KGChat User Manual

1 Introduction

KGChat (Knowledge Graph Chat) is a chatbot designed to empower users to query and interact with knowledge graphs derived from their own event logs. With KGChat, users can uncover and analyze business processes within their data, gaining valuable insights through an intuitive and conversational interface. The application utilizes Large Language Models (LLMs) to process user queries and generate responses. It employs Graph Neural Network (GNN) to process and comprehend the structure of the knowledge graph. Additionally, it leverages Graph Retrieval-Augmented Generation (RAG) to ensure that contextual information is incorporated in order to generate accurate and relevant answers.

2 Getting Started

You have the option to just compile and start a Docker container or create an environment and run the website directly in there. There are a couple of config arguments that are important when starting the program. For those have a look at the Configuration section.

2.1 Requirements

The locally installed NVIDIA CUDA driver version has to match the one in the `Dockerfile`, when using Docker, and otherwise look at it when installing CUDA. Make sure you read the whole *Getting Started* section before trying it yourself to make sure you understood everything. Depending on the configuration, you either need a `huggingface` token or an `openai-api-key`. Both have their own config file in the `webapp/src/utils/` folder.

2.2 Usage with Docker

To use Docker for the execution, first make sure you have Docker installed and the service is running:

1. Navigate to the `webapp` directory:

   ```
   $ cd webapp
   ```

2. Compile the container:

   ```
   $ docker build -t kgchat .
   ```

3. Start the container (the port opening is important to access the webpage):

   ```
   $ docker run -p 5000:5000 kgchat
   ```

2.3 Usage without Docker

If you just want to start without Docker, you first have to create an environment to run it in:

1. Navigate to the `webapp` directory:

   ```
   $ cd webapp
   ```

2. Create and activate a conda environment from the `environment.yml`:

   ```
   $ conda create -f environment.yml -n g-docker
   $ conda activate g-docker
   ```

3. Install CUDA driver and replace `XX` at the end with the version you have installed (for example: v.11.8 = 118):

```
$ nvcc −−version # shows you the CUDA version
    that is installed (if one is installed)
$ conda run −n g−docker pip install torch
    torchvision torchaudio −−index−url
    https://download.pytorch.org/whl/cuXX
```

4. Start the app:

```
$ python app.py
```

3 Configuration

The app can be started with different arguments. These have to be changed in the `config.txt` inside the `webapp` folder. The options are:

3.1 `--host`

- Regulates how to access the website
- Expects an IP address
- Defaults to `127.0.0.1` for local use only
- Use `0.0.0.0` for access through the host machine's IP address if the router forwards the ports 80-¿5000 for the host's IP

3.2 `--mode`

- Changes which LLM type is used
- Expects either `local` (default) or `remote`
  - `local`: Uses a Huggingface Model which will be downloaded and run locally on your machine
    - This requires a lot of power
    - A Huggingface Access Token is required in the corresponding file in `/webapp/src/utils/HUGGINGFACE_TOKEN.txt`
    - Ensure the token allows access to repos that are not yours
    - Ensure you have requested (and been granted) access to the Model you want to use

  - `remote`: Uses the ChatGPT API to run the LLM remotely
    - A ChatGPT API Key is required in the corresponding file in `/webapp/src/utils/OPENAI_API_KEY.txt`
    - Ensure you have topped up the balance, otherwise, the request will return an error

3.3 `--llm_model`

- Changes the model/repo to clone from Huggingface for the `local` use case
- Defaults to `meta-llama/Llama-2-7b-chat-hf`
- Ensure it is a text-based model
- Ensure you have enough resources to execute the model

3.4 `--gpt_model`

- Changes the model which the ChatGPT API uses for the `remote` use case
- Defaults to `gpt-3.5-turbo-16k`
- Be aware: the newer the model, the higher the cost per token
- Only use text generation models

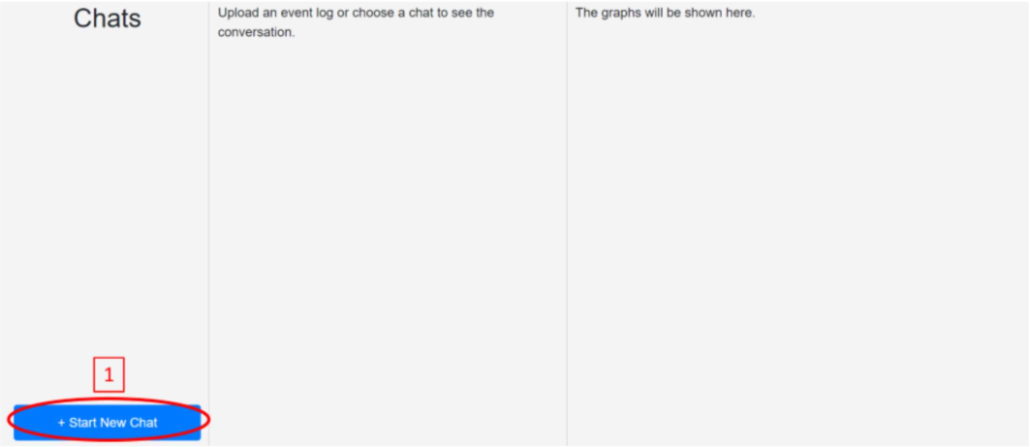
3.5 `--debug`

- Changes whether there are debug outputs in the console
- If the flag is included, the debug messages are shown

4 User Interface and Features

4.1 Importing Event Log

After launching the application on the chosen port and accessing it on a web browser, you will arrive on the landing page. By clicking on the `Start New Chat` button, you can import the event log to be analyzed from your file system. It is required that the event log to be in XES format, otherwise an error will be thrown.



Chats

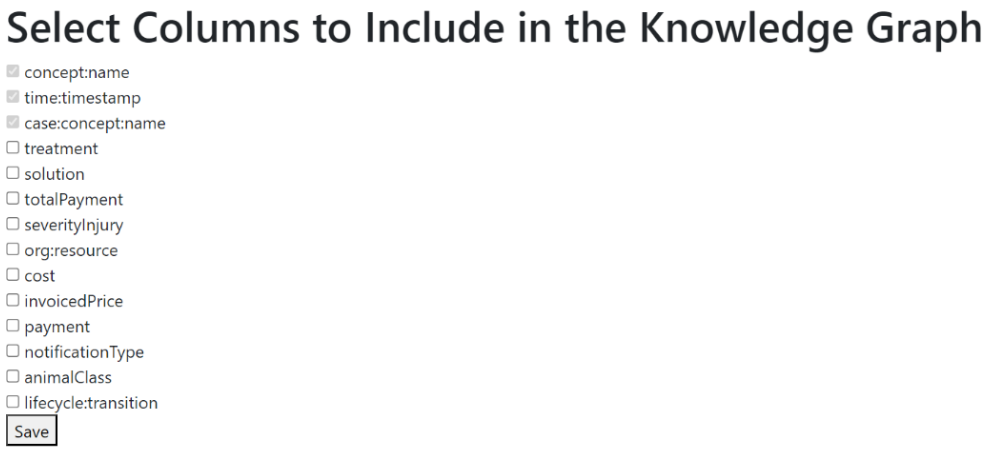
Upload an event log or choose a chat to see the conversation.

The graphs will be shown here.

Figure 1: Importing Event Log

4.2 Selecting Attributes / Columns

Once the event log is successfully imported, you will be prompted to choose the attributes, or more specifically, columns from your event log that should be included for generating the knowledge graph and be analyzed. The three attributes *concept:name*, *time:timestamp* and *case:concept:name*, which correspond to the activity name, timestamp and case ID, are selected by default since they are essential for process discovery. To pick an attribute, click on the box next to the attribute name. Once you are done, click on the `Save` button on the bottom of the page to proceed. It should be noted that at least one attribute other than the three mandatory columns has to be selected.

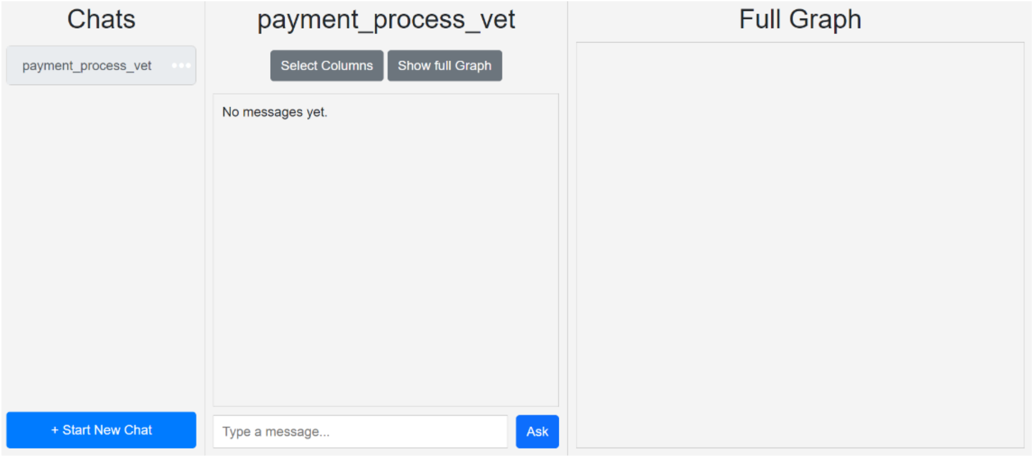


Select Columns to Include in the Knowledge Graph

☑ concept:name
☑ time:timestamp
☑ case:concept:name
☐ treatment
☐ solution
☐ totalPayment
☐ severityInjury
☐ org:resource
☐ cost
☐ invoicedPrice
☐ payment
☐ notificationType
☐ animalClass
☐ lifecycle:transition

Figure 2: Selecting Attributes / Columns

If everything runs smoothly, you will be redirected to the main page. You will find a chat session is created below the section `Chats`. By default, the chat session is given a name similar to your event log.



No messages yet.

Figure 3: Chat Session Created

4.3 Renaming and Deleting Chat Session

At any time, it is possible to rename or delete a chat session. Click on the button . . . beside a chat session and you will find the buttons for renaming and deleting it.

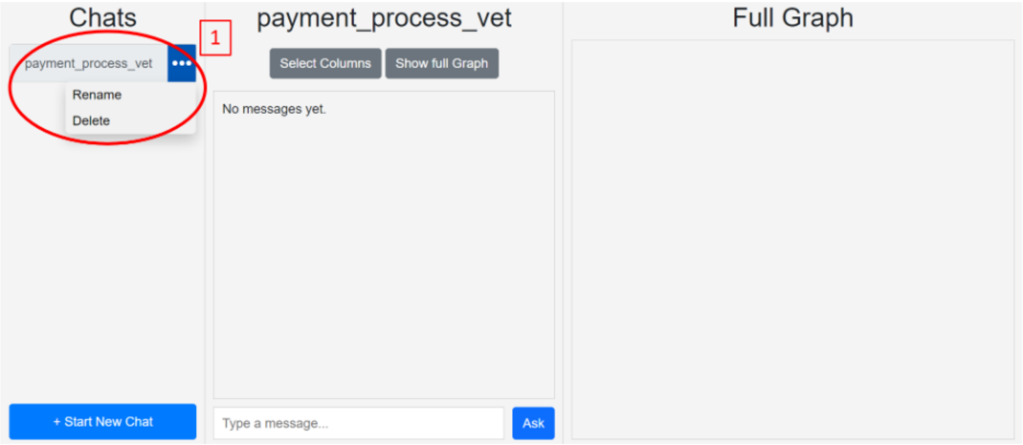


Figure 4: Renaming and Deleting Chat Session

4.4 Changing Attributes / Columns

The user is not restricted by the initial choice of the columns considered for analysis. It is also possible to select new columns or removing columns which were selected previously. This can be done by clicking on the `Select Columns` button.

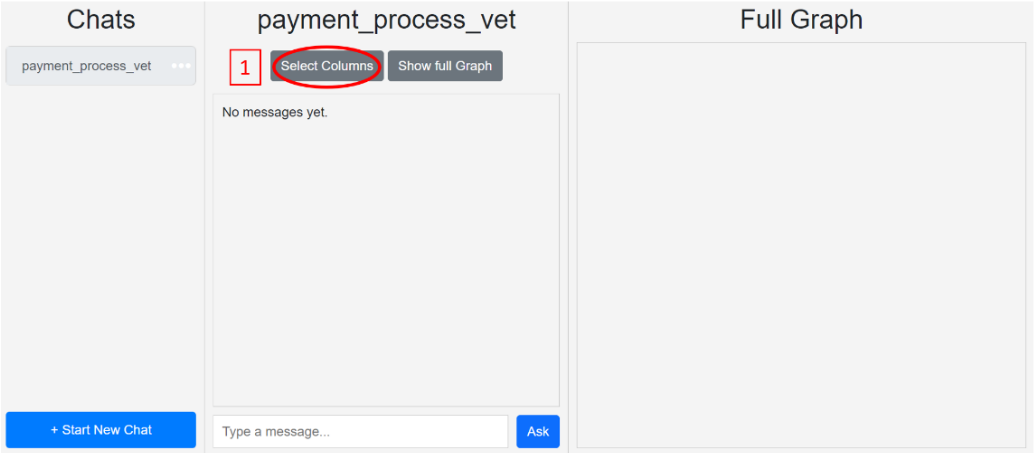
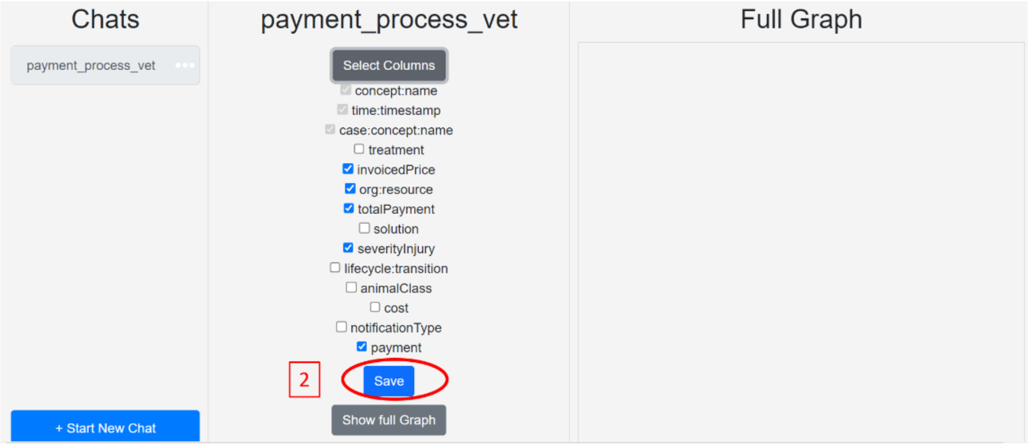


Figure 5: Changing Attributes / Columns

To add new columns, click on the button beside the columns' names. To remove a previously selected column, uncheck the box beside the column's name. At the end, click on the `Save` button to save the new selection of columns.



Chats

payment_process_vet

+ Start New Chat

payment_process_vet

Select Columns

☑ concept:name
☑ time:timestamp
☑ case:concept:name
☐ treatment
☑ invoicedPrice
☑ org:resource
☑ totalPayment
☐ solution
☑ severityInjury
☐ lifecycle:transition
☐ animalClass
☐ cost
☐ notificationType
☑ payment

Save

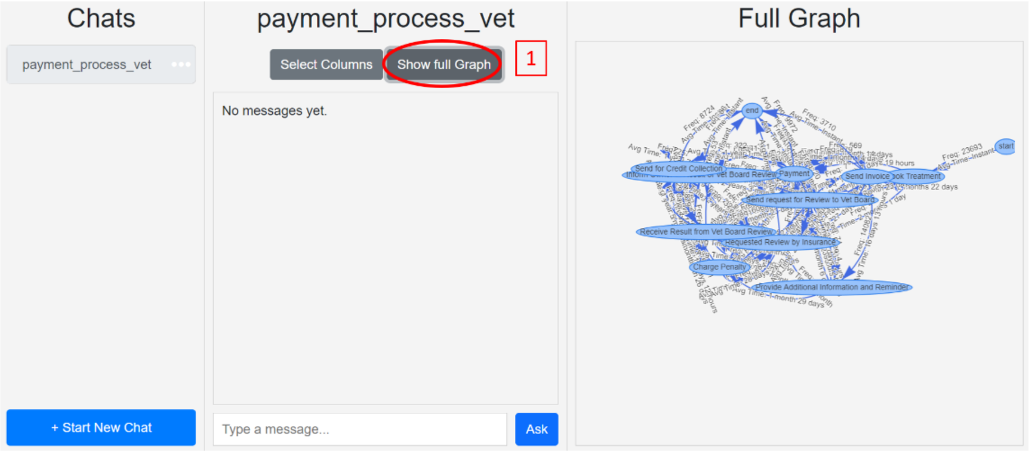
Show full Graph

Full Graph

Figure 6: Adding or Removing Columns

4.5 Displaying the Full Knowledge Graph

To see the complete knowledge graph generated from the given event log, click on the `Show Full Graph` button. The knowledge graph will then be displayed on the rightmost section of the web page.



Chats

payment_process_vet

+ Start New Chat

payment_process_vet

Select Columns

Show full Graph

No messages yet.

Type a message...

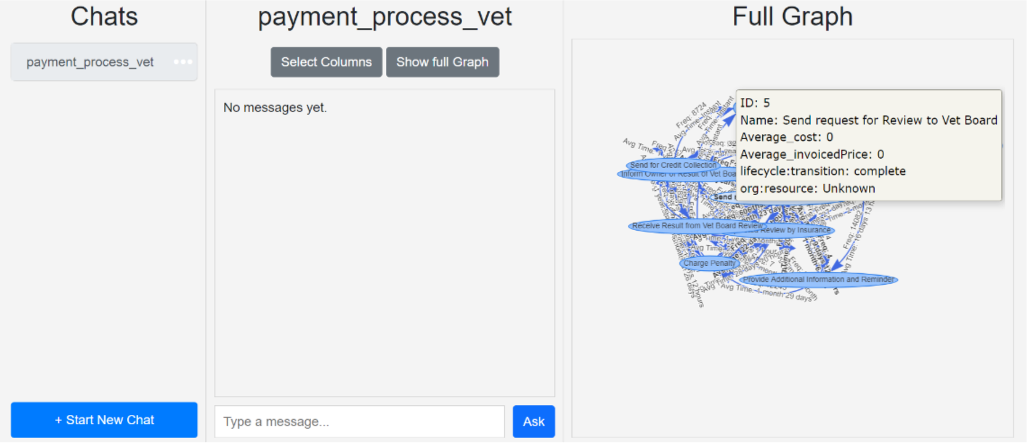
Ask

Full Graph

end
start
Send for Credit Collection
Vet Board Review Payment
Send Invoice ...ok Treatment
Send request for Review to Vet Board
Receive Result from Vet Board Review
Requested Review by Insurance
Charge Penalty
Provide Additional Information and Reminder
Freq: 23693
Freq: 3710
Freq: 8724

Figure 7: Displaying the Full Knowledge Graph

The displayed graph is not a static image, so it is possible to interact with it by clicking on the nodes or edges and moving your mouse around. On top of that, by hovering above a node in the graph, information regarding the node will be shown.



No messages yet.

ID: 5
Name: Send request for Review to Vet Board
Average_cost: 0
Average_invoicedPrice: 0
lifecycle:transition: complete
org:resource: Unknown

Figure 8: Interactive Knowledge Graph

4.6 Chatting with the ChatBot

To pose a query regarding your process, click on the white rectangular column and type it out. Once you are done, click on the `Ask` button on the right and the application will process your question. Our application does not only support processing questions in English but also in other languages. We have tried posing questions in German, Chinese, Spanish and Korean, and have received satisfied answers.

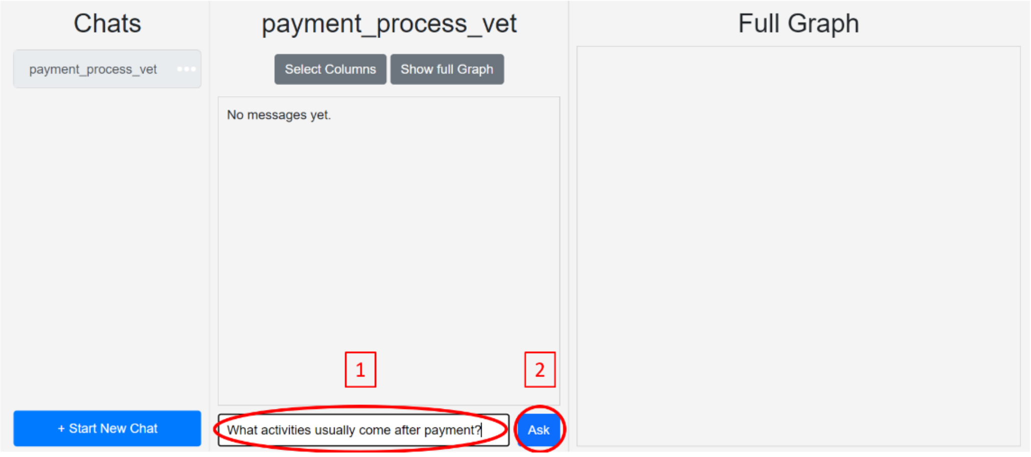
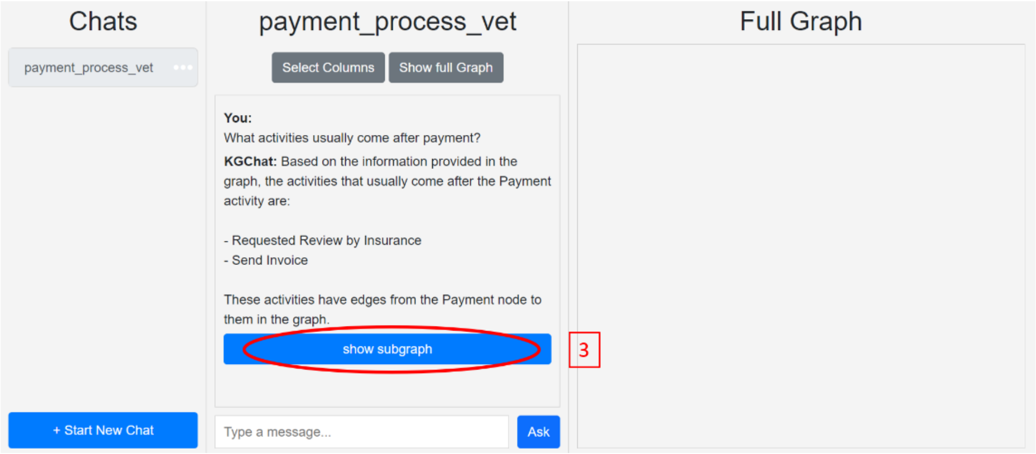


Figure 9: Chatting with the ChatBot

The questions and the corresponding generated answers will be displayed above the text box. To see the subgraph which is contextually relevant to the question, click on the button `show subgraph` below the conversation. For every question, there will be a corresponding contextually-relevant subgraph generated and it can be accessed at any time with the aforementioned button.



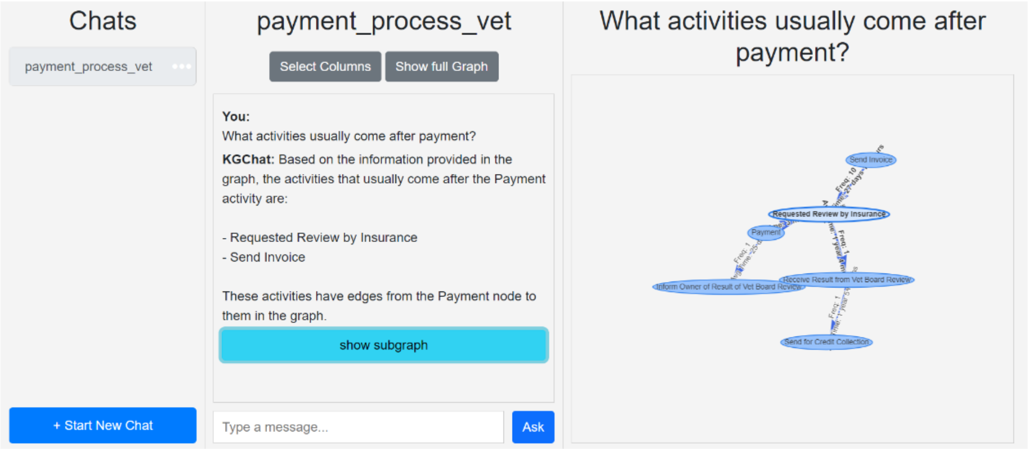
You:
What activities usually come after payment?
KGChat: Based on the information provided in the graph, the activities that usually come after the Payment activity are:

- Requested Review by Insurance
- Send Invoice

These activities have edges from the Payment node to them in the graph.

Figure 10: Viewing Subgraph

After clicking on the `show subgraph` button, the subgraph will be displayed on the right side of the conversation section. Similar to the full graph, the subgraph is also interactive. Above the subgraph is the corresponding user's question that is contextually related to the subgraph.



Chats
payment_process_vet
+ Start New Chat
payment_process_vet
Select Columns
Show full Graph
show subgraph
Type a message...
Ask
What activities usually come after payment?
Send Invoice
Freq: 10
Requested Review by Insurance
Payment
Freq: 1
Freq: 1
Inform Owner of Result of Vet Board Review
Receive Result from Vet Board Review
Freq: 1
Send for Credit Collection

You:
What activities usually come after payment?

KGChat: Based on the information provided in the graph, the activities that usually come after the Payment activity are:

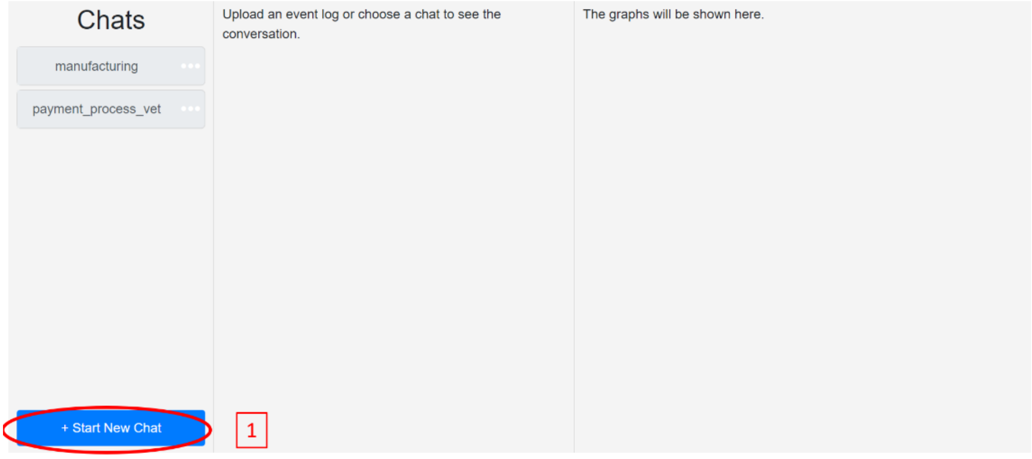
- Requested Review by Insurance
- Send Invoice

These activities have edges from the Payment node to them in the graph.

Figure 11: Interactive Subgraph

4.7 Starting New Chat Session

To start on a new chat session, click on the `Start New Chat` button and the following procedure is the same as importing a new event log. A new chat session will then be created and is accessible within the Chats section of the web page.



Upload an event log or choose a chat to see the conversation.

The graphs will be shown here.

Figure 12: Starting New Chat Session

5 Troubleshooting

5.1 Common Issues and Solutions

5.1.1 Issue 1: Error loading .xes files

- **Description**: The application fails to load event logs in formats other than .xes.
- **Solution**:
  - Ensure the event log is in .xes format.
  - Check that the .xes file is not corrupt and is properly formatted.

5.1.2 Issue 2: Application crashes when generating subgraphs

- **Description**: The application becomes unresponsive or crashes during the generation of subgraphs.
- **Solution**:
  - Ensure your system meets the minimum requirements and has sufficient memory.
  - Reduce the number of selected columns to minimize processing load.
  - Try using a smaller event log.

5.1.3 Issue 3: API Key Not Found

- **Description**: Receiving an error that the OpenAI or Huggingface API key is not found.
- **Solution**:
  - Ensure that your API keys are correctly placed in the respective files located in `webapp/src/utils/`.
  - For OpenAI, use `OPENAI_API_KEY.txt`.
  - For Huggingface, use `HUGGINGFACE_TOKEN.txt`.

5.1.4 Issue 4: Insufficient API Balance

- **Description**: Receiving an error indicating that the API request failed due to insufficient balance.
- **Solution**:
  - Ensure that your OpenAI API balance is topped up.

5.2 System Errors

5.2.1 Error: "This model's maximum length is 16385 tokens. However, you requested XXXXX tokens"

- **Description**: This error occurs when the combined length of the prompt and the expected response exceeds the model's limit.
- **Solution**:
  - Shorten the questions or context provided.
  - Summarize previous interactions to fit within the token limit.
  - Ensure that the maximum tokens for responses are set appropriately.
  - Start a new conversation.

5.2.2 Installing the dependencies

Error: Cannot Import Name 'triu' from 'scipy.linalg' This issue is related to the version of SciPy (the `scipy.linalg` functions `tri`, `triu` & `tril` are deprecated and will be removed in SciPy 1.13). To solve it:

1. Uninstall the current version of SciPy:

```
$ pip uninstall scipy
```

2. Install a compatible version of SciPy. For Python versions 3.8 to 3.11, you can use SciPy 1.10.1, for Python 3.12 you can use SciPy 1.11.2:

```
$ pip install scipy==1.11.2
```

Error: Long Path Names on Windows Enable support for long paths in Windows:

1. Press `Win + R`, type `regedit`, and press Enter.
2. Navigate to `HKEY_LOCAL_MACHINE`.
3. Find the `LongPathsEnabled` value. If it does not exist, create it as a new `DWORD (32-bit)` value.
4. Set its value to 1.

6 Frequently Asked Questions (FAQ)

6.1 How can I change the configurations for the LLM models?

Configurations can be adjusted in the `config.txt` file inside the `webapp` folder. You can change settings like the LLM model, GPT model, and debugging options.

6.2 How do I configure the API keys for OpenAI and Huggingface?

Place your API keys in the corresponding files located in `webapp/src/utils/`.
For OpenAI, use `OPENAI_API_KEY.txt`, and for Huggingface, use `HUGGINGFACE_TOKEN.txt`.

6.3 How do I select or change the attributes for my knowledge graph?

After importing the event log, you can select the attributes by clicking on the `Select Columns` button. To change the attributes later, click on the same button and update your selection.

6.4 How can I change the language of the chatbot responses?

KGChat supports multiple languages. You can pose questions in your preferred language, and the application will process and respond accordingly.

6.5 How do I handle large event logs that slow down processing?

Reduce the size of the event log or select fewer attributes for analysis. Ensure your system has adequate resources to handle large files.

7 Glossary and Index

7.1 Large Language Models (LLMs)

"Large language models (LLMs) are a category of foundation models trained on immense amounts of data making them capable of understanding and generating natural language and other types of content to perform a wide range of tasks." [1]

7.2 Graph Neural Networks (GNNs)

"Graph neural networks apply the predictive power of deep learning to rich data structures that depict objects and their relationships as points connected by lines in a graph. In GNNs, data points are called nodes, which are linked by lines — called edges — with elements expressed mathematically so machine learning algorithms can make useful predictions at the level of nodes, edges or entire graphs." [2]
An article [3] by Sanchez-Lengeling and his team is also a good starting point for understanding GNNs for beginners.

7.3 Knowledge Graph

"A knowledge graph, also known as a semantic network, represents a network of real-world entities—such as objects, events, situations or concepts—and illustrates the relationship between them." [4] It is made out of three main components: nodes, edges and labels. Every edge between a pair of nodes is labelled to denote the relationship between them.

7.4 Retrieval Augmented Generation (RAG)

"Retrieval-Augmented Generation (RAG) is the process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating a response." [5]

7.5 Graph Retrieval Augmented Generation (Graph RAG)

“Retrieval-Augmented Generation (RAG) is a technique to search for information based on a user query and provide the results as reference for an AI answer to be generated. GraphRAG uses LLM-generated knowledge graphs to provide substantial improvements in question-and-answer performance when conducting document analysis of complex information.” [6] The technology behind Graph RAG utilizes knowledge graphs as a source of context or factual information for more accurate and contextual answers. [7]

References

[1] IBM. What are Large Language Models (LLMs)? Retrieved June 25, 2024, from `https://www.ibm.com/topics/large-language-models`

[2] Merritt, R. (2022, October 24). What Are Graph Neural Networks? Retrieved June 25, 2024, from `https://blogs.nvidia.com/blog/what-are-graph-neural-networks/`

[3] Sanchez-Lengeling, et al., ”A Gentle Introduction to Graph Neural Networks”, Distill, 2021.

[4] IBM. What is a knowledge graph? Retrieved June 25, 2024, from `https://www.ibm.com/topics/knowledge-graph`

[5] Amazon. What is RAG (Retrieval-Augmented Generation)? Retrieved June 25, 2024, from `https://aws.amazon.com/what-is/retrieval-augmented-generation/`

[6] Larson, J., & Truitt, S. (2024, February 13). GraphRAG: Unlocking LLM discovery on Narrative private data. Retrieved June 25, 2024, from `https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/`

[7] ontotext. What is Graph RAG? Retrieved June 25, 2024, from `https://www.ontotext.com/knowledgehub/fundamentals/what-is-graph-rag/`