

XAgent: A Conversational XAI Agent Harnessing the Power of Large Language Models

Van Bach Nguyen¹, Jörg Schlötterer^{1,2} and Christin Seifert¹

¹University of Marburg, Germany

²University of Mannheim, Germany

Abstract

For humans, explanations are conversations, while explanations of models are one-directional result presentations. To make model explanations conversational, previous work presented a framework for conversational XAI agents. However, this early research prototype heavily relied on template-based natural language responses and a Jupyter Notebook user interface, limiting its application to a lab environment. In this work, we present XAgent, a conversational XAI agent addressing these limitations. We designed XAgent as Web application with a chat interface, harnessing the power of the Llama-2 language model to leverage the template-based design to natural conversations. The inclusion of LLAMA-2 further extends the scope of the agent to also answer questions beyond XAI, such as for example factual questions or questions about terminology and supports in prompting the user for additional information in case of incomplete requests. With these improvements, we lift the previous research prototype to a deployment-ready conversational XAI agent, bringing us closer to our goal of creating a real-world XAI agent that explains machine learning models in a conversational style. The source code for XAgent is available at <https://github.com/aix-group/XAGENT/>.

Keywords

Conversational XAI, XAI chatbot, Conversational Agent for XAI

1. Introduction

Miller [1] argues that human explanations are inherently conversational. After an initial explanation, the explainee can ask subsequent questions. An explanation is good and satisfactory, if all questions of the explainee are answered by the explainer, i.e., the explainee does not have follow up questions.

In the field of XAI, explanations are almost exclusively one-directional and usually one-shot, albeit Conversational XAI has gained increasing attention from the research community recently [2, 3, 4, 5]. Among the proposed solutions, our previous work [3] presents a system capable of answering a wide range of XAI-related questions introduced in the XAI question bank [6]. However, the system has limitations: (1) it is limited to a template-based Natural Language Generation (NLG) component, which lacks flexibility and provides a poor user experience, (2) it cannot answer questions requiring external knowledge, and (3) it is implemented

Late-breaking work, Demos and Doctoral Consortium, collocated with The 2nd World Conference on eXplainable Artificial Intelligence: 17-19 July, 2024, Malta, Valletta

✉ vanbach.nguyen@uni-marburg.de (V.B. Nguyen); joerg.schloetterer@uni-marburg.de (J. Schlötterer); christin.seifer@uni-marburg.de (C. Seifert)

 0000-0002-4576-9302 (V.B. Nguyen); 0000-0002-3678-0390 (J. Schlötterer); 0000-0002-6776-3868 (C. Seifert)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

in a Jupyter Notebook environment, thereby limiting the installation to a lab environment. To address these limitations, we introduce XAgent, a conversational agent using Large Language Models (LLMs) for generating answers. XAgent is based on the previously introduced XAI agent framework [3], and extends its components. Specifically, we augment the system by integrating Large Language Models to (1) enhance the NLG component, generating more flexible responses and (2) enable the system to answer questions about factual knowledge, such as the meaning of specific terminologies. Additionally, we (3) demonstrate a real-world system with a graphical user interface (GUI) and hosting capabilities, facilitating the deployment and accessibility of XAgent¹. XAgent is a deployment-ready conversational XAI agent that can explain machine learning models in a conversational manner.

2. System overview

The general architecture of XAgent is depicted in Figure 1. We built upon the architecture from our previous work [3] with four components.

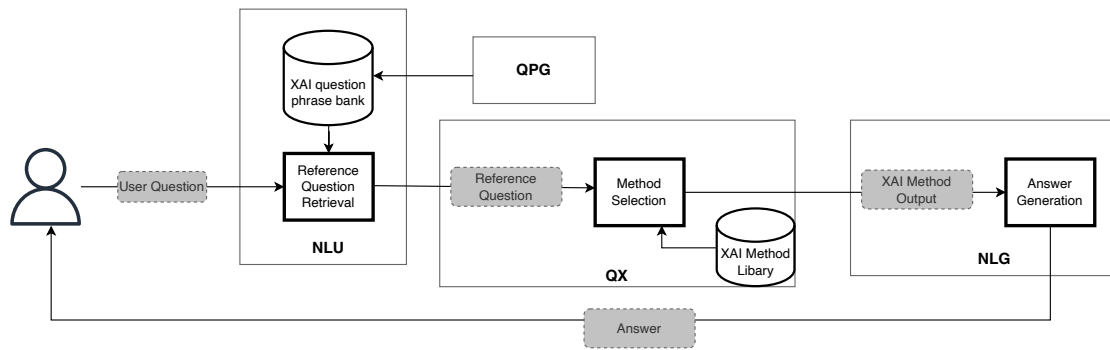


Figure 1: Incorporating XAI in conversational agents: An overview architecture, adapted from [3], with improved components highlighted in bold.

Question-Phrase-Generation (QPG): This component employs a paraphrase generation model on questions sourced from the XAI question bank [6]. The generated candidates are scored by multiple annotators and ranked, resulting in the XAI question phrase bank. This question phrase bank is used to train the NLU component. In this work, we adopt the question phrase bank for NLU without making any modifications.

Natural Language Understanding (NLU): This component is responsible for identifying the user’s actual intent from a wide range of XAI utterances. The NLU model is trained on the question phrase bank from QPG. *We extend this component by adding a feature that suggests alternative questions to choose in case NLU fails to identify the intent.*

Question-XAI (QX): This component takes the intent from NLU, maps it to a corresponding XAI method that can address the intent and then returns the resulting explanation of the chosen method (XAI method output). *In this work, we incorporate factual questions requiring external knowledge and employ a Large Language Model (LLM) to provide explanations.*

¹<https://github.com/aix-group/XAGENT/>

XAgent

Hello, I am XAgent, an AI model designed to provide explanations for predictions made by an AI model trained on the **German Credit dataset**. My role is to help you understand how the model arrived at a specific prediction and provide insights into the factors that influenced that prediction. **Assume you are a customer who wants a loan from a bank and provide your information.** An AI model will then predict whether you are able to **get a loan or not**. After that, you can ask questions about the predictions. Type anything when we shall start.

ok

The dataset has the following attributes:
[Age (Your age in years)] [Sex (Your gender)] [Job (The type of job you have)] [Housing (The type of housing you live in)] [Savings accounts (The amount of money you have saved in a savings account)] [Checking account (The amount of money you have in a checking account)] [Credit amount (The amount of credit in Euro you are requesting)] [Duration (The duration of the credit in months)] [Purpose (Purpose of the loan)]

We prepared the following sample instance of a customer:

Age	Sex	Job	Housing	Savings accounts	Checking account	Credit amount	Duration	Purpose
22	female	skilled	own	little	moderate	5951	48	radio/TV

If you like to continue with this instance, please type anything. Otherwise, if you like to input a different instance, please type the `/change instance`. You can change the instance anytime by the command.

What is up? >

(a) Introduction

/change instance

What is your age in years? Please provide a number.

50

Give me your Sex [male, female]. Please choose one of the following values.

none

The input value is not valid, please select one of the following options: male, female.

male

How about your current job? Please select one of the following options: [0 - unskilled and non-resident, 1 - unskilled and resident, 2 - skilled, 3 - highly skilled].

2

How about your Housing? Please choose one of the following values: [own, free, rent].

free

(b) Information Collection

Figure 2: Input Phase: The phase begins with the introduction of the conversation alongside an example instance of the dataset, as depicted in Figure (a). Figure (b) illustrates the process to specify a desired input instance, different from the provide examples.

Natural Language Generation (NLG): This component converts the output of the XAI method (e.g., a table, graph, or number) to a natural language answer. *We integrate LLMs in the final step of the textual answer generation in the NLG component to improve the answer's text quality from a pure template-driven answer to natural language.*

3. Implementation

A conversation in XAgent consists of multiple conversation steps and each conversation step consists of two phases: the Input Phase and the Explanation Phase. The Input Phase refers to the process of data collection from the user. Once all relevant data has been successfully gathered, the system transitions to the Explanation Phase, wherein the agent explains the model and its prediction to the user.

3.1. Input Phase

Before users can inquire about the model and request explanations for the predictions, they must first provide the necessary information about the instance to be predicted. We refer to this as the Input Phase. In this phase, we introduce the agent to the user with an example instance, but also provide the option for the user to change the instance by using the command `/change instance`. The introduction is illustrated in Figure 2a. For this demonstration, we

use the German-credit dataset² as a fixed dataset for interaction. However, technically, we can change the dataset by typing `/change dataset`. We also provide a pre-trained Random Forest classifier with the accuracy of 85% on this dataset. If the user chooses to provide their own instance, we implement a series of questions to gather the required information. Each question is accompanied by a constraint to ensure that the user provides valid input, such as selecting from predefined categories for categorical features or entering a numeric value for numerical features. If the user provides unexpected input, they are prompted to repeat the entry until it meets the specified requirements. Once all the necessary information has been collected, the system uses a pre-trained classifier to generate the prediction results, which are then presented to the user. All of the information of the instance and the prediction for it is stored in the memory. Importantly, the questions and answers provided by the agent during this input phase are enhanced using a LLAMA-2 [7] to ensure more flexible responses instead of fixed templates. Further details can be found in Section 3.2.4. An example of the Input Process is illustrated in Figure 2b.

3.2. Explanation Phase

Once the prediction of an instance has been presented to the user, in the subsequent phase, users may ask questions about the prediction, the model and the instance. This marks the initiation of the Explanation Phase, which is illustrated in Figure 3. Specifically, a user's question is processed by the NLU component. Here, a sentence embedding model matches the user's question to a corresponding question within the system's question phrase bank. The selected question, along with the information extracted from the user's question, is then processed by the Dialog Manager. Concurrently, the extracted information is stored in memory, and the preprocessed question is passed to the QX component. This component retrieves information from a corresponding XAI method capable of answering the question. The stored information is requested and received through the Dialog Manager. Once the Dialog Manager obtains the XAI method's information from the QX, it delegates the task of plugging this information into predefined templates to the NLG component. The templated answers are then returned to the LLM (LLAMA-2) for language improvement before being presented to the user. An example of explanation observed can be seen in Figure 4a. This section provides detailed descriptions of each mentioned component.

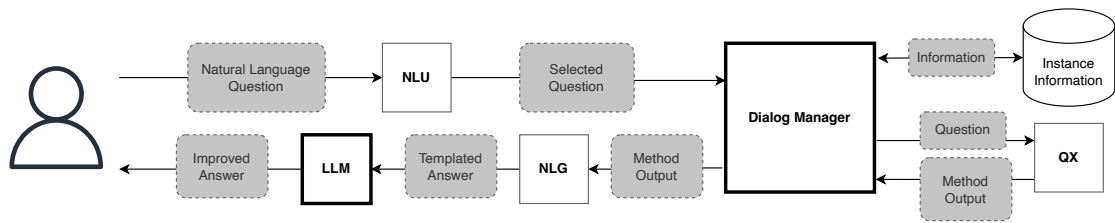


Figure 3: Illustration of the explanation phase process flow. New components are highlighted in bold.

²<https://www.kaggle.com/datasets/uciml/german-credit>

I recorded the information: [18.0,male,0,own,rich,little,5000.0,12.0,car]. You will get the loan You can ask questions about the machine learning model, such as: Why was the prediction made? Why was Y not predicted? What changes would make a prediction for Y? Please type your question.

Why was the prediction made?

The graph below highlights key factors that can impact your loan application. Red features enhance your chances of approval, while blue features may hinder it.



Which algorithm was used?

The Random Forest model was employed as an ensemble learning algorithm, comprising a collection of decision trees for classification and regression tasks.

How accurate of the prediction?

The accuracy of the predictions made by the Random Forest model is roughly 85%. Please let me know if you need any further assistance.

what is the definition of "accuracy"?

Accuracy refers to the degree of closeness of a model's predictions to the actual true values. It is a measure of how well a model estimates the target variable. Accuracy is calculated as the ratio of correctly predicted instances to the total number of instances in the test dataset. A higher accuracy indicates better performance, while a lower accuracy indicates poorer performance.

What has happened?

Sorry, I am not exactly sure whether I understood your question correctly. Did you mean any of the below? Please select a number or type a new question:

1. How is this instance given this prediction?
2. Why is this instance predicted loan accepted instead of ['loan rejected']?
3. What kind of instance gets this prediction?
4. What are the top features it uses?
5. What features of this instance lead to the system's prediction?
6. See more questions

6

Here are some example questions. Please select a number or type a new question:

6. What kind of instance gets a different result?
7. How should this instance change to get a different prediction?
8. In what conditions will this prediction be different?
9. How should a feature change for this instance to get a different result?
10. How does input {Age, Sex, ...} impact the model?
11. Which feature is given importance?
12. Which attributes does the model use?
13. How high do feature values have to be to still be predicted the same thing?

How should Saving accounts and Purpose change for this instance to get a different result?

The savings accounts should be changed to little and the purpose should be changed to education to avoid being denied a loan.

(a) Explanation Process

(b) Questions Suggestion

Figure 4: Explanation Phase: The phase starts after the agent gathers all necessary information from the instance, and the user transitions to seeking clarification or interpretation of the prediction results (Figure a). In scenarios where the Natural Language Understanding (NLU) component exhibits low confidence in comprehending the user's questions, it proactively suggests a list of potential clarifying questions for the user to select (Figure b).

3.2.1. Natural Language Understanding (NLU)

We use the NLU component from our previous work [3]. In summary, we use the XAI question phrase bank within the QPG component to train the NLU by framing the task of matching user questions to reference questions as multi-class classification. Sentence embeddings are generated for user and reference questions using SimCSE [8] and RoBERTa-large pre-trained model [9]. Then, a feed-forward neural network classifies user questions into reference questions. The output identifies the reference question that best fits the user's intent. If the probability falls below a threshold, θ , the question is considered an unknown variation or paraphrase.

In such scenarios, our preceding system declines to provide an answer or a suggestion to the question and the conversation halts. In this work, we enhance the NLU components by presenting the user with a curated list of questions for selection. The order of questions in this list is based on their cosine similarity values within the question bank. Initially, the list presents

the top 5 questions with the highest similarity, with subsequent options provided if the desired question is not among the initial selection. Moreover, users are provided with the opportunity to rephrase their questions. An example of this question selection case is depicted in Figure 4b.

3.2.2. Dialog Manager

The Dialog Manager (DM) component serves as the central processing unit, orchestrating information processing and decision-making within the system. Specifically, within the DM, we define four states corresponding to four situations that the agent might encounter: the Input state, Explanation state, Suggestion state, and Missing Data state. The Input and Explanation states correspond to the Input and Explanation phases, while the Suggestion and Missing Data states represent two situations that may arise during the Explanation Phase. The dialog begins with the Input state. Once all information has been collected, the DM transitions the state to Explanation and maintains this state until another event occurs, such as missing data or failing to comprehend the user’s question. In cases of missing data, the DM sets the state to Missing Data and sends a question to the user to request the necessary information. For instance, if a user asks *How does this feature impact its prediction?* without explicitly specifying the feature, the DM will ask the user to provide the name of the feature. For situations where the question is unclear, the DM sets the current state to the Suggestion stage, in which the NLU returns a list of suggested questions.

The DM also functions as a component for information extraction and storage within the system’s memory. This facilitates the retrieval and utilization of information as needed. Specifically, during the input phase, the DM stores each user response to a question in a format compatible with the predefined classifier. Subsequently, it returns the classifier’s results.

3.2.3. Question-XAI (QX)

For the QX component, we employ the mapping established in the original work [3]. When encountering factual questions, such as *What does {ML terminology} mean?*, we use LLAMA-2 to generate the answer. Other types of questions can be integrated into the framework by providing either an answer template and/or an XAI method (alongside with an answer template) to answer that question. Each question type is handled through a distinct implementation. For questions necessitating specification of a class or instance, a fill-in template is employed. Once the template is completed, the system can further process the question. The filling process is facilitated by the Dialog Manager component, operating in the Missing Data state, which iteratively prompts the user until the necessary information is obtained to complete the template. Moreover, for factual questions about terminology, the system utilizes LLAMA-2 to provide the answer by prompting the following question to the Large Language Model (LLM):

Question: {Question}. Explain it in 100 tokens.

It should be noted that the identification of factual questions is performed by the NLU component, as there exists a general question, *What does [ML terminology] mean?* If the NLU matches the user’s question to this query, the QX will activate LLAMA-2 to respond to the user’s inquiry. Actually, the terminology in question is not limited to ML terminology but can encompass any term. We acknowledge the well-known issue of hallucinations, but favor conversation flow

over precision in such questions. That is, we favor an inaccurate answer over getting no answer at all.

3.2.4. Natural Language Generation (NLG)

In the context of the NLG component, each dataset possesses its own set of templates, which are defined within a JSON file. To accommodate a new dataset, a corresponding new template is required. This template encapsulates information about the features, as well as the responses for each question and case. The final response, derived from the template, is subsequently processed through the LLAMA-2 model using the specific prompt below:

```
Improve the language of the text but keep the original intent of the below
text. Do not add or omit any information, only adapt the language. Please keep
the information in square brackets unchanged. Only return the text without the
double quotes. Do not ask for anything else. Do not add any comments.
{Text}.
Improved text:
```

3.3. User Interface

We use Streamlit³ as the framework for both the user interface (UI) and server deployment. To address scenarios involving concurrent interactions from multiple users with the agent, we implement session handling. Specifically, we initialize all components only once except for the Dialog Manager. For each user session, a new agent object is created, equipped with its own Dialog Manager initialized with default states. This approach ensures that each user interacts with their respective agent object, leveraging the shared NLU, LLAMA-2, QX, and NLG components.

4. Discussion and Conclusion

Figure 2 and Figure 4 show screenshots of a conversation from the web application of XAgent. In Figure 4a, LLAMA-2 enhances all answers, thereby increasing their flexibility with each user-agent interaction. Additionally, LLAMA-2 generates the answer to the question *what is the definition of 'accuracy'?*. Generally, LLAMA-2 demonstrates benefits in terms of language improvement and answering factual knowledge questions. However, it also has its own shortcomings. For example, in Figure 2b, LLAMA-2 generates “Give me your Sex [male, female]. Please choose one of the following values,” which is unnatural as the list of values should appear after the request, not before. Additionally, in the same figure, we observe inconsistencies in the presentation of the list of values, as they sometimes appear in square brackets and sometimes in parentheses.

In summary, by integrating LLAMA-2 and replacing the Jupyter Notebook environment by the Streamlit framework, we lift the previous strongly constrained research prototype to a deployment-ready conversational XAI agent. Future work will involve tuning the prompts for LLAMA-2 and fine-tuning the conversation model based on collected log data. Furthermore,

³<https://streamlit.io/>, accessed April 2024

a promising direction for future exploration involves extending the application to other data types, particularly visual data, and utilizing multimodal architectures like Vision-Language models for explanation.

References

- [1] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *Artificial Intelligence* 267 (2019) 1–38. URL: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>. doi:10.1016/j.artint.2018.07.007.
- [2] D. Slack, S. Krishna, H. Lakkaraju, S. Singh, Explaining machine learning models with interactive natural language conversations using talktomodel, *Nature Machine Intelligence* 5 (2023) 873–883.
- [3] V. B. Nguyen, J. Schlötterer, C. Seifert, From Black Boxes to Conversations: Incorporating XAI in a Conversational Agent, in: L. Longo (Ed.), *Explainable Artificial Intelligence*, Springer Nature Switzerland, Cham, 2023, pp. 71–96.
- [4] C. Werner, Explainable AI through Rule-based Interactive Conversation, in: *EDBT/ICDT Workshops*, 2020.
- [5] M. Kuźba, P. Biecek, What would you ask the machine learning model? identification of user needs for model explanations based on human-model conversations, in: *ECML PKDD 2020*, Springer, 2020.
- [6] Q. V. Liao, D. Gruen, S. Miller, Questioning the AI: Informing Design Practices for Explainable AI User Experiences, in: *Proc. CHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, 2020, p. 1–15. doi:10.1145/3313831.3376590.
- [7] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. [arXiv:2307.09288](https://arxiv.org/abs/2307.09288).
- [8] T. Gao, X. Yao, D. Chen, SimCSE: Simple Contrastive Learning of Sentence Embeddings, in: *EMNLP, ACL, 2021*, pp. 6894–6910. URL: <https://aclanthology.org/2021.emnlp-main.552>. doi:10.18653/v1/2021.emnlp-main.552.
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) [cs].