Combining Information Retrieval and Large Language Models for a Chatbot that Generates Reliable, Natural-style Answers

Andreas Lommatzsch, Brandon Llanque, Vinay Srinath Rosenberg, Syed Ali Murad Tahir, Hristo Dimitrov Boyadzhiev and Maurice Walny

Technische Universität Berlin, Straße des 17. Juni 135, D-10623 Berlin, Germany

Abstract

Chatbots efficiently support users in finding relevant answers in a dialog. Traditional chatbot are mostly based on a set of rules, mapping user questions to predefined answers using a knowledge base. Users often miss an adequate adaptation of the answers to the individual needs and the concrete question. Rule-based chats are often perceived as artificial and "wooden". Large language models trained on huge collections of texts and dialog datasets promise to provide natural answers for a broad spectrum of questions. These models perform well in small talk and questions about popular general knowledge. The systems often fail in cases of domain-specific questions providing incorrect, but plausibly sounding answers. In this work we develop and evaluate a chatbot prototype for the administration of a major German city. We combine a traditional knowledge base with different Large Language Models. We evaluate the system with respect to the fluency and the correctness of the answers as well as on the resource demands and response time of the models.

The evaluation of the system shows that existing Large Language Models are able to generate well-understandable answers matching the user's questions. Besides technical issues such as high resource consumption and limited scalability, adequate prompts are crucial to force the model to use reliable data from a trusted knowledge base for generating the answers to avoid both hallucinations and formulations that are not well suited for the concrete context.

1. Introduction

The rapid advances in automatic natural language processing and text generation lead to high expectations of chatbots. Traditional chatbot approaches are mainly rule-based and require manual effort to adapt the required knowledge for a chat. The systems do neither support an adaptation to the concrete user question nor follow-up questions.

Large Language Models (LLMs) are trained to answer user questions in a natural dialogue. The models are able to provide world knowledge for small talk and for answering simple questions. Large Language Models are trained on huge document and web page collections making the training process very resource-consuming and costly. Weaknesses of the approach

andreas.lommatzsch@tu-berlin.de (A. Lommatzsch); b.llanque_kurps@tu-berlin.de (B. Llanque); vinay.rosenberg@campus.tu-berlin.de (V. S. Rosenberg); s.tahir@campus.tu-berlin.de (S. A. M. Tahir); boyadzhiev@campus.tu-berlin.de (H. D. Boyadzhiev); m.walny@tu-berlin.de (M. Walny)

 In: M. Leyer, Wichmann, J. (Eds.): Proceedings of the LWDA 2023 Workshops: BIA, DB, IR, KDML and WM. Marburg, Germany, 09.-11. October 2023, published at CEUR Workshop Proceedings (CEUR-WS.org)

are training on historic (often outdated) documents, biases in the training dataset and the problem of hallucinations (well-sounding, but incorrect answers).

In this work, we analyze the scenario of developing a chatbot providing user questions related to the administrative services of a major German city. The challenge is to ensure correct, reliable answers while providing them in a natural style as well as adapting the answers to the user questions.

We develop a prototype combining an Information Retrieval-based approach with Large Language Models for generating reliable, natural-style answers. We study several locally deployed LLMs as well as a cloud-based language model. For ensuring a real-world evaluation, we implement a web-based chat application allowing users to chat with our bot. The users can rate the answers and give feedback for the dialogues. In the evaluation, we focus of the following research questions:

- How good are the generated answers in terms of fluency and correctness
- What is the user's perception of the answer quality?
- How do different large language models perform? What is the technical complexity for using large language models in a chatbot system?
- How do LLM-based chatbots perform compared with traditional chatbot system?

The remainder of this paper is organized as follows. In Section 2 related research is discussed to identify promising methods and strategies. Section 3 describes our approach and the used data. Subsequently, the evaluation of our approach is presented in Section 4, through quantitative and qualitative analysis of the chosen methods and developed chatbots. Finally, Section 5 discusses the accomplishments of this paper and provides an outlook on future work.

2. Related Work

In this section we review existing chatbot system and explain recent Large Language Models.

2.1. Chatbot Systems

Chatbot systems, also known as conversational agents, are heterogeneous systems that converse with humans through text or voice. Their field of application range from education, administration, entertainment, and health care to numerous other domains.

One of the first chatbot systems designed in the 60s, **ELIZA** [1] used pattern matching to answer user queries. The system was designed to mimic a psychiatric interview, to give the user the feeling of being heard and understood. The system generates answers by picking words from the user input and filling template answers using a fixed set of rules. Due to the limited number of used rules and patterns, *ELIZA*'s responses repeat themselves after a short period of use.

ALICE, also a pattern matching-based chatbot system, uses AIML (Artificial Intelligence Mark-up Language) files to define new patterns and rules. The AIML files made it simple for adding multiple pattern matching rules to the chatbot. ALICE won the Loebner-Prize [2] three

different times [3]. The used pattern-matching mechanism makes it very costly to develop chatbots for a large knowledge domain since each user-answer pattern must be defined explicitly. The variation in user language, spelling mistakes as well as very specific questions can hardly be handled by this approach.

More recent, well-known conversational agents are **Siri**[1] by Apple, Amazon's Alexa[2], and Google Assistant[3]. These systems mainly combine predefined rules with queries mapping user inputs to document lists or database entries. The system do not support follow-up questions nor the generation of answers with respect to the concrete user question.

In 2022 OpenAI released their publicly available chatbot **ChatGPT**[4]. ChatGPT (based on the model *GPT-3.5*) is capable of producing human-like texts and can manage user queries from many domains. The system provides natural language answers, but answers for specific domains are often superficial or generated based on outdated knowledge. In addition, there is the risk of hallucination (wrong, but correct sounding answers). Due to well-formulated answers, ChatGPT is a very promising approach. The availability of an API opens the possibility of optimizing prompts and for combining the *GPT-3.5* model with other methods.

In contrast to chatbots designed to answer general questions, domain-optimized chatbots exist. For questions related to the public administration, several German cities provide chatbots. E.g., the German city Heidenheim released the chatbot Kora [4] in 2018. It works as an electronic city hall assistant and covers topics ranging from weather to events [4]. The chatbot uses a database containing the most frequently requested answers and maps user inputs to the best matching answers.

The city of Berlin runs the chatbot Bobbi [5, 6]. Chatbot Bobbi imports the service description from the official Berlin website and converts it in a format optimized for a chat. Instead of providing long texts, Bobbi provides the paragraphs matching best to the user question and guides the user through the service description by recommending additional relevant information. The system provides reliable answers, but the answers are often perceived as difficult to understand. For ensuring reliability of answers, the chatbot uses the sentences extracted from the official web site - thus, the answers are only adapted to the concrete user question and context.

The analysis shows that the existing chatbots are useful, but answers are often not well adapted to the user question and the context. The use of answer generation (as demonstrated by ChatGPT) instead of providing predefined answers seems to be a promising approach to overcome the problem. To solve the issue of hallucinations and outdated data, the answers should be generated based on a reliable, trusted knowledge base.

2.2. Large Language Models

With the publication of ChatGPT[5], Large Language Models (LLMs) have received a lot of attention due to their ability to generate emphatic, human-like responses. The ability of LLMs

[1] https://www.apple.com/de/siri/
[2] https://developer.amazon.com/de-DE/alexa
[3] https://assistant.google.com/intl/de_de/
[4] https://openai.com/
[5] https://openai.com/blog/chatgpt

to engage in conversations like a human is most often based on the Transformer architecture, specifically the decoder section. The Transformer model, first introduced in the paper "Attention Is All You Need" [7], improved the natural language processing by effectively capturing relationships between words, applying a multi-head attention mechanism [8].

Large Language Models have also been adapted for conversational AI scenarios. The models leverage the transformer architecture, which effectively captures intricate relationships between words, enabling them to achieve remarkable results in understanding user queries and generating contextually relevant, coherent, and human-like responses. However, training LLMs requires substantial amounts of data, leading to high computational costs. To address this challenge, several pre-trained LLMs have been released, allowing for fine-tuning on specific datasets while reducing computational resources [9]. Among the noteworthy pre-trained models are Bidirectional Encoder Representations from Transformers (BERT) [10, 11], and Llama 2.0 [12], an open-source large language model recently released by Meta. These models enable developers to build powerful chatbots that can interact naturally with users.

In the last months several different LLMs have been released that differ in the number of parameters, the architecture, the training data, and the used license model. Locally deployed, open source models are promising to avoid legal problems; in addition, these models may support a scenario-specific fine-tuning. This is the motivation for us to analyze recently published Large Language Models.

Dolly 2.0 is an open-source large language model developed by Databricks[6], which can be used for commercial purposes [13]. The model has 12 billion parameters and shows good performance in the Huggingface Open LLM Leaderboard[7]. The weaknesses of the model are that the model is not optimized for German and shows a reduced answers quality for complex questions.

LLaMA (Large Language Model Meta AI) is a LLM designed by *Meta* for research purposes. The model is trained on a large unlabeled dataset, making it ideal for fine-tuning on a variety of tasks. It comes in sizes of 7B, 13B, 33B, and 65B parameters. The smallest model, *LLaMA 7B*, is trained on one trillion tokens [14]. **Alpaca 7B** is a fine-tuned version of Meta's *LLaMA 7B* model that was trained on 52K instruction-following demonstrations generated in the style of self-instruct [15].

Vicuna 13B is an open-source chatbot developed through fine-tuning the *LLaMA* model. The training process involved using a dataset obtained from ShareGPT, consisting of approximately 70,000 user-shared conversations gathered through ShareGPT.com's public APIs. During the evaluation, OpenAI's *GPT-4* model served as the judge, demonstrating that *Vicuna 13B* achieves over 90% quality compared to OpenAI *GPT-3.5*. In addition, it has been reported, that in more than 90% of cases, Vicuna 13B outperforms other models like *LLaMA* and Stanford *Alpaca* [16].

Zicklein is a German variant of *Alpaca 7b* that has been fine-tuned using LoRA (Language Overlap Relevance Adaptor) [17]. For training *Zicklein*, a translated version of the cleaned Stanford Alpaca dataset was used, ensuring its alignment with the German language. By fine-tuning *Alpaca 7b* with a specific focus on German language processing, *Zicklein* has been tailored to better understand and generate responses in German. This adaptation enables the *Zicklein*

[6]https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm
[7]https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

model to effectively comprehend and produce accurate results in German conversations[8].

The LLaMA-based models provide acceptable results for English and German queries. The larger models show a remarkable better answer quality, but larger model are much slower and more resource-demanding. This is a strong limitation for a chatbot system from which users expect quick responses.

RWKV is an LLM that, other than the previous examples listed here, is based on the RNN instead of the transformer architecture [18]. It's attention equivalent is of linear instead of quadratic complexity as in transformers, allowing for far larger context window sizes [19]. The model used here was trained for sizes of 8,192 tokens (as compared to the 2,048 tokens common in other open source models). The *RWKV* model is interesting due to the large number of supported input tokens, making it possible to consider longer conversations (by adding the dialog history in the prompt) and more complex data (retrieved from a knowledge base).

GPT-3.5 (Chat Generative Pre-Trained Transformer) is a conversational AI model developed by *OpenAI*. The model comes in various size of 125M, 350M, 760M, and 1.3B parameters. The model has been fine-tuned for conversational applications using a combination of supervised and reinforcement learning techniques [20].

There are very fast developments in the domain of Large Language Models. Researchers train models on larger data collection to improve the result quality. Another trend are models optimized to work with a smaller number of parameters reducing the need for expensive hardware. Based on the literature analysis alone it is hard to predict how the existing models perform in a given scenario. This is the motivation for us to implement a system that provides an API for integrating different models. This allows us to analyze the strengths in weaknesses of the models in a real-world setting.

2.3. Discussion

Analyzing the strengths and weaknesses of existing chatbots, using template-based answers, and LLMs shows, that both techniques are complementary: LLMs enable natural-style answers, but tend to provide incorrect answers; template-based chatbots provide reliably correct answers, but poorly adapt to the context and the user questions. A promising approach may consist of combining the techniques: The reliable correct answers provided by IR-based methods could be used as the input for LLMs that generate natural-sounding, context-adapted answers.

3. Approach

We develop a chatbot prototype combining knowledge retrieval and Large Language Models for generating natural-style answers. In order to evaluate different models and approaches in a real-world setting, we implement a web application allowing "normal" users to chat with the bot. Users can rate the answers and give feedback on the bot-generated answers. The chatbot back-end uses APIs to different LLMs allowing us to evaluate the answer quality and the technical complexity of the LLMs.

[8] https://github.com/avocardio/Zicklein

3.1. Scenario and used Knowledge Base

As interesting example application use case, we decided to develop a chatbot optimized to answer questions related to the citizens services offered by Berlin's administration (similar to the already existing chatbot *Bobbi*). The chatbot should reliably and correctly answer questions related to administrative services (e.g., "How to apply for a new ID card", "What do I need for getting a parking permit" or "How much is a new passport"). Moreover, the bot should also be able to handle any question that a human chat partner also would be able to answer.

As reliable knowledge source for our chatbot, we use the descriptions of the services offered by the Berlin's administration[9] providing detailed data for about 1,000 services. We store the data in an *Apache SolR*[10] server enabling an efficient fuzzy full-text search. The service descriptions (optimized for the official Berlin website) are often complex and in a formal language. Thus, the knowledge base provides reliable data, but the descriptions are not designed for an interactive chat.

3.2. System Architecture

The developed system consists of six main components (Fig. 1): The web-based user interface, the chatbot backend, the *SolR*-Server, the prompt generation module, and the Large Language Models. A user query, send from the chat window in a browser is received by the backend. The backend queries relevant information from the SolR-Server. If the user question is general small talk or "off-topic", the SolR-Server returns an empty result. Then, the backend selects a LLM for the user questions. Dependent on the selected model, a prompt is generated. The prompt consist of an instruction set, the user question and the knowledge retrieved from *SolR* (if relevant knowledge could be retrieved). The output of the LLM is send back via the backend to the user interface. The user can rate the answer quality. Collected ratings are stored in a SQL-based feedback database.

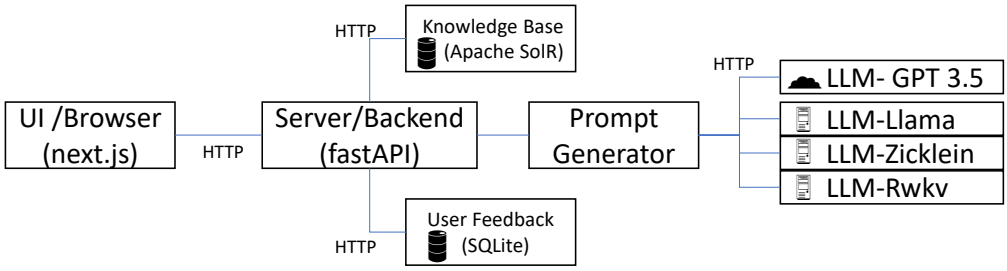

Figure 1: *The System Architecture: The User Interface interacts with a backend that receives the user question. The backend queries the data relevant for the question from the knowledge base. Then a LLM is selected. Based on the user input, the knowledge retrieved from SolR and the chat history, a prompt for the selected LLM is generated. The answer generated by the LLM is sent back to the user interface.*

[9] https://service.berlin.de
[10] https://solr.apache.org/

3.3. Implementation and Deployment

We implement the system using widely adopted, open source components. For the implementation of the web-based user interface we use the *next.js* framework. The backend is implemented in *Python 3* using *FastAPI*. The backend interacts with the *Apache SolR* knowledge base using *HTTP*. The large language models are integrated based on REST-API enabling the seamless interaction both with cloud-based servers and locally deployed models.

In the deployed system we integrated *OpenAI GPT-3.5* (cloud-based) and several locally deployed models, such as *Llama.cpp*, *zicklein*, and *rwkv*. For collecting user feedback and user ratings for answers, we use an SQLite database[11].

The implementation enables the flexible integration of additional knowledge sources as well as the integration of new LLMs. The most challenging task in the implementation was the set-up of the locally deployed LLMs. Most models are not very mature and poorly documented. Finding adequate hyper parameters for receiving acceptable results had been more time consuming than expected.

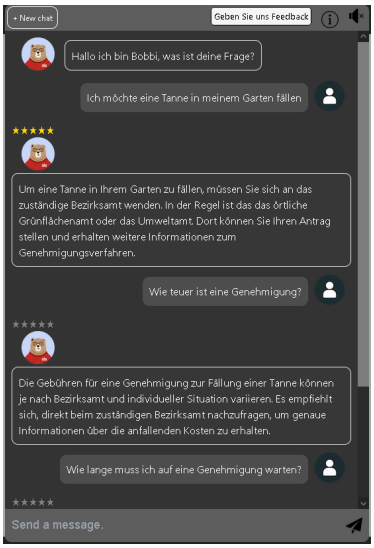

Figure 2: A screenshot of the developed chatbot

We deployed the system on a standard VMware-based virtual machine having 8 cores, 32GB RAM, and 800 GB disk space, a system similar that is used by the official Bobbi chatbot. Note, that the VM does not have a GPU; thus, the chatbot system uses pre-trained models.

Overall, the developed application works well in the used deployment. Fig. 2 shows a screenshot of the application. The bot provides compact, natural style answers, adapted to the user question. Users can rate each answer; in addition, users may provide text-based feedback using the button in the page header.

4. Evaluation

In our evaluation we analyze the overall system performance and investigate the strengths and weaknesses of the used components in detail. We focus on the selection of LLMs fitting best the requirements of our scenario and the overall answer quality.

4.1. Evaluation of Local LLMs

As discussed in the related work section several locally deployable LLMs exits. These models avoid the legal problem of cloud services. In addition, most of the models can be fine-tuned to the requirements of a specific scenario or vocabulary. However, they are still in development - this raises the question whether these models perform well enough to be integrated in our chatbot system.

Two very promising LLMs for a local deployment are *Dolly 2.0* and *Vicuna13b* due to the reported good results on English benchmark datasets and their open source license. We decided

[11] https://www.sqlite.org/index.html

to evaluate them more closely and selected two medium-size datasets: the *History of Hamburg* dataset [21] (in German) and the *London city* dataset [22], which is composed of historical facts about the city of London in English. The tests have been conducted on a desktop PC equipped with a GPU, unlike our shared test server.

Dolly 2.0 For the evaluation, we downloaded the Dolly 2.0 model to our training machine and used the *LangChain* library to train the model on the history of the Hamburg dataset. Comparing the trained Dolly2.0 model-based results, we observed satisfactory results for questions posed in English; but the model frequently returned hallucinations when handling questions in German. Furthermore, the model fell short in generating engaging in chat-like interactions, resulting in answers inadequately considering the context and the language style. Our attempts to fine-tune the pre-trained model using our German dataset could not significantly improve the answer quality for German questions. We decided that the model cannot be used for our application due to the limitation described above, even though *Dolly 2.0* initially appeared very promising.

Vicuna13b Similar to the *Dolly 2.0* model, we also evaluate the open source model *Vicuna13b*. For the English dataset, the model performed exceptionally well. It outperformed *Dolly 2.0* significantly, demonstrating the ability to answer questions about London with good accuracy and generating responses that appeared remarkably human-like. The model showed the ability in engaging, extended and contextually coherent conversations. However, the model provided unstable responses for German questions with a majority of answers ranging from somewhat correct to entirely incorrect. The model's limited understanding and inaccuracies were evident in its attempts to comprehend and respond appropriately to queries in German. This made the model similarly unsuitable for our needs.

Discussion The evaluation of both the Dolly2.0 and Vicuna13b models showed the inadequate performance of handling German requests. The relatively big fraction of hallucinations and poor responses, raised concerns about their reliability and suitability for our intended application. Given that the models struggled to handle smaller datasets effectively, integrating them with our *SolR* database would likely exacerbate these issues. As a result, we decided not to use these models for our chatbot prototype. We only considered *Llama*, *zicklein*, and *rwkv* as local LLMs.

4.2. User Study Evaluation

For this evaluation, we use the developed web application, integrating our *SolR* database, several local LLMs, and the cloud-based *GPT-3.5*. The initial Local LLM models we chose to integrate were Llama variants, such as *zicklein*, and *rwkv*. Each of these models is known for its distinct characteristics and by evaluating them side by side with *GPT-3.5* we aim to identify which model best suits our specific use case.

From the available OpenAI LLMs, we decided to utilize the *gpt-3.5-turbo-16k* version. This model has demonstrated superior performance compared to its predecessors. There are several key reasons why we have opted for this particular model over other alternatives: Firstly, the *gpt-3.5-turbo-16k* model provides highly optimized conversations at a significantly reduced cost, amounting to just 1/10th of the price associated with *text-davinci-003* model. Furthermore, it offers an impressive context length of 16,384 tokens, which is four times greater than the 4k *GPT-3* base model's capacity.

4.3. Locally deployed Models vs. the Cloud-based GPT-3.5 Model

During the users' evaluation, participants had the chance to interact with the local LLMs. The results revealed significant limitations. One of the most prominent issues reported by users was the slow response time, with some queries taking 2-5 minutes to receive a reply (most likely due to lack of GPU support). In some cases, users had to iterate multiple times without ever getting a response, leading to frustration. Furthermore, the accuracy of the LLM's responses was severely lacking. E.g., when asked about how to apply for a "Führungszeugnis" (police certificate), the local models often provided answers related to "Führerschein" (Driver's License), showing a lack of understanding of the context. The models also struggled to summarize text effectively, often failing to grasp the essence of the questions. Many queries went unanswered, and even when responses were generated. Moreover, the local LLMs were not capable of handling long user input queries due to unacceptable response times and due to token limitations. While the *rwkv* model showed relatively better results compared to others, it still fell short of providing satisfactory answers to user questions regarding the Berlin administration.

A noteworthy feature of *GPT-3.5* was its capability to understand follow-up questions. For example, when asked about the nearest "Bürgeramt", the model inquired about the user's exact location and then provided a list of Bürgeramt locations nearby. Upon asking for other Bürgeramt options in the same region, the model accurately provided three additional correct responses. Follow-up questions require the support of long user inputs, since older interactions of the conversation must be provided to the LLM. Due to limitations of the number of supported input tokens, the answers for follow-up questions was limited from the local LLMs; additional LLM-specific prompt tuning may improve this issue.

Compared to local LLMs, answers from *GPT-3.5* were remarkable better in terms of performance, response time, and accuracy. While other local LLMs often took an average of 3 minutes to generate a response, *GPT-3.5* was able to provide quick replies within a matter of seconds. This improvement in response time enhanced the user experience, allowing for more fluid and interactive conversations. The quality of responses provided by *GPT-3.5* was far more natural compared to our current live chatbot Bobby. The model's ability to adapt to the context and user intent enabled it to generate responses that felt more human-like and contextually relevant. Users expressed their appreciation for the coherence and readability of the replies *GPT-3.5* model in the evaluation. Another notable aspect of the model was its high accuracy in providing summaries. Regardless of the type of question asked, the model consistently delivered precise and concise summaries. This accuracy contributed to more efficient information retrieval and saved users valuable time in searching through lengthy responses. The strengths of summaries lies in the fact, that the risk of hallucinations is minimized making the generated answers more reliable.

Analyzing the multi-language support of the different models, we observer that *GPT-3.5* exhibited an very good ability to understand and provide responses in languages other than German, despite being connected to our database primarily for German data. As a result, customers who were not native speakers of German or English could still use the chatbot efficiently and receive clear information in their language of choice.

4.4. User Study Results

To gain a deeper understanding of our models, we conducted a user analysis by providing a set of questionnaires to the participants after they tested our models. The participants have been recruited from university students.

The overall model performance of the answers was rated between 4.1 and 4.4 on a scale of 1 to 5. The majority of users expressed that the model performed significantly better compared to the official chatbot *Bobby*. Users appreciated the LLM's capability to understand and generate informative responses. However, the *GPT* model showed limitations when asked questions in extreme detail, such as providing a link to a specific website, form, or the exact address of a "Bürgeramt". In such cases, the model occasionally produced inaccurate results.

Analyzing the quality of follow-up questions (essential for a natural dialog) we found, that the uses were very satisfied with the answers (making big difference to most existing chatbots). This is an improvement compared to the official chatbot Bobby that only a limited set of aspects as follow-up question. LLMs are able to provide context-adaptive formulations - this is perceived as helpful, if an initial answer does not provide sufficient information.

However, we observed that the model's performance decreases if the number of follow-up questions increases or if the content of the conversation becomes lengthy. In such cases, the *GPT-3.5* model exhibited signs of hallucination. Moreover, when combining two or more questions together in a single query, the results were not as accurate. The model struggled to provide precise responses when faced with more complex, multipart questions. During the user study, a significant number of users faced difficulties accessing the local LLM models due to long wait times, and in some cases, the models failed to respond even after multiple app restarts. However, there were some users who managed to use the local LLMs, and the results they obtained were less promising when compared to the performance of the OpenAI model.

Overall, *GPT-3.5* emerged as a far better model compared to the local LLM models. It showcased quicker response times, higher accuracy, and more proficient handling of follow-up inquiries, making it the superior choice.

5. Conclusion

In this work, we presented a chatbot system that combines Information Retrieval Methods and Large Language Models for generating reliable, natural style answers for questions related to services offered by the public administration. The developed prototype enabled us to evaluate the models in a real-world setting and to collect user feedback.

Overall the approach worked. The system provided easily readable answers in a natural, less administrative style compared to the official source. *GPT-3.5* summarized complex service descriptions excellently. By adding a question-dependent introduction, the answers fitted with the questions and were perceived as a useful, well matching answer.

Analyzing the weaknesses, the most critical problem are hallucinations. Even though we prompted the LLMs to generate answers based on the knowledge from the official knowledge base, the answers are not always correct. This may be because not all the required knowledge could be retrieved from the knowledge base (e.g. if a user intent in not well covered in the service database). In such cases the chatbot may generate a plausible answer that sounds correct to the user, but is not correct from an expert's perspective. We expect that optimizing both the IR

process and the prompt generation for the LLMs could solve this issue. Since the summarization and the adaptation to a specific style works well using *GPT*, the prompts should be optimized in the way that answers are very close to the reliable information (retrieved from the knowledge base). Alternatively, the LLM should check how close a generated answer is to the data in the knowledge base to give the users an explanation which parts of the answer are less reliable.

When analyzing the answers generated by the OpenAI *GPT-3.5* service the answers have the typical call-center style: First saying that the bot understands the problem and tries to help, followed by the requested information. Most answers close with the sentence that the information should be double-checked with the website. This pattern works well, if a third party portal (e.g. a web forum) provides an answer. If the authority responsible for the concrete problem (e.g. the official administration) points out that the answers may not correct, it reduces the trust in chatbots and the authority itself. On the other hand, a summarization of complex texts always results in a simplification that may leave out rare exceptions.

Another important aspect is that the evaluation of chatbot system designed for answering user questions is difficult, since users do not know the correct answer for a question. An answer is typically perceived as "good", if the content of the answer sounds plausible and the wording is "human". Since texts provided by administrative officials are often complex, formal, and not well adapted to an individual question, "normal" users mostly prefer LLMs-generated answers over the approved answers provided by the administration; on the other hand, administrative experts prefer the 100% correct answers in formal language.

Comparing the cloud-based *GPT-3.5* model with the locally deployed models, the *GPT* model outperformed the local model significantly in terms of result quality and time of answering a request. Due to data privacy concern, cloud-based approaches are not always an option; we expect that by updating our hardware infrastructure and a fine tuning of the local models, these problems can be mitigated. But LLMs are computational expensive due to the huge number of parameters, limiting the scalability of the models. On the other hand, the domain-specific optimization of models and prompts is an interesting research task.

As future work, we plan to conduct a larger user study to analyze in detail the properties of the generated answers depending on the context and concrete question. In addition, we plan to improve the IR method to ensure that all necessary knowledge is included in the prompt to ensure reliable answers. We are also working on improving the local LLMs and optimizing the prompt generation.

Acknowledgment

We thank the ITDZ Berlin for supporting the development of the chatbot framework.

References

[1] J. Weizenbaum, Eliza—a computer program for the study of natural language communication between man and machine, Communications of the ACM 26 (1983) 23–28.

[2] M. M. al Rifaie, AISB - The Society for the Study of Artificial Intelligence and Simulation of Behaviour - Loebner Prize, https://web.archive.org/web/20190715032609/http://www.aisb.org.uk/events/loebner-prize, 2019.

[3] B. A. Shawar, E. Atwell, A comparison between Alice and Elizabeth chatbot systems, University of Leeds, School of Computing research report 2002.19, Leeds, 2002.
[4] Heidenheim, neue Webseiten Jan 2021, 2023. URL: https://www.heidenheim.de/neue+webseiten+jan+2021.
[5] Chatbot Bobbi, 2023. URL: https://service.berlin.de/chatbot/chatbot-bobbi-606279.php.
[6] A. Lommatzsch, J. Katins, An information retrieval-based approach for building intuitive chatbots for large knowledge bases, in: Proceedings of the LWDA conference 2019, CEUR Workshop Proceedings, 2019, pp. 343–352.
[7] V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, N. Aidan, K. Lukasz, P. Illia, Attention is all you need, https://arxiv.org/abs/1706.03762, 2017.
[8] C. Qiwei, Z. Huan, L. Wei, H. Pipei, O. Wenwu, Behavior sequence transformer for e-commerce recommendation in alibaba, https://arxiv.org/abs/1905.06874, 2019.
[9] K. Shah, Pre-training, fine-tuning and in-context learning in Large Language Models (LLMs), https://medium.com/@atmabodha, 2022.
[10] D. Jacob, C. Ming-Wei, L. Kenton, T. Kristina, Bert: Pre-training of deep bidirectional transformers for language understanding, https://arxiv.org/abs/1810.04805, 2018.
[11] V. Jannis, G. Johannes, S. Rico, Swissbert: The multilingual language model for switzerland, https://arxiv.org/abs/2303.13310, 2023.
[12] Meta, Meta and Microsoft Introduce the Next Generation of Llama, https://about.fb.com/news/2023/07/llama-2/, 2023.
[13] C. Mike, H. Matt, M. Ankit, X. Jianwei, W. Jun, S. Sam, G. Ali, W. Patrick, Z. Matei, X. Reynold, Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM, https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm, 2023.
[14] Meta, Introducing LLaMA: A foundational, 65-billion-parameter Large Language Model, https://ai.meta.com/blog/large-language-model-llama-meta-ai/, 2023.
[15] T. Rohan, G. Ishaan, Z. Tianyi, D. Yann, L. Xuechen, G. Carlos, L. Percy, B. Tatsunori, Alpaca: A Strong, Replicable Instruction-Following Model, https://crfm.stanford.edu/2023/03/13/alpaca.html, 2023.
[16] The Vicuna Team, Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality, https://lmsys.org/blog/2023-03-30-vicuna/, 2023.
[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, 2021. arXiv:2106.09685.
[18] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV, X. He, H. Hou, P. Kazienko, J. Kocon, J. Kong, B. Koptyra, H. Lau, K. S. I. Mantri, F. Mom, A. Saito, X. Tang, B. Wang, J. S. Wind, S. Wozniak, R. Zhang, Z. Zhang, Q. Zhao, P. Zhou, J. Zhu, R.-J. Zhu, RWKV: Reinventing RNNs for the Transformer Era, https://arxiv.org/pdf/2305.13048, 2023. arXiv:2305.13048.
[19] F. D. Keles, P. M. Wijewardena, C. Hegde, On The Computational Complexity of Self-Attention, 2022. arXiv:2209.04881.
[20] J. D. Chang, K. Brantley, R. Ramamurthy, D. Misra, W. Sun, Learning to Generate Better Than Your LLM, 2023. arXiv:2306.11816.
[21] Wikipedia, Geschichte Hamburg, https://de.wikipedia.org/wiki/Geschichte_Hamburgs, 2023.

[22] B. Ehrlich, H. Clout, M. Hebbert, London, https://www.britannica.com/place/London, 2023.