted in Fig. 6 involved a query answerable via the BFS retrieval method. Given the lack of a direct relationship between the entities ("Ms. Weems" and "Wednesday Addams") and the non-obvious connection, a direct Cypher query was inapplicable. Thus, the agent explored neighbouring entities, which yielded results and enabled the identification of one of the possible connections, demonstrating the framework's robustness and versatility.

>Have you heard anything about Wednesday?
Thought: (Implicit) I can answer without any more tools!
 Yes, I am familiar with the TV series "Wednesday". It is a popular show that revolves around the character Wednesday Addams from "The Addams Family". The series follows Wednesday as she navigates her teenage years at Nevermore Academy, a peculiar boarding school. Is there anything specific you would like to know about the show?

>Yes, tell me about the leading actress of the series
Thought: I need to use a tool to help me answer the question.
 Action: text2cyther. Action Input: {'input': 'leading actress of Wednesday TV series'}
 Graph Store Query:
 MATCH (a:Entity)-[:PORTRAYS]->(b:Entity)-[:WEDNESDAY_ADDAMS]->(c:Entity)
 RETURN a
 Graph Store Response: []
 Observation: There is no information available in the graph database about the leading actress of the Wednesday TV series.
Thought: The "text2cyther" tool did not provide any information about the leading actress of the Wednesday TV series.
 Action: subgraph_retriver. Action Input: {'input': 'leading actress of Wednesday TV series'}
 Observation: Empty Response
Thought: I cannot answer the question with the provided tools.
 Sorry, I cannot provide information about the leading actress of the Wednesday TV series at the moment.

>Are you sure you don't know anything about Jenna Ortega?
Thought: I need to use a tool to help me answer the question.
 Action: text2cyther. Action Input: {'input': 'Jenna Ortega'}
 Graph Store Query:
 MATCH [j:Entity {id: "Jenna Ortega"}] RETURN j
 Graph Store Response: []
 Observation: Jenna Ortega is not found in the graph database.
Thought: The "text2cyther" tool did not provide any information about Jenna Ortega. I will try using the "subgraph_retriver" tool to see if it can provide any relevant details.
 Action: subgraph_retriver. Action Input: {'input': 'Jenna Ortega'}
 Observation: I'm sorry, but I don't have enough information to provide an answer to your query about Jenna Ortega.
Thought: I cannot answer the question with the provided tools.
 Sorry, I cannot provide information about Jenna Ortega at the moment.

Figure 4. Dialogue about missing information.


```

> Ok, let's talk about Wednesday Addams then. Which languages can she speak?
Thought: I need to use a tool to help me answer the question.
Action: text2cyther. Action Input: {'input': 'Which languages can Wednesday Addams speak?'}
Graph Store Query:
MATCH (w:Entity {id: "Wednesday Addams"})-[:SPEAKS]->(l:Entity)
RETURN l
Graph Store Response:
[{'l': {'id': 'German'}}, {'l': {'id': 'Latin'}}, {'l': {'id': 'Italian'}}]
Observation: Wednesday Addams can speak German, Latin, and Italian.
Thought: (Implicit) I can answer without any more tools!
Great! According to the information I found, Wednesday Addams can speak German, Latin, and Italian. She is a multilingual character in the TV series. If you have any more questions, feel free to ask!
> Does Wednesday Addams speak any European language?
Thought: I need to use a tool to help me answer the question.
Action: text2cyther. Action Input: {'input': 'Does Wednesday Addams speak any European language?'}
Graph Store Query:
MATCH (w:Entity {id: "Wednesday Addams"})-[:SPEAKS]->(e:Entity)-[:IS_OF_LANGUAGE_TYPE]->(t:Entity {id: "European"})
RETURN COUNT(e) > 0
Graph Store Response: [{'COUNT(e) > 0': True}]
Observation: Yes, Wednesday Addams speaks at least one European language.
Thought: The "text2cyther" tool has provided a response indicating that Wednesday Addams speaks at least one European language. I can answer the question without using any more tools.
Yes, Wednesday Addams speaks at least one European language.
> Which of the languages Wednesday speaks is of Indo-European type?
Thought: I need to use a tool to help me answer the question.
Action: text2cyther. Action Input: {'input': 'Which of the languages Wednesday Addams speaks is of Indo-European type?'}
Graph Store Query:
MATCH (w:Entity {id: "Wednesday Addams"})-[:SPEAKS]->(l:Entity)-[:IS_OF_LANGUAGE_TYPE]->(t:Entity {id: "Indo-European"})
RETURN l.id
Graph Store Response: [{'l.id': 'Latin'}]
Observation: Wednesday Addams speaks Latin, which is of Indo-European type.
Thought: I can answer without using any more tools.
Wednesday Addams speaks Latin, which is of Indo-European type.
> What's the name of the school Wednesday attends?
Thought: I need to use a tool to help me answer the question.
Action: text2cyther. Action Input: {'input': 'What is the name of the school Wednesday Addams attends?'}
Graph Store Query:
MATCH (w:Entity)-[:STUDENT_OF]->(s:Entity)
WHERE w.id = "Wednesday Addams"
RETURN s.id
Graph Store Response: [{'s.id': 'Nevermore Academy'}]
Observation: Wednesday Addams attends Nevermore Academy.
Thought: I can answer without using any more tools.
Wednesday Addams attends Nevermore Academy.

```

Figure 5. Dialogue with Cypher queries.

```

> How are Ms. Weems and Wednesday Addams connected?
Thought: I need to use a tool to help me answer the question.
Action: text2cyther. Action Input: {'input': 'connection between Ms. Weems and Wednesday Addams'}
Graph Store Query:
MATCH (e1:Entity {id: "Ms. Weems"})-[:SPEAKS]->(e2:Entity {id: "Wednesday Addams"})
RETURN e1, e2
Graph Store Response: []
Observation: There is no direct connection between Ms. Weems and Wednesday Addams based on the given graph query response.
Thought: The 'text2cyther' tool did not provide any information about the connection between Ms. Weems and Wednesday Addams. I will try using the 'subgraph_retriver' tool to gather more information.
Action: subgraph_retriver. Action Input: {'input': 'Ms. Weems, Wednesday Addams'}
Observation: Ms. Weems is an ex-roommate of Morticia Addams. Wednesday Addams is the daughter of Morticia Addams.
Thought: Based on the information gathered, Ms. Weems is an ex-roommate of Morticia Addams, and Wednesday Addams is the daughter of Morticia Addams. Therefore, there is an indirect connection between Ms. Weems and Wednesday Addams through Morticia Addams.
Ms. Weems is an ex-roommate of Morticia Addams, who is the mother of Wednesday Addams.

```

Figure 6. Dialogue with subgraph retrieval.

C. Limitations and future work

The implementation demonstrated represents a basic version of the proposed conversational system framework integrated with a knowledge graph. To comprehensively assess the framework's capabilities and limitations, additional

experiments are required, involving more complex configurations and larger knowledge graph databases.

It is important to note that, apart from ChatGPT-3.5 [13], an experiment was conducted with the Zephyr 7b-alpha model[16]. Unfortunately, this model proved to be insufficiently powerful, leading to the generation of Cypher queries containing errors. This outcome suggests that smaller language models may not be suitable for the proposed framework, underscoring the need for more robust models.

Future directions for this research include the development of an interface for the conversational agent.

VI. CONCLUSION

Integrating knowledge graphs with xAI techniques significantly enhances the explainability of conversational agents. Through a systematic review and the proposal of a framework, this study demonstrates the potential of KGs in improving the transparency and interpretability of AI-driven conversational systems. The adaptable framework, validated by a simple setup, highlights the feasibility and effectiveness of leveraging KGs across different conversational systems architectures.

The search for relevant scientific papers across four major academic databases (Google Scholar, Semantic Scholar, ACM Digital Library and IEEE Xplore) highlighted a gap in the literature concerning the integration of KGs, conversational agents and Explainable AI all in one study. The analysis of selected articles revealed that most efforts in creating an xAI conversational agent with KG tend to rely on complex algorithms for extracting information from graphs. Examples include using attention-based methods to predict edges between entities or employing reinforcement learning agents for navigating through the graph. These techniques, while innovative, add layers of complexity to the system, making it harder for users to understand how it selects information from the graph. This results in a system that is less clear and transparent to its users.

Future research avenues include delving into conversational systems wherein agents autonomously interact with KGs. This area appears to be relatively untapped and holds considerable promise, particularly in more sophisticated scenarios where conversational agents are involved in a critical decision-making environment. Such contexts demand higher standards for the system's performance, necessitating not only accurate and efficient response generation but also ensuring that the underlying reasoning processes are transparent and comprehensible to users.

REFERENCES

- [1] Hepenstal, S., Neesha Kodagoda, Leishi Zhang, Pragya Paudyal and B. L. William Wong. “Algorithmic Transparency of Conversational Agents.” *IUI Workshops* (2019).
- [2] Liu, Zhibin, Zheng-Yu Niu, Hua Wu and Haifeng Wang. “Knowledge Aware Conversation Generation with Explainable Reasoning over Augmented Graphs.” *Conference on Empirical Methods in Natural Language Processing* (2019).
- [3] Moon, Seungwhan, Pararth Shah, Anuj Kumar and Rajen Subba. “OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs.” *Annual Meeting of the Association for Computational Linguistics* (2019).
- [4] Ma, Wenchang, Ryuichi Takanobu and Minlie Huang. “CR-Walker: Tree-Structured Graph Reasoning and Dialog Acts for Conversational Recommendation.” *Conference on Empirical Methods in Natural Language Processing* (2020).
- [5] Shi, Nuobei, Qin Zeng and Raymond S. T. Lee. “Xai Language Tutor – A Xai-Based Language Learning Chatbot Using Ontology and Transfer Learning Techniques.” *EduRN: Foreign Language (Topic)* (2020): n. pag.
- [6] Wen, Bingbing, Xiaoning Bu and Chirag Shah. “EGCR: Explanation Generation for Conversational Recommendation.” *ArXiv abs/2208.08035* (2022): n. pag.
- [7] Wilcock, Graham. “Generating More Intelligent Responses and Explanations with Conversational AI and Knowledge Graphs.” (2022).
- [8] Tian, Yijun, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, N. Chawla and Panpan Xu. “Graph Neural Prompting with Large Language Models.” *ArXiv abs/2309.15427* (2023): n. pag.
- [9] Zafar, Ahtsham, Venkatesh Balavadhani Parthasarathy, Chan Le Van, Saad Shahid, Afaq Khan and Arsalan Shahid. “Building Trust in Conversational AI: A Comprehensive Review and Solution Architecture for Explainable, Privacy-Aware Systems using LLMs and Knowledge Graph.” *ArXiv abs/2308.13534* (2023): n. pag.
- [10] Pan, Shirui, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang and Xindong Wu. “Unifying Large Language Models and Knowledge Graphs: A Roadmap.” *ArXiv abs/2306.08302* (2023): n. pag.
- [11] Wednesday (TV series). (2024, February 7). wikipedia.org. [https://en.wikipedia.org/wiki/Wednesday_\(TV_series\)](https://en.wikipedia.org/wiki/Wednesday_(TV_series))
- [12] Neo4j Graph Database & Analytics – The Leader in Graph Databases.
- [13] Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever and Dario Amodei. “Language Models are Few-Shot Learners.” *ArXiv abs/2005.14165* (2020): n. pag.
- [14] LlamaIndex. (2022). [Software] https://github.com/jerryliu/llama_index
- [15] Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever. “Language Models are Unsupervised Multitask Learners.” (2019).
- [16] Tunstall, Lewis, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush and Thomas Wolf. “Zephyr: Direct Distillation of LM Alignment.” *ArXiv abs/2310.16944* (2023): n. pag.