Swisscom project <teamName>

Tornike Onoprishvili, Riccardo Sacco, Carla Lopez Zurita, Roberts Kalvitis, Michele Smaldone

1. OBJECTIVE

Sam is a retrieval augmented chatbot for supporting Swisscom customers via question answering and limited agentic capabilities. Given modern AI capabilities, Sam has a potential to reduce operating costs and improve customer experience. In this report, we discuss the main limitations of Sam, and details on our methods that overcome these restrictions.

First, Sam frequently responds in German when the customer's request is obviously in English. Second, Sam often provides irrelevant or unhelpful URLs, especially for technical questions (broken modem). Finally, in our opinion, the terse responses from Sam might discourage customers from seeking chatbot's help, and instead escalate the chat to an operator. We address all these points in our improved approach.

Our goal is to create an effective chatbot that can assist users effectively across a broad range of questions. The chatbot should:

- Detect language and apply it to conversation correctly;
- Give users useful responses that are based on Swisscom's publicly available data on their website;
- Filter out irrelevant user questions, without sounding too harsh;

2. APPROACH

Figure 1: Workflow of the chat

Initially we iterated through multiple chatbot versions in a rapid succession. We settled for the LangChain framework to handle data preparation, retrieval and composing the response.

Data preparation is split into three steps - splitting documents into overlapping chunks, storing chunks embeddings, and building a vector database. A chunk size of 2000 characters, with a 10% overlap was used. OpenAI text-embedding-3-large was used as the embedding model. We used Chroma for retrieval.

The data retrieval was improved in several steps. We improved the RAG retrieval process by chaining two search methods, similarity search and BM25. First, we chose K=30 for top K document retrieval using similarity search. Afterwards, we apply a reranking step with the BM25 algorithm with top K=3 that summarises the retrieved documents and keeps only the relevant information. We

found that this implementation made the response language specific without the need to add any additional parameters or filters.

OpenAI's GPT-4o model was used as a core LLM for the generation step. We evaluated the model on our own dataset to produce the best working representation of the AI assistant, according to Swisscom's guidelines. When in chat mode, after each user question, the chat history is summarised using another LLM summarization step. This summary is passed into each invocation along with the context and question allowing the chatbot to use the context of the ongoing conversation on future replies.

During development we iterated changes quickly. In order to have a principled and robust way of evaluating the effect of changes on the end-performance, we created an automatic evaluation suite. An LLM was tasked with acting as a judge to compare responses from Sam chatbot, to the responses from our bot. A simple scoring system was designed that awarded +1 if our bot outperformed Sam, 0 if there was a tie, and -1 otherwise. An average score of all trials was used to systematically determine the effect of each individual change in our architecture

3. RESULTS

Our chatbot is able to provide answers to single questions as well as hold a conversation taking the chat history into account. Our chatbot responds coherently and adapts to the user's language, supporting English, French, German and Italian.

If the user prompts a question not related to Swisscom, the chatbot nudges the user to an offered service to further engage with the content, without driving the user to seek an escalation.

On our evaluation suite, the best cumulative average score was 0.78 based on 85 total questions in 4 different languages. This indicates that on average, an unbiased LLM judge favours our bot response over Sam's **8 times out of 10**.

The chatbot gives up-to-date answers based on the dataset given for training and provides a reference URL when appropriate and to an existing webpage.

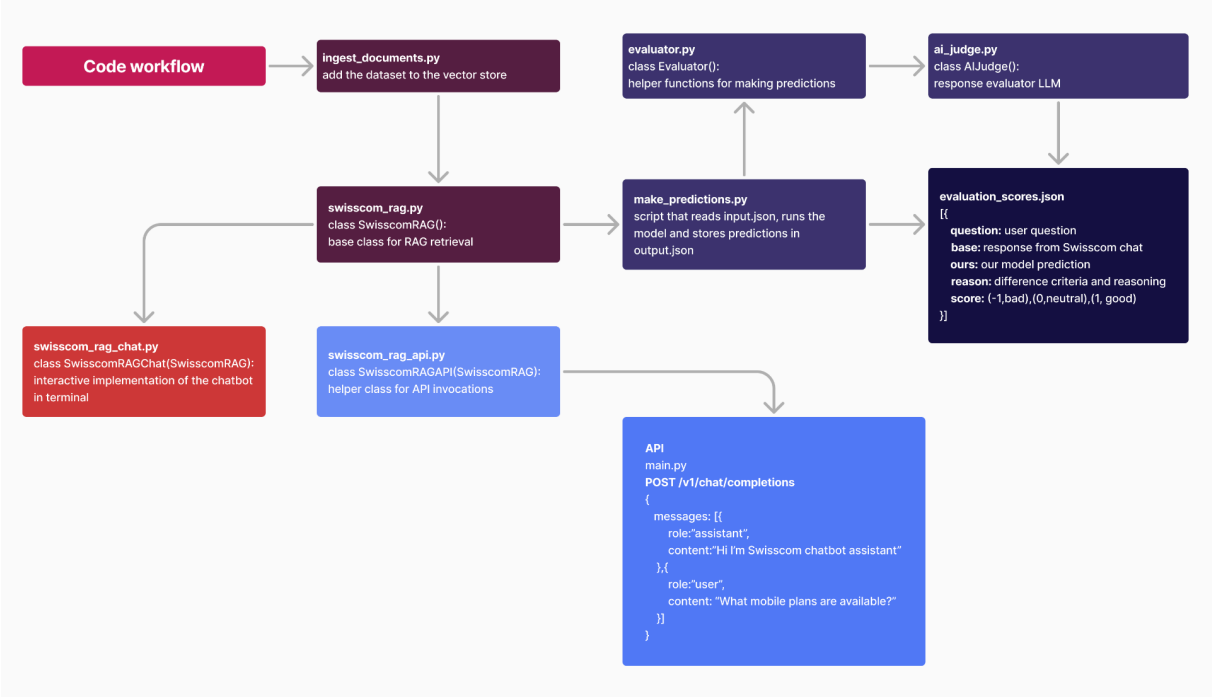

Figure 2: Workflow of the code

4. CHALLENGES

We wanted to implement a local LLM to ensure privacy compliance by creating a chatbot that can be run in a private server environment using open-source solutions. Nevertheless, this proved too costly and resource intensive to serve any purpose for this challenge in this time frame. Also since the data is publicly available, privacy was not really a concern at the moment.

At the start, the responses from the chatbot had several flaws. For example, the chatbot's answer included irrelevant information, did not act like an assistant, and did not include the citation for the information. We solved this by creating a good system prompt.

Summarising the data played a crucial role in the end solution because it allows the chatbot to reduce costs and remove information that is not relevant.

We needed to be very careful when using persistent directories for the vector store. Changing the path would create, without us knowing, a new cache that had not been populated and we would end up not consuming the provided parsed data.

We also encountered some difficulties with the original implementation where the chatbot had an extra language parameter. For a given prompt, the language detector would fail and produce inaccurate results. Switching to the chunk rerank implementation solved this issue, which improves our solution when compared to the actual Swisscom chatbot.