KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Contextual short-term memory for LLM-based chatbot

MIKAEL TÖRNWALL

Stockholm, Sweden, 2023

Contextual short-term memory for LLM-based chatbot

MIKAEL TÖRNWALL

Master's Programme, Computer Science, 120 credits
Date: July 12, 2023

Supervisor: Alireza Mahmoudi Kamelabad
Examiner: Olov Engwall
School of Electrical Engineering and Computer Science
Swedish title: Kontextuellt korttidsminne för en LLM-baserad chatbot

Abstract

The evolution of Language Models (LMs) has enabled building chatbot systems that are capable of human-like dialogues without the need for fine-tuning the chatbot for a specific task. LMs are stateless, which means that a LM-based chatbot does not have a recollection of the past conversation unless it is explicitly included in the input prompt. LMs have limitations in the length of the input prompt, and longer input prompts require more computational and monetary resources, so for longer conversations, it is often infeasible to include the whole conversation history in the input prompt. In this project a short-term memory module is designed and implemented to provide the chatbot context of the past conversation. We are introducing two methods, LimContext method and FullContext method, for producing an abstractive summary of the conversation history, which encompasses much of the relevant conversation history in a compact form that can then be supplied with the input prompt in a resource-effective way. To test these short-term memory implementations in practice, a user study is conducted where these two methods are introduced to 9 participants. Data is collected during the user study and each participant answers a survey after the conversation. These results are analyzed to assess the user experience of the two methods and the user experience between the two methods, and to assess the effectiveness of the prompt design for both answer generation and abstractive summarization tasks. According to the statistical analysis, the FullContext method method produced a better user experience, and this finding was in line with the user feedback.

Keywords

Chatbot, Artificial Intelligence, Machine Learning, Language Model, Large Language Model, GPT-3, Natural Language Processing, Text Summarization, Dialogue Summarization, Prompt Design, Prompt Programming

Sammanfattning

Utvecklingen av LMs har gjort det möjligt att bygga chatbotsystem kapabla till mänskliga dialoger utan behov av att finjustera chatboten för ett specifikt uppdrag. LMs är stateless, vilket betyder att en chatbot baserad på en LM inte sparar tidigare delar av konversationen om de inte uttryckligen ingår i prompten. LMs begränsar längden av prompten, och längre prompter kräver mer beräknings- och monetära resurser. Således är det ofta omöjligt att inkludera hela konversationshistoriken i prompten. I detta projekt utarbetas och implementeras en korttidsminnesmodul, vars syfte är att tillhandahålla chatboten kontexten av den tidigare konversationen. Vi introducerar två metoder, LimContext metod och FullContext metod, för att ta fram en abstrakt sammanfattning av konversationshistoriken. Sammanfattningen omfattar mycket av det relevanta samtalet i en kompakt form, och kan sedan resurseffektivt förses med den påföljande prompten. För att testa dessa korttidsminnesimplementationer i praktiken genomförs en användarstudie där de två metoderna introduceras för 9-deltagare. Data samlas in under användarstudier. Varje deltagare svarar på en enkät efter samtalet. Resultaten analyseras för att bedöma användarupplevelsen av de två metoderna och användarupplevelsen mellan de två metoderna, och för att bedöma effektiviteten av den snabba designen för både svarsgenerering och abstrakta summeringsuppgifter. Enligt den statistiska analysen gav metoden FullContext metod en bättre användarupplevelse. Detta fynd var även i linje med användarnas feedback.

Nyckelord

Chatbot, Artificiell Intelligens, Maskininlärning, Språkmodell, Stor Språkmodell, GPT-3, Naturlig Ppråkbehandling, Textsammanfattning, Sammanfattning av Dialog, Design för Inmatningsprompt, Inmatningsprompt Programmering

Acknowledgements

I want to thank my thesis supervisor, doctoral student Alireza Mahmoudi Kamelabad for his efforts and guidance throughout the project, and my thesis examiner, professor Olov Engwall for his feedback and help in finalizing my thesis. I am also grateful to my opponent, Samuel Söderberg, for his feedback and suggestions.

In addition, I want to thank each study participant for their time and commitment to participating in this study.

Lastly, special thanks to Filippa Kärrfelt and Isak Pettersson for their moral support, and Mikael Lundqvist for his help with the Swedish translations.

Helsinki, July 2023
Mikael Törnwall

Acronyms

AI Artificial Intelligence

ANN Artificial Neural Network

API Application Programming Interface

ASR Automatic Speech Recognition

BERT Bidirectional Encoder Representations from Transformers

CUQ Chatbot Usability Questionnaire

DL Deep Learning

DM Dialog Manager

GPT Generative Pre-trained Transformer

GPT-2 Generative Pre-trained Transformer 2

GPT-3 Generative Pre-trained Transformer 3

LM Language Model

LLM Large Language Model

ML Machine Learning

NL Natural Language

NLP Natural Language Processing

NLU Natural Language Understanding

NN Neural Network

RoBERTa Robustly Optimized BERT Pretraining Approach

ROUGE Recall-Oriented Understudy for Gisting Evaluation

SOTA State of the art

SUS System Usability Scale

Contents

Chapter 1

Introduction

1.1 Background

LMs are probability distributions trained with text corpora. Given a sequence of words LMs predict the probability of such sequence. Large Language Models (LLMs) are LMs that are trained with large text corpora and contain a large number of parameters, usually in the billions. LLMs, such as Generative Pre-trained Transformer 3 (GPT-3) [2], have transformed both research and application of Natural Language Processing (NLP), an area of computer science that studies how computers process text data.

LLMs can be used to produce a Natural Language (NL) text output given some NL input prompt, and one of the application areas is predicting the next turn in a conversation. This means that LLMs can be used as the prediction engine for chatbot systems without the need for additional task-specific datasets [16]. Another application area of LLMs is abstractive text summarization [9, 31].

LMs are *stateless*. This means that LMs have no memory of previous events unless these are explicitly introduced in the input prompt. In the context of chatbots, this means that a chatbot can only remember what is included in the input prompt, but with long conversations, it is infeasible to have tens and hundreds of conversation turns included in the input prompt due to time, monetary and system limitations. Having a memory of the conversation history has the potential to improve the user experience of chatbot systems, as such systems may be able to produce more personalized and relevant responses.

This study investigates ways to implement a short-term memory for a LM based chatbot to improve the quality of a conversation while keeping the input prompt short. The quality is mainly assessed by the user experience. The chatbot system used in this study utilizes GPT-3 to predict the output and to create an abstractive summary of the conversational history that we can use as a contextual short-term memory. Two methods for the short-term memory implementation, LimContext method and FullContext method, are introduced. Both methods generate summaries using a number of previous conversation turns. The difference is that the FullContext method also uses the previous summary to generate the new summary. This way the information from the beginning of the conversation can persist without the need for long input prompts.

The motivation for this approach is twofold. Firstly, the parameters of the LM do not change as the conversation progresses. This means that the LM has no recollection of the previous conversation apart from what is included in the input prompt. From the user's perspective, it would be good if the chatbot had the ability to remember the conversation history, for example, to avoid repetition or to have a more personalized experience. Secondly, as the conversation progresses the length of the conversational history increases. This is bad for several reasons; more computational resources are required, the system administrator experiences higher monetary costs and LLMs may have a limit for the maximum prompt length, which means that the conversation window that can be included in the input prompt is limited in size. This means that conversational history as it is cannot be included in the input prompt in longer conversations. By creating an abstractive summary of the conversation history and including it in the input prompt, we can sustain the context and details of the conversation while keeping our prompts short.

This study also contributes to the larger goal of designing effective prompts for LLMs, both for dialogue turn generation and summarization tasks and for designing user-friendly chatbots.

1.2 Problem

Creating a chatbot that can sustain long and meaningful conversations is difficult due to the fact that both the GPT-3 language model and user inputs are complex and outputs are hard, if not impossible, to predict [16]. Designing a whole system is out of the scope of this study. The method proposed may be one step towards enhanced prediction performance for GPT-3.

The prompt is the only task-specific information we can provide for the LM, so it is preferable that we can include much of the relevant information in the prompt. As the prompt length is limited and longer prompts require more resources, it is important to investigate ways to include the most fundamental information without the need to append most of the conversational history.

1.3 Research Questions

RQ1 How is the user experience of the two methods for short-term memory summarization for a GPT-3 based chatbot system? Is there a difference in the user experience between the two methods?

RQ2 What are the effective prompt designs for GPT-3 based chatbots

- **RQ2.1** for conversation response generation?
- **RQ2.2** for creating abstractive summaries from written dialogues?

1.4 Goal

The goal of this study is to implement a short-term memory for a LLM based chatbot and study the user experience of the users interacting with the system. Two different methods are introduced.

In addition, some techniques for prompt design and abstractive dialogue summarization are studied.

1.5 Benefits, Ethics, and Sustainability

This study may be of interest to both researchers in the field of NLP and companies developing chatbot systems or working with LMs.

From an ethical perspective, there is a challenge with user privacy. The user might give out personal information without understanding how it will be stored and used in the study. To mitigate this concern, this information is provided to the users prior to starting the conversation and users must give consent before proceeding.

Another ethical challenge is that the summaries produced by the system might not accurately represent the content of the conversation and can hence affect the content and direction of the conversation. The chatbot might be less reliable than what the user expects.

This project relates to the following Sustainable Development Goals of the United Nations [**noauthor_united_nodate**]:

- 12. Ensure sustainable consumption and production patterns
- 9. Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation
- 4. Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all
- 8. Promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all

Related to sustainable consumption and production patterns (SDG 12), training LLMs requires substantial computational resources and this causes emissions. However, once trained these models can be used and fine-tuned without significant computational resources. Each time a LLM is used, an input prompt is given. The length of the input prompt affects the computational resources required, and long prompts demand more computational power, causing emissions, and costing more money. Designing methods to reduce the length of the input prompts can have effects on sustainability when scaling these models for more users, for example, ChatGPT [23] has over 100 million users.

Building systems on top of pre-trained LMs gives people access to powerful Artificial Intelligence (AI) without the need for massive resources (SDG 9). LMs and chatbots allow developing systems for education that are accessible by people with otherwise restricted access to education (SDG 4). The development of AI, and more specifically LLMs and chatbots, affects the nature and productivity of work, for example, by allowing faster information retrieval and outsourcing of tasks such as content creation and customer service and thus affecting the economic growth (SDG 8).

1.6 Methodology

A user study is conducted, and both quantitative and qualitative data are collected on the user experience of the study participants interacting with the system. This includes a survey with a questionnaire and written feedback, as well as data collected during the interaction, such as user sentiment and conversation transcripts, and conversation summaries.

Quantitative data are analyzed using statistical methods to test for statistical significance. For qualitative data, both sentiment analysis and human evaluation are used as evaluation methods depending on the data.

The study is divided into three phases; system development, user study, and analysis of results.

System Development

In the first phase, a pre-existing chatbot system developed at KTH Royal Institute of Technology is modified by implementing a short-term memory feature.

User Study

In the second phase, the system is handed out to a number of volunteer study participants, and data is collected.

Analysis of Results

In the third phase, the collected data is analyzed and results are reported and discussed.

1.7 Stakeholders

This project is done in the division of Speech, Music, and Hearing at KTH Royal Institute of Technology under the supervision of Alireza Mahmoudi Kamelabad and as part of the project Early Language Development in the Digital Age (e-LADDA). This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie Actions grant agreement No. 857897.

Other stakeholders include the target audience of this project, such as the research community and companies working with LMs and chatbot systems.

1.8 Delimitations

GPT-3 is trained with data from the internet, so it reflects the biases found in its training data [2]. These biases can be related for example to gender, race, or religion. For example, gender stereotypes are enforced in associating occupations requiring higher levels of education or hard physical labor with males, and care-related occupations with females.

Combating these biases is out of the scope of this study. The prompts in this study are designed not to exacerbate such biases. However, due to the nature of GPT-3 mimicking its training data, it is not given that the model outputs are bias-free.

The size of the user study is relatively small, consisting of 9 participants, of which each interacted with the system twice, once with each summarization method. In addition, we were present in each of the test sessions and personally knew each participant. These may have an effect on the reliability of the results and should be taken into consideration.

1.9 Outline

In chapter 2 the relevant background of the study is introduced and discussed. This includes LMs and GPT-3, text summarization, prompt design, and chatbots. In chapter 3 the chatbot system is introduced and methods for the user study are introduced. In chapter 4 the results of the study are presented and analyzed. In chapter 5 the results of the study are discussed, and limitations and future work are introduced. In chapter 6 the final remarks and conclusions are presented.

Chapter 2

Theoretical Background

In this part, the theoretical background for the study is introduced. In Section 2.1 LMs and LLMs are introduced, as well as how to utilize them for different tasks using prompt programming. Then a specific LLM, GPT-3 is introduced in detail in Section 2.2, both its architecture and capabilities relevant to this study, as it will be used both in generating the chatbot answers and the short-term memory summaries. In Section 2.3 the task of text summarization is covered in more detail. In Section 2.4 a comprehensive introduction to chatbots is given, covering chatbot design, chatbot architectures, and previous work on chatbot systems.

2.1 Language Models

LMs predict the probability of a sequence of words. This allows LMs to be used to predict the following word or words given some sequence of words. The probability for such a sequence comes from a probability distribution based on the training data used to train the LM.

LMs are based either on probabilistic methods or Artificial Neural Networks (ANNs). Only neural LMs are of interest in this study. Neural LMs assign probability values to the weights of an ANN based on the training data, and given some NL input assigns a probability for such an input. There are several use cases for LMs including generating NL text and Natural Language Understanding (NLU) [14].

In recent years we have seen a great increase in the interest towards LMs. The increase in computational power and availability of text data has resulted in larger LMs and led to the introduction of models such as Bidirectional Encoder Representations from Transformers (BERT) [5], Robustly Optimized BERT Pretraining Approach (RoBERTa) [22] and GPT-3 [2]. Such models are also referred to as LLMs.

2.1.1 Large Language Models

Recently, LLMs have changed the nature of NLP research. In the past LMs were first trained on large amounts of data and then fine-tuned with task-specific datasets. These LMs often worked well on benchmarks, but could not generalize outside of the given task [2, 17].

Scaling has not only improved the performance of LMs in many downstream tasks but has also introduced *emergent abilities* [31]. Emergent abilities can be defined as abilities that are not present in the smaller models but are present in the larger models. LLMs, such as GPT-3 [2] have shown that by increasing the size of the dataset and number of model parameters, the model can obtain generalization capabilities. This means that the same model can be used for many tasks without the need to update its weights. Instead of fine-tuning, the model adjusts to a new task by being introduced with a task-specific input prompt.

Due to their scale, LLMs are able to learn patterns in language data so well that it results in very fluent and human-like responses. These language abilities extend to many domains, such as text summarization, reading comprehension, and question answering [26].

2.1.2 Prompt Design

A prompt is a NL description of the task expected from a LM. Prompt programming [27] is the act of creating a prompt in natural language for a language model that uses natural language inputs and produces natural language outputs. This is a difficult task since natural language is not deterministic and the consequences of specific natural language prompts are difficult if not impossible to predict. Different tasks require different kinds of prompts.

The goal of prompt design is to increase the likelihood of a desired output [16]. As a response to a prompt the probability distribution over how *any person* would continue the prompt, not how *a person* would [27]. In other words, if the instructions are not clear, the model may not produce coherent responses, but a mixture of different ideas for what is expected. This implies the need for clarifying the context clearly so that there is less room for ambiguity. If there are many ways to interpret the prompt, there are many ways to respond to it, many of which are undesirable for the given task. Hence constraining behavior in prompt design may be a desirable approach; not only do we want to clearly imply what kind of output we expect, but also which kind of output we do not want to see.

Input prompts can include either a description of the expected behavior, examples of the expected behavior, or both. If the prompt only contains the description of the task, it is referred to as *zero-shot* prompt. If it contains one example of the expected behavior, it is called a *one-shot* prompt, or if more than one example is introduced, it is called a *few-shot* prompt.

2.2 GPT-3

GPT-3 is an *autoregressive* LM. Autoregressive LM is a *Feed-forward* Neural Network (NN), which predicts an output value based on the data used to train the model. GPT-3 [2] is the descendant of the LM Generative Pre-trained Transformer 2 (GPT-2) [26], and is based on the same model and architecture, but with many more parameters.

2.2.1 Architecture

Feed-Forward Network

Feed-forward NN is an ANN that consists of nodes in consecutive layers, and the nodes on each layer are connected to each node on the next layer (Figure 2.2.1). This is referred to as *fully connected network*. More specifically, the network has one *input layer*, one or more *hidden layers*, and one *output layer*. The information in the network moves only forward, not backward. A node in the network is called a *perceptron*. A perceptron is a binary classifier that takes a vector as an input and outputs a single value. The connections between the nodes have *weights*. A weighted sum of the input vector on each layer is calculated, and an activation function, for example, a binary step function or a sigmoid function, is then applied to compute the input for the next layer. Feed-forward NNs can be used in classification and regression tasks.

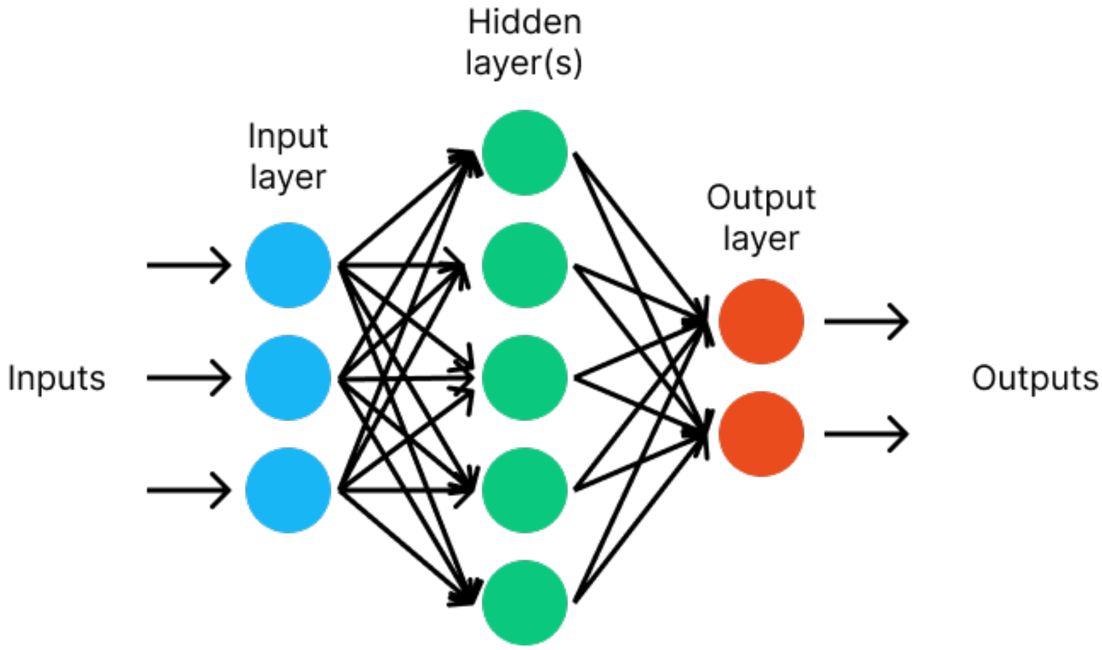

Figure 2.2.1: Example of a feed-forward network architecture

Word Embeddings

For computer algorithms to process text data, a numerical representation of the text data is needed. This concept is called *word embedding*. Word embeddings are multidimensional vectors that encompass semantic similarity between different elements in the input sequence. This means that similar words are located closer to each other in some multidimensional vector space based on some similarity metric (euclidean distance, cosine similarity). Using word embeddings allows the model to be taught which elements typically appear in similar contexts. In the pre-training phase of GPT-3, the model learned word embeddings that capture the semantic relationships between words, and this helps GPT-3 to produce a semantically coherent text.

Transformer

GPT-3 is a transformer-based model [29]. Transformers are encoder-decoder [28], or sequence to sequence, networks that use an *attention* mechanism.

Encoder-decoder networks (Figure 2.2.2) consist of two ANNs. The first network, the encoder, encodes the input vector into a fixed-length context vector. The decoder then takes this encoding and decodes it into the output. Both encoder and decoder have typically the same but reverse structure. Typical use cases for encoder-decoder models are for example text translation and text summarization.

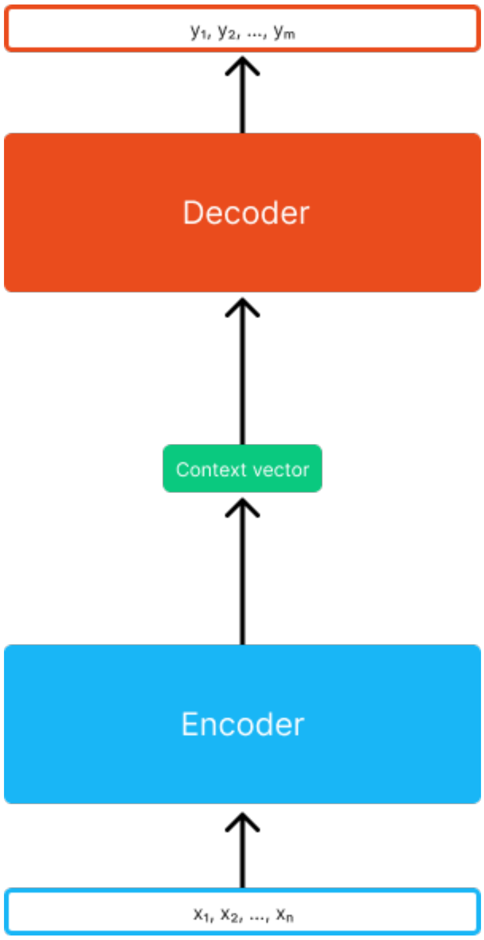

Figure 2.2.2: Encoder-decoder architecture [29]

Attention Mechanism

The attention mechanism defines which parts of the input sequence should the decoder focus on. In other words, given the context, how relevant is the ith element relative to all other elements in the input sequence. *Self-attention* mechanism defines how much each element in the output sequence is related to every other element in the output sequence. In *multi-head self-attention*, the attention module runs the computations several times in parallel. Then these attention scores are combined together to produce the final attention score.

A common formula for calculating attention scores [29] is the *scaled dot-product attention*:

$$Attention(Q, K, V) = softmax\frac{QK^T}{\sqrt{d_k}}V \tag{2.1}$$

Matrix Q represents the current element in the input sequence. Key K and value V matrices contain references to words that have been previously generated. d_k represents the dimensions of matrices Q and K. Using matrices Q, K, and V we can define which key in K is most similar to the query Q and then find the corresponding value in V. The resulting matrix contains contextualized word embeddings for each element in the input sequence, which tells us how much each element should be weighted in relation to every other element in the sequence, or how much attention each element should be given.

GPT-3 Transformer

GPT-3 uses the same model architecture as GPT-2, which is a variant of the original transformer architecture [29]. Instead of encoder-decoder architecture, it consists of 96 layers of transformer decoders [25]. An illustration of the decoder layer can be found in Figure 2.2.3.

2.2.2 Training

For GPT-3 [2] 175 billion parameters have been trained with 45TB of text data, including CommonCrawl [4] and WebText [26] datasets, internet-based books, and Wikipedia in the English language. More than 90% of this data is in English text. GPT-3 has been trained using unsupervised learning. This means that GPT-3 has learned its ground truth labels from the data itself, not from example labels, as in supervised learning.

2.2.3 Using GPT-3

The input for GPT-3 [2] is a NL prompt, that contains the description of desired model behavior, in other words, what kind of output we hope the model will produce. GPT-3 can be prompted either with zero-shot, one-shot, or few-shot prompts. After GPT-3 was released it was considered to perform best in the few-shot setting [2], more recent studies have shown that with careful prompt design zero-shot approach can in fact outperform the few-shot approach [16, 27], but the feasibility of this also depends on

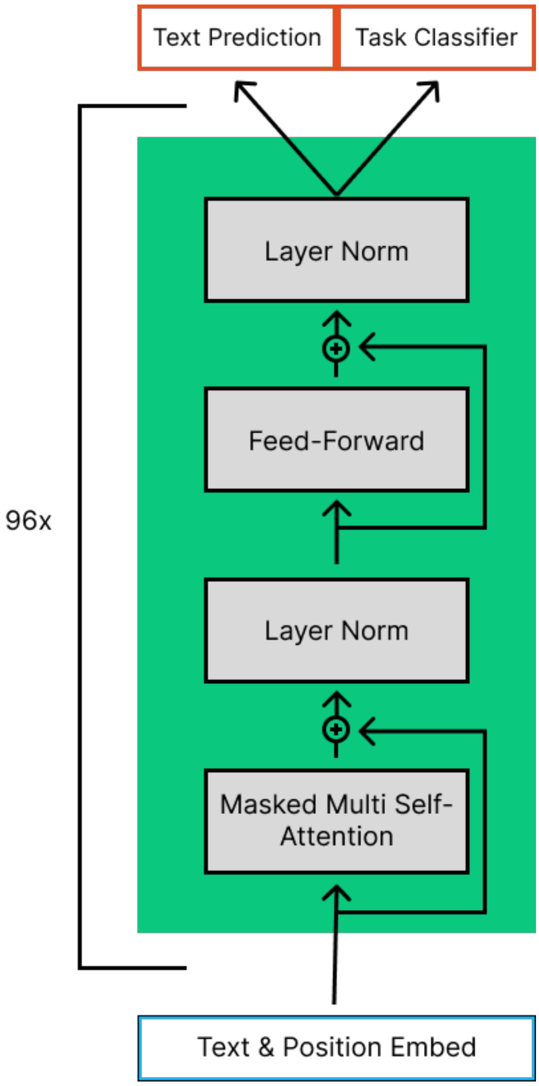

Figure 2.2.3: Transformer decoder layer [25]

the task at hand.

Next, two example prompts and corresponding outputs utilizing zero-shot and few-shot paradigms are introduced. Text summarization task can be achieved by including *Tl;dr* after the text to be summarized, and it is an example of a zero-shot prompt [2]:

Input prompt for zero-shot Tl;dr summarization

> *Teacher and student are having a conversation about food:*
>
> *Teacher: What is your favorite food?*
>
> *Student: My favorite food is pizza.*
>
> *Teacher: Why is pizza your favorite food?*

Student: Because I like cheese and tomato sauce.

Tl;dr

Output

Student's favorite food is pizza because they like cheese and tomato sauce.

For response generation for a dialogue describing the scenario and giving a few examples of the conversation turns is an example of a few-shot prompt:

Input prompt for few-shot text generation

Teacher and student are having a conversation about food:

Teacher: What is your favorite food?

Student: My favorite food is pizza.

Teacher: Why is pizza your favorite food?

Student: Because I like cheese and tomato sauce.

Output

Teacher: What type of pizza do you like the most?

Model DaVinci version 3 (text-davinci-003) was used for producing both of the examples [2].

As GPT-3 makes predictions only based on the information in the input, it is important to include all the required information in the input prompt. However, longer prompts take more time to process and have a higher monetary cost, so if time or price are critical elements, carefully designing prompts that both contain all the necessary information and are as short as possible is important.

The size of the input prompt in GPT-3 is limited to 2048 *tokens*. The act of encoding text into tokens is called *tokenization*. By tokenizing the input prompt the text is pieced into pieces of words, or tokens. These tokens do not necessarily correspond to separate words and may contain leading or trailing spaces as well as sub-words. On average one token corresponds to 4 characters in the English language.

OpenAI provides a web Application Programming Interface (API) for GPT-3 [2]. This API allows software developers to use GPT-3 in their own systems. There are four models to choose from: Ada, Babbage, Curie, and DaVinci, each model being more capable than the previous one, but the capability comes with higher cost and slower computation. The current models served by the OpenAI API are called InstructGPT [24] models, which are fine-tuned GPT-3 models trained to follow human instructions. InstructGPT models contain 1.3 billion parameters, which is 100 times less than the original GPT-3 model but produces outputs that are more preferred by humans.

As the chatbot system we are using in this study utilizes GPT-3, the direct and indirect references of capabilities and behavior of LLMs refer to GPT-3. Future research is needed to find out whether these capabilities and behaviors can be generalized to other LLMs trained with similar data as well.

2.3 Text Summarization

Text summarization is a common application area of Deep Learning (DL). DL refers to Machine Learning (ML) techniques where ANNs with multiple layers are used. *Abstractive* text summarization refers to generating a summary with the main ideas of the source text, similar to summarizing a text using your own words. *Extractive* text summarization means that specific sentences from the text are extracted as a summary [9].

In this study, we are interested in abstractive dialogue summarization. Previous literature on dialogue summarization is scarce, partly due to the lack of standardized dialogue datasets for summarization [6]. As of writing this study, we were unable to find a study that utilizes the abstractive, prompt-based approach for dialogue summarization.

A common benchmark in text summarization is news summarization [9]. Compared to for example news articles and academic papers, summarizing conversations is a more challenging task because the main ideas are often spread across multiple turns and two or more interlocutors. In addition, the lack of benchmark datasets has further limited comparing different methods for conversation summarization [6].

Most academic literature on conversation summarization defines conversations as something that happens between two or more people. In this study, we are solely focusing on dialogue, the conversation between two interlocutors. In addition, in this study, we are only interested in summarization using GPT-3 [2, 9].

A recent study compared the performance of prompt-based models to models fine-tuned on task-specific datasets [9] in the news summarization benchmark task. GPT-3 was able to outperform the fine-tuned models based on human evaluation. The study also showed that common metrics for evaluating the results of summarization are not well-suited for evaluating summaries generated by GPT-3. A more suitable evaluation method was manual evaluation, i.e. evaluation by humans. In this study, text summarization is used to summarize the conversation history so that the chatbot has a recollection of the different conversation topics and content, and can then utilize that in response generation. So far the theoretical background for the chatbot system utilized in this study has been introduced. In the next section background and different implementations of other chatbot systems are introduced to link this study to the broader literature on chatbot systems.

2.4 Chatbots

Chatbot refers to a software program that allows natural language interaction with a computer in the form of a dialogue [8, 12]. This dialogue can either be mediated as text or as spoken language if an Automatic Speech Recognition (ASR) and speech synthesis system is integrated into the conversational agent. For meaningful conversation, the conversational agent must also have NLU capabilities [13] to be able to analyze the interlocutor input and respond to it in natural language.

The first chatbot called ELIZA was developed several decades ago, in 1966 by Joseph Weizenbaum [12]. More recent chatbots include Gunrock [33] and Gunrock 2.0 [19] and ChatGPT [23]. Chatbots have many use cases. Some common use cases for chatbots include question-and-answer, website customer service, personal assistance (e.g. Amazon Alexa, Siri), and education [8, 18].

Chatbots may differ both in terms of whether the interaction with the chatbot is short-term, for example, web service assistant, or long-term, for example, studying a foreign language or acting as a virtual friend, and whether the interaction is user-driven or chatbot-driven. These aspects are important to consider when designing a chatbot system for a specific use case. In long-term interaction, it is feasible to consider ways to improve the user experience by taking into account user-specific information and previous parts of the conversation [8]. In this study, we are interested in the long-term interaction with the chatbot across two or more different topics. The system proposed is user-driven, as the user decides the topic of discussion and steers the conversation.

2.4.1 Designing Chatbots

Literature for designing chatbot systems for different use cases is somewhat limited [8]. Larger technology companies, such as Google and Amazon do provide guidelines for different aspects of chatbot design, such as UX design and conversational interaction design. A common goal in designing conversational chatbots is to make them appear human-like [16].

2.4.2 Chatbot Architectures

Chatbots can be divided into rule-based [7, 10, 15, 18, 19] and LM-based [16, 23] systems. The difference is that rule-based systems use a decision tree to determine output, whereas LM-based systems predict the output using the model weights.

Rule-based systems typically consist of multiple modules that are responsible for different areas, for example, NLU, mini-skills, and knowledge storing and extraction [19]. The benefit of rule-based systems is that the creator of the system has more control over its behavior as the conversation flow can be constrained [16]. The downsides include a limited ability to have longer conversations and difficulties adapting to unexpected input [7]. Interaction with rule-based chatbots may often be repetitive and impersonal [18]. To mitigate this, some recent work suggests introducing personalization to the system, which can adapt to the individual characteristics of the user and provide a more individualized experience [18, 19].

Some chatbots utilize a rule-based approach, but also include fine-tuned LMs as modules. So the distinction between rule-based and LM-based approach is not necessarily clear. In this study when referring to LM-based chatbots we are referring to systems that are not rule-based, and that use LLMs as the main conversation engine due to the generalization capabilities such models have. Building chatbot systems on top of LMs makes simpler architectures possible. This is because a single LM can be responsible for many of the functions that would require separate modules in rule-based systems.

2.4.3 Large Language Models and Chatbots

The availability of high-quality conversational data has been a major bottleneck for the development of chatbots [19]. The introduction of LLMs has mitigated the need for conversation-specific datasets as LLMs, such as GPT-3, tend to perform relatively well empirically in the conversational context [2, 16].

The limitations of LLMs are the unpredictability of the output, the tendency to get stuck in loops, and the unreliable factual content of the output [2]. LLMs also struggle with answering questions that require multi-step inference. The problem with non-factual content and multi-step inference is in many cases not due to the model's lack of knowledge, but due the method of extracting such knowledge. So these problems can be mitigated to some extent with careful prompt design [30, 32]. Compared to the rule-based approach, where the conversational agents often relied on smaller datasets and knowledge bases [19], LLMs tend to be more flexible in adapting to new topics and opinion-related conversations due to their generalization capabilities. These generalization capabilities also provide interesting opportunities for chatbot development as we can design chatbots for different purposes by programming suitable prompts.

In the academic literature, LLM-based chatbots have not been researched extensively. One example is a personal coach for mental well-being that utilizes GPT-3 [16]. The study investigated how to design prompts for GPT-3 so that the empirical performance of the chatbot is optimized. The study suggests using three dimensions in the prompt design:

1. *Identity* - describes the identity of the chatbot
2. *Intent* - describes the intention of the conversation
3. *Behavior* - describes the expected behavior from the chatbot

This prompt design proved successful, despite some problems with the conversation getting stuck in a loop. The study also suggested that the zero-shot approach can outperform the few-shot approach if the prompt is designed well, further supporting the discoveries of Reynolds et al. (2021) [27]. This prompt design is used in this study for the chatbot response generation, but with some further improvements that will be introduced in the next chapter.

2.4.4 Previous Research on Chatbots

Next, a brief introduction to some popular or interesting chatbot systems will be given.

Sounding Board

Sounding Board [7] is the 2017 Amazon Alexa Prize winner. The interaction with the system is ASR based, and it is capable of having conversations on various current topics, including sports, politics, and entertainment. The goal of the system is to sustain long conversations, and it averaged a duration of 10:22 in the competition in 2017. The system is described as both *user-centric*, meaning that users control the conversation, and *content-driven*, meaning that it uses a growing collection of content to be able to provide meaningful output for the conversation.

Typical to a rule-based system, Sounding Board's architecture consists of separate modules for NLU, answer generation, and dialog management. Dialog Manager (DM) is a module responsible for managing the conversation state and conversation flow. Sounding Board uses a knowledge graph for generating content on different topics. A knowledge graph is a structure where related entities are linked with each other, making relevant information accessible fast.

Gunrock

Gunrock [19] was the Amazon Alexa Price winner in the following year 2018. Gunrock is also an ASR based chatbot. Gunrock has a modular architecture consisting of NLU, a DM, and answer generation modules, as well as several knowledge bases equipped with information regarding factual, current, and personal information. A personal information database is referred to as *Persona Backstory*, and its purpose is to store information about Gunrock's personality and background so that it is equipped to answer questions about itself.

Some key findings with Gunrock were that both longer sentences from the user and questions about Gunrock's backstory correlated positively with the user experience. Another finding was that combination of personal opinions and stories, and factual information in the conversational flow had a positive effect on the user experience.

Gunrock 2.0

Gunrock 2.0 [19] is an adaptive chatbot built on top of Gunrock [33]. Gunrock 2.0 can have conversations over various topics including games, fashion, movies, and travel. Typical of a rule-based system and similar to Sounding Board [7] and Gunrock [33], it consists of several modules, such as NLU module, answer generator module, DM and a knowledge base, including the Persona Backstory similar to Gunrock. The dialog manager of Gunrock 2.0 is a hierarchical system divided into high-level and low-level dialog management. The high-level dialog manager is responsible for recognizing user intent and referencing the previous dialog module state to select the favorable dialog module. The low-level module is then responsible for generating a response to the user.

In Gunrock [33], the system itself had a sort of personality through the Persona Backstory, but it could not adapt to different users. Gunrock 2.0 includes personalization capabilities, and is able to adapt to the user by learning from the user inputs, for example by defining the user's gender and then adapting its vocabulary to match that, for example, ”cologne” vs ”perfume”.

Edubot

Edubot is a web-based chatbot [18]. It follows the design of Gunrock 2.0 but is adapted to the context of second language learning. To do this, the system uses the information in the user's profile to adapt to the user's language proficiency. The idea behind this is to the experience feel more personalized to the user and increase user engagement. The major focus point in the study is correcting user mistakes. Although interesting, it is out of the scope of this study. Edubot is a good example of applying a more general conversational chatbot system to a specific use case.

ChatGPT

During conducting this study OpenAI released a new chatbot, ChatGPT [23]. It is a conversational chatbot system that can have a discussion on almost any topic. Apart from other chatbots introduced in this chapter, ChatGPT is a LLM-based chatbot. ChatGPT is based on InstructGPT [24]. The difference to InstructGPT is that ChatGPT is specifically designed for having conversations instead of being a general purpose LM. It is currently in a research phase and can be accessed through an online interface

[3].

2.5 Summary

GPT-3 can perform a variety of tasks, including text summarization and predicting the next turn in a dialogue by designing suitable input prompts. This allows us to design chatbots that do not require separate modules for NLU, text generation, knowledge bases, and so on. We can cover most requirements with a single LLM and adjust its use by prompt programming. Due to the abstractive summarization capabilities of GPT-3 it can be used for producing the summaries that act as short-term memory and will be used for the prompts for answer generation. GPT-3 is able to produce natural and human-like responses, which makes building chatbots on top of it a good option.

Chapter 3

Methods

3.1 The System

In order to explore the effects of a short-term memory implementation on **RQ1** the first version of a chatbot system initially designed for language learning was repurposed for this study. The functionality for grammar correction was removed, and the following features were implemented:

1. A module for summarizing conversation history
2. Prompt generator module that utilizes prompt description, summary, and conversation history
3. Data collection module for collecting summaries, conversation transcripts, and additional metrics
4. Implemented a message pop-up that in mid-conversation suggests new conversation topics to test how switching the topic affects the conversation flow given that the system remembers the previous conversation topic

The system is a web-based chatbot application built with Python and JavaScript, and it utilizes the Python Flask framework. It uses OpenAI's [2] GPT-3 API for dialogue summarization and response generation. The reason for using the GPT-3 API is its accessibility and high performance, as well as, the vast amount of academic literature.

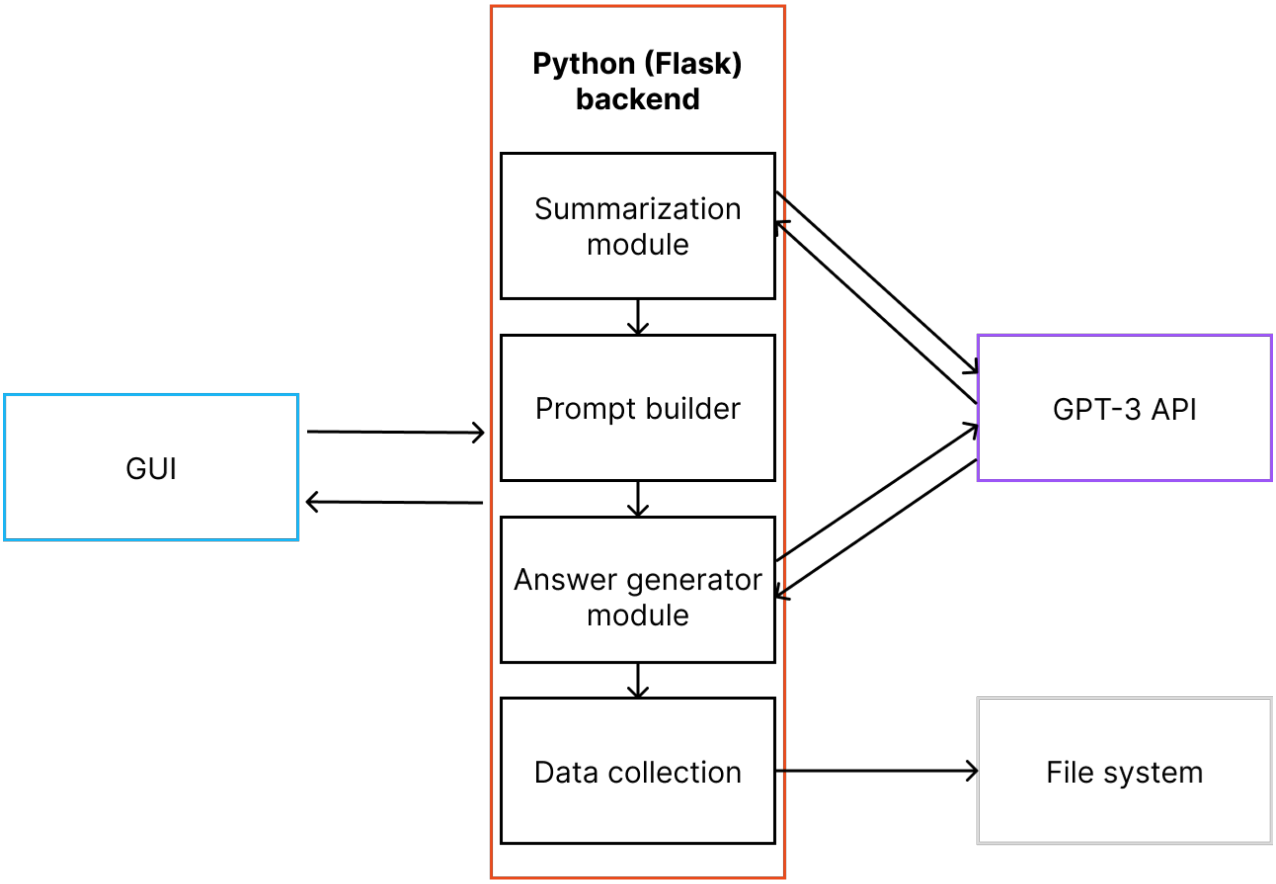

Figure 3.1.1: System architecture

3.1.1 Language Model

The system uses LM GPT-3 [2]. More particularly DaVinci model version 3 (text-davinci-003) through OpenAI API for both summarization and answer generation tasks. It is accessible through the OpenAI web API. It is currently the most sophisticated GPT-3 model available, but also the slowest and most expensive. The *max_tokens* parameter was set to 100 for the summarization task, and to 50 for the text generation task. The max_tokens parameter represents the upper bound for the output length. The remaining model parameters are not modified and are the default values.

3.1.2 User Interface

Users interact with the system through a web interface. The interface contains the current conversation, a field for user input, and send button. The chatbot responses are inside red areas and user responses are inside green areas (Figure 3.1.2).

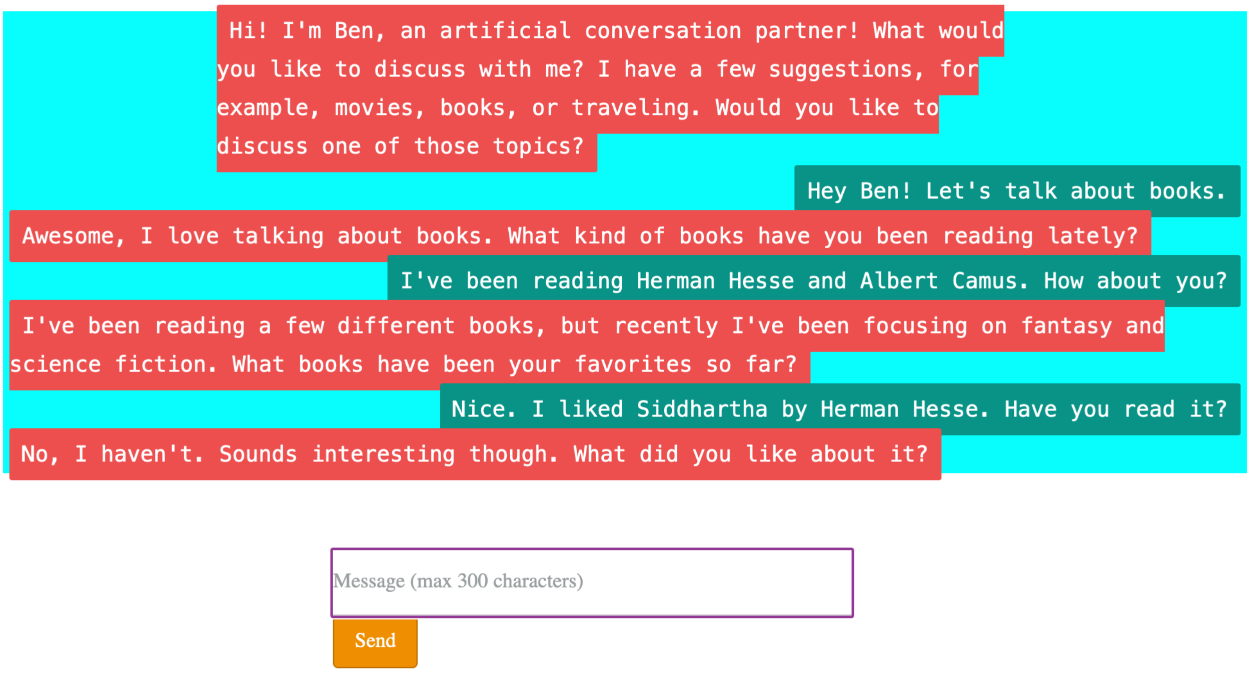

Figure 3.1.2: Chatbot user interface

3.1.3 Backend

The system backend is a web server built with Python that handles the input and output data, as well as the data collection. The backend is responsible for generating the conversation summaries, GPT-3 prompts, and chatbot answers.

3.1.4 System Interaction

Figure 3.1.3 describes the system interaction flow.

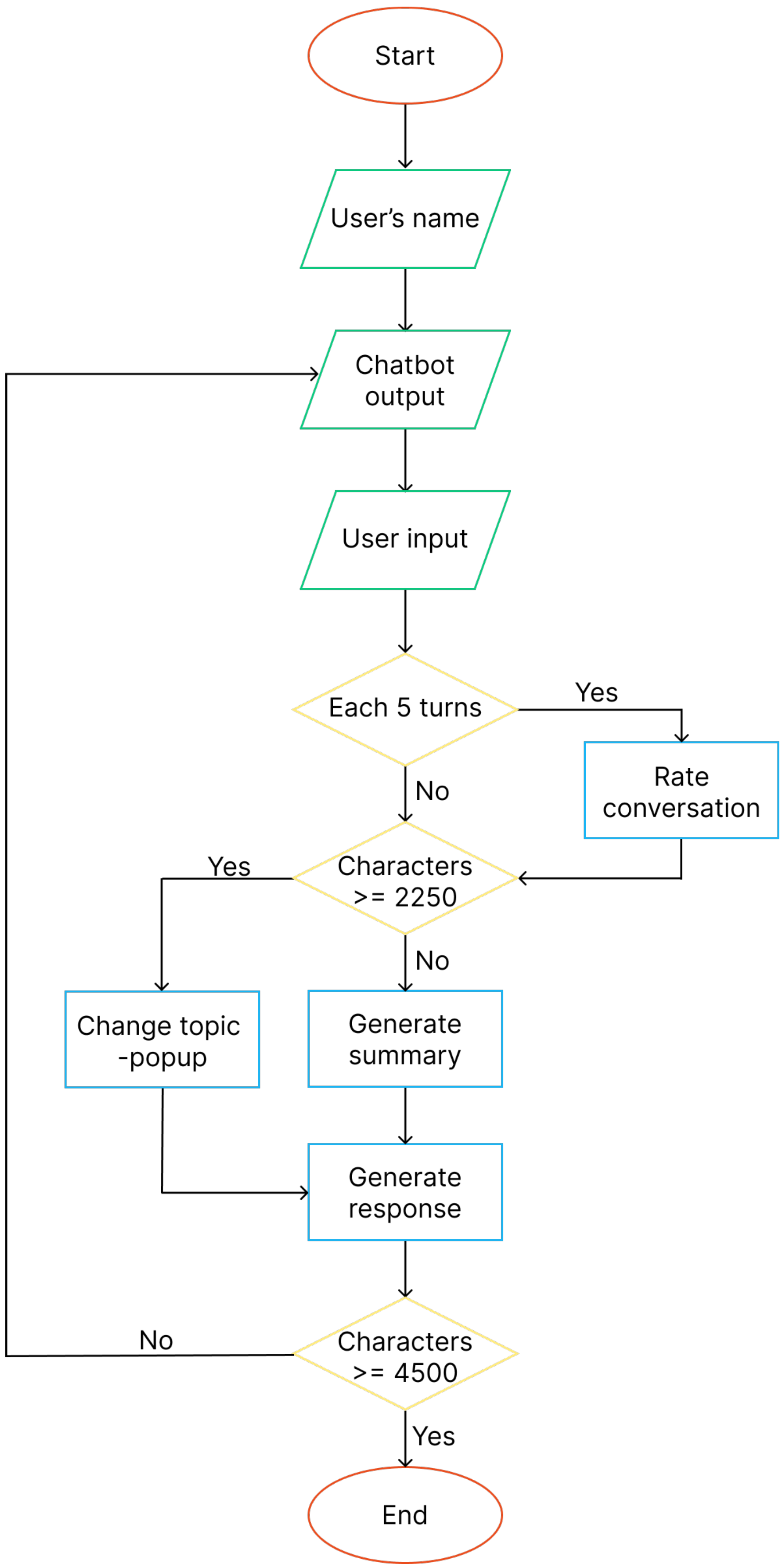

Figure 3.1.3: Chatbot system interaction flow

The interaction with the system includes multiple rounds of outputs from the chatbot and inputs from the user. Next, the whole interaction flow is introduced step by step.

Start

The conversation begins.

User's Name

The user's name is asked for by a pop-up prompt at the beginning of the conversation. This name is then stored as a parameter and used in the GPT-3 prompts to improve the user experience by allowing the chatbot to know the user's name.

Chatbot Output

The interaction starts with the chatbot outputting a pre-defined message (Figure 3.1.2). After that, on each turn, the next chatbot response is generated by the system and displayed in the user interface.

User Input

The user writes a text input and submits it (Figure 3.1.2). The maximum input length is limited to 300 characters.

Each 5 Turns

After every five turns the user has submitted an input the system asks the user to rate the conversation so far. This is a binary option between good and bad (Figure 3.1.4). This data is collected and used in analyzing the study results for user sentiment.

Characters >= 2250

Mid-conversation, after 2250 input characters, the user will get a message to change the topic of discussion (Figure 3.1.5).

To reinforce changing the topic, the string ”Let's talk about music now. What are your favorite bands?” is automatically set to the input field, but can be modified before submitting it. The purpose of this is to ensure that the conversation does not stay for

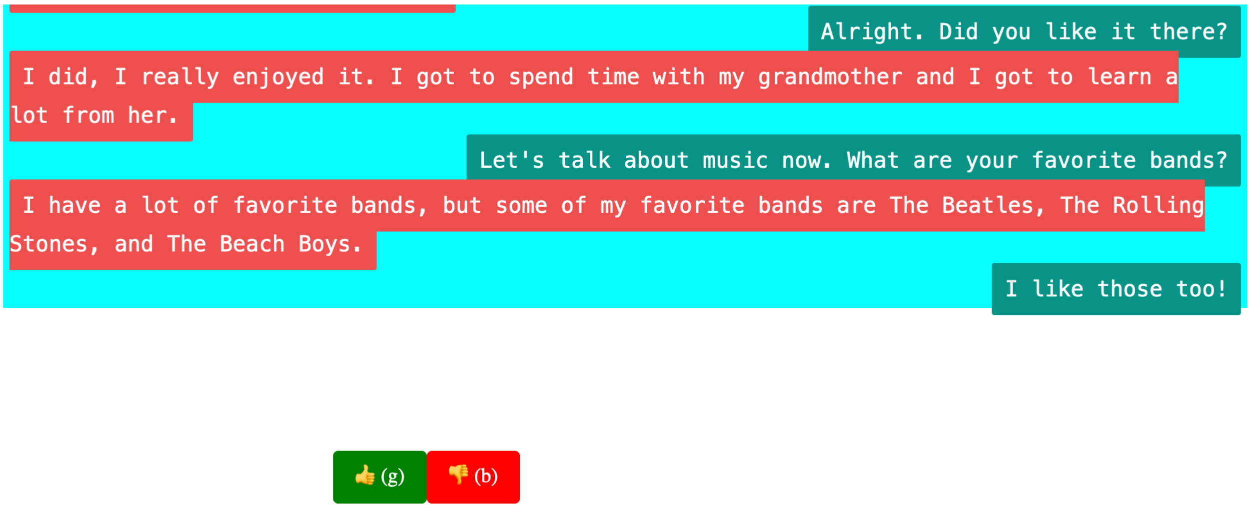

Figure 3.1.4: Rating the conversation

too long on a single topic and to challenge the summarization by introducing variation to the conversation.

Generate Summary

Based on the user input, previous conversational history, and the previous summary (for FullContext method), the system generates a summary of the conversation using GPT-3 (Figure 3.1.1). The summary is updated dynamically on every turn so only one turn from the chatbot and one turn from the user is introduced to form the new summary. A detailed description of summary generation can be found in Section 3.2.1.

Generate Response

A new prompt is created as the combination of the prompt template, the summary, and previous turns. Then GPT-3 is used to generate the system output (Figure 3.1.1). Detailed description can be found in Section 3.2.2.

The resulting output is processed so that only full sentences, in other words, word sequences ending with ".", "!" or "?" are included, as there is no guarantee that GPT-3 outputs are full sentences. This ensures that the output for the user appears complete.

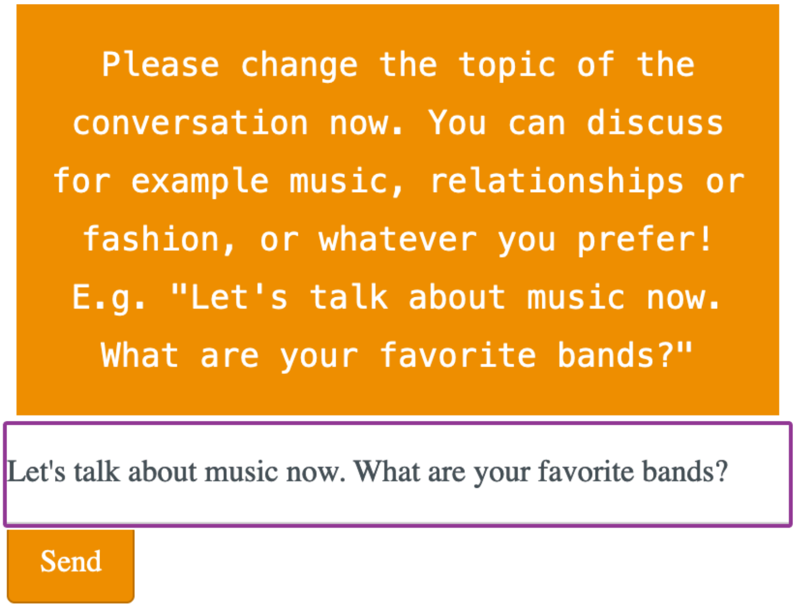

Figure 3.1.5: Message pop-up suggesting topic change

Characters >= 4500

The system checks whether the limit of 4500 characters is reached. The conversation ends when 4500 characters have been submitted by the user, otherwise, the conversation continues.

End

The conversation ends and the user is given a link to the online survey (Section 3.3).

3.2 Prompt Design

In this study GPT-3 [2] is used both for summarizing the conversation to produce an abstractive summary for context reference and for generating the chatbot answers (Figure 3.1.1). In order to answer **RQ2**, which is about designing effective prompts, we will discuss the approaches to prompt design for both tasks.

3.2.1 Summarization

For summarizing the conversation a zero-shot prompt is created by injecting the conversation history into the prompt (**RQ2.2**) (Section *Generate summary*). The length of the input prompt affects both the quality of the output, as more data contains more information for producing the output, but also affects the processing time of the algorithm which causes a delay in the user interaction. We have tried to find a balance between these two to preserve both the output quality and minimize the latency. This was achieved by several trials between different prompts and by evaluating the results with the human eye, as we were unable to find any benchmark results from the academic literature.

Users are asked for their names at the beginning of the conversation (Section *User's name*). From this point forward users' names are referred to as *<USER>* in the examples to protect the privacy of the study participants. The name of the AI agent is *Ben*. For constructing the summary prompt, two methods, referred to as *Limited Context method (LimContext)* and *Full Context method (FullContext)* from now on, are considered and compared.

The prompt used for summarization is

> *Summarize the following conversation between two friends:*
>
> *<LimContext method> / <FullContext method>*
>
> *The summary should describe the context of the conversation in the past tense.*

Next, we will show examples of both methods and corresponding summaries for both methods:

LimContext method: Previous, a maximum of, 16 turns in the conversation are injected in the summary prompt.

Example:

> *Ben: What would you like to discuss with me? I have a few suggestions, for example, movies, books, or traveling. Would you like to discuss one of those topics?*
>
> *<USER>: Hello Ben! Let's discuss traveling*
>
> *Ben: Great idea! What have you found to be the most fascinating place that*

> *you've ever traveled to?*
>
> *<USER>: I think Sumbawa was great! It's an island in Indonesia. How about you?*
>
> *Ben: That sounds beautiful! What aspects were so fascinating about your experience in Sumbawa?*
>
> *<USER>: It's one of the best places for surfing.*
>
> *Ben: Wow, that sounds like a great time! Did you go alone or with someone else?*
>
> *<USER>: I went with a friend.*
>
> *Ben: Was it just the two of you or were there any other people around?*
>
> *<USER>: Well yes, there were other people around, but we did not know them*
>
> *Ben: Cool, so did you take advantage of the great surfing opportunities or did you explore some of the local towns?*
>
> *<USER>: We did both, but focused on surfing. Have you been to Indonesia?*
>
> *Ben: No, never been there unfortunately. What was it like, would you go back?*
>
> *<USER>: I would definitely back, it was great!*
>
> *Ben: What specifically did you like about it? Was the surfing really good?*
>
> *<USER>: I liked the food, people, weather, nature and the surf.*

Example summary for LimContext method:

> *Ben and <USER> had a conversation about their travels. <USER> told Ben that he had been to the island of Sumbawa in Indonesia and Ben asked him what made the experience so fascinating. <USER> mentioned that the destination was great for surfing and that he had gone along with a friend. Ben asked if they had taken advantage of the surfing opportunities and had also explored the local towns. <USER> said they did both but had focused on surfing.*

FullContext method: The previous summary and the previous, maximum of, 12 turns in the conversation are injected into the summary prompt.

Example:

Two friends discussed their experience travelling to Tibet and saw Dalai Lama's old home and temple, which was strictly controlled by Chinese officials. They engaged with a tour guide and found out that Chinese authorities are trying to force the locals to speak Chinese instead of Tibetan language. The Tibetans felt sad and angry, but couldn't demonstrate their feelings as Chinese government had a tight control of the area.

Ben: That sounds like it must have been a great experience. What kind of things did you learn from him?

<USER>: That for example Chinese officials are trying to force everyone to speak Chinese instead of Tibetian local language.

Ben: Wow,that is outrageous. How do the Tibetans feel about that?

<USER>: They are really sad and angry. But they can't really demonstrate their feelings as China has a really tight grip of the whole land.

Ben: That sounds really unfair. Is there anything that can be done to help the Tibetans?

<USER>: We could try to impact the Chinese government for example by having pro Tibet demonstrations. But to be honest its really hard to help Tibetans at the moment.

Ben: That's really unfortunate. What do you think would be the best way to help the Tibetans?

<USER>: Like I just said, I think by having for exampel pro Tibet demonstrations

Ben: That's a great idea. Do you think by having enough people come out for these demonstrations it would be able to create a positive effect on the Chinese government?

<USER>: Not really as China is so large country and it's not easy to have an impact to it's policies.

Ben: That's true. So what do you think we can do then to help bring awareness of the situation in Tibet?

<USER>: We should have more news articles about the situation and we all

should just talk more about the situation over there.

Example summary for FullContext method:

Two friends discussed their experience travelling to Tibet and saw Dalai Lama's old home and temple, which was strictly controlled by Chinese officials. They engaged with a tour guide and found out that Chinese authorities are trying to force locals to speak Chinese instead of the Tibetan language, which made the locals feel sad and angry, but unable to demonstrate their feelings due to the tight control of the area.

3.2.2 Answer Generation

Common principles for conversational prompt design are to describe the identity, intent, and behavior [2, 16] of the AI agent to guide the content and tone of the answers (**RQ2.1**). It has been pointed out that GPT-3 can understand analogies, and we can proxy certain intent and behavior through a character or a characteristic that possesses such intention or behaves in a certain manner. Such a feature is called a *memetic proxy* [27]. This means that instead of specifying certain intent or behavior explicitly, for example, "Explain things in detail and in an understandable way, and tell me if I'm wrong", we can assign the identity of a teacher. Or "be kind, compassionate, and supportive", which are behaviors often expected from a friend. Because we tend to affiliate certain characteristics in our cultural consciousness to identities, these characteristics transfer to the language models from the training data and affect the output of the language model. There are also use case-specific intents and behaviors that are not transferred through memetic proxy. In these cases, we can explicitly state the intent or behavior in the prompt.

In the study, the identity of the chatbot is a friend. In the prompt design, the idea of the memetic proxy is applied, and then a specific scenario where the friend operates is described (Section *Generate response*). This extends the intent of a friend into our scenario.

Each prompt starts with the scenario description containing the identity:

A friend named Ben has a discussion with <USER>. <USER> decides the conversation topic and Ben and <USER> discuss this topic. Ben asks questions often.

And this is then complemented with the initial message:

What would you like to discuss with me? I have a few suggestions, for example, movies, books, or traveling. Would you like to discuss one of those topics?

Identity	**Scenario (extended intent)**
Friend	A [Identity] named Ben has a discussion with <USER>. <USER> decides the conversation topic and Ben and <USER> discuss this topic. Ben asks questions often.

Table 3.2.1: Prompt design used in the study and the relationship between the Identity and the scenario. User is asked for their name before the conversation and the name is injected into <USER> in the prompt

Each conversation starts with the initial prompt to which the user is expected to answer. The complete prompt for generating the chatbot answers is constructed from the following four parts:

1. Scenario description (Table 3.2.1)
2. The summary of the previous conversation based on LimContext method or FullContext method
3. Previous, maximum of, 4 conversation turns, i.e. 2 turns from the chatbot and 2 turns from the user
4. The prompt ends with *"Ben: "* to illustrate that the expected output is predicting what the AI agent would say next

An example of a generated prompt after 8 conversation turns:

A friend named Ben has a discussion with <USER>. <USER> decides the conversation topic and Ben and <USER> discuss this topic. Ben asks questions often.

Ben and <USER> had a conversation about their travels. <USER> told Ben that he had been to the island of Sumbawa in Indonesia and Ben asked him what made the experience so fascinating. <USER> mentioned that the destination was great for surfing and that he had gone along with a friend. Ben asked if they had taken advantage of the surfing opportunities and had also explored the local towns. <USER> said they did both but had focused on surfing.

Ben: No, never been there unfortunately. What was it like, would you go back?

<USER>: I would definitely back, it was great!

Ben: What specifically did you like about it? Was the surfing really good?

<USER>: I liked the food, people, weather, nature and the surf.

Ben:

The user is responsible for deciding the conversation topic and is expected to inform the chatbot regarding which topic they prefer to discuss.

3.3 Data Collection

Each participant has two conversations with the chatbot, one with each summarization method implementation. The conversation ends after 4500 input characters from the user.

All conversation transcripts are collected, as well as three summaries of each conversation. This information is stored in the local file system (Figure 3.1.1). Time stamps of each user input are included in the transcript logs. Every five turns the user is asked to rate whether the conversation is going well or not, and these user sentiments are saved (Section *Each 5 turns*). At the end of each conversation, the participants are asked to fill out an online survey through Google Forms. The link to the survey can be found in Appendix A. The survey consists of demographic information, namely gender and age range, multiple choice questions, and text feedback.

3.3.1 Questionnaire

A part of the online survey is the Chatbot Usability Questionnaire (CUQ), a questionnaire specifically designed for chatbot usability studies from Ulster University [11]. CUQ is comparable with a more well-known usability questionnaire, System Usability Scale (SUS) [1], which is a usability questionnaire intended for studying conventional computer systems. CUQ has been specifically designed for measuring the usability of chatbot systems, which is the reason it was selected for this study.

CUQ consists of sixteen questions with a Likert scale, for reference see Table 3.3.1. The questions relate to either the positive (odd-numbered questions) or negative (even-numbered questions) aspects of the chatbot system. CUQ gives a single score out of

100 and is comparable to the scoring of SUS. This score can be calculated with the equation 3.1.

$$CUQ = \left(\left(\sum_{n=1}^{m} 2n - 1 \right) - 5 \right) + \left(25 - \left(\sum_{n=1}^{m} 2n \right) \right) \times 1.6 \tag{3.1}$$

where $m = 16$, which is the total number of questions, and $n = individual\ question\ specific\ score$.

In addition to the CUQ, one more multiple choice statement was added to the survey: *"I would be willing to interact with the chatbot again"*. This score was not calculated together with CUQ metrics but was evaluated separately. Similar metric has been used in several other chatbot studies, such as with Sounding Board [7], Gunrock [33], and Gunrock 2.0 [19].

Rating	**Meaning**
1	"Strongly disagree"
2	"Disagree"
3	"Neutral"
4	"Agree"
5	"Strongly agree"

Table 3.3.1: Likert scale

3.4 Evaluation

In this study both the conversation summaries used for the input prompts and the conversations are evaluated.

3.4.1 Summaries

Typical methods for summary evaluation in machine learning are using automatic metrics, such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [20] and BERTScore [34]. However, it has recently been observed that abstractive summaries produced by GPT-3 tend to get lower scores in automatic evaluation compared to State of the art (SOTA) summarization techniques, but outperform these in human evaluation [9]. Hence using existing automatic metrics does not provide a reliable evaluation of the summaries produced by GPT-3, and we will take a qualitative approach of human evaluation; comparing generated summaries and selecting the methods which produce the best result according to the human eye.

3.4.2 Conversations

For evaluating the conversation with the chatbot both quantitative and qualitative data is collected. This data includes participants themselves evaluating the conversation every five turns and by filling out a questionnaire after the full conversation as well as human evaluation of the conversation transcripts.

We do not expect the study participants to have a unified perception of what is considered a preferable conversational experience and what is not. It is possible that some participants would simply rate the experience highly because they are impressed or excited by the capabilities of the chatbot system, or due to *novelty effect* [12], while others might rate the conversations based on other standards such as text chatting with a human. To mitigate such concerns, each participant is asked to use as a baseline the assumption of text chatting with a human interlocutor.

3.4.3 Quantitative Data

Quantitative results evaluated in this study include ratings of how well the conversation is going measured as the positive user sentiment, the duration of each conversation, the average user input length in each conversation, and the survey results, consisting of the CUQ, and the users' willingness to re-engage with the chatbot.

Average input length is a method for measuring the engagement of the participants. Short inputs on average are expected to imply lower engagement, frustration, or bad-quality output from the chatbot, whereas longer inputs are generally expected to imply that the participant is actively involved in the conversation. Earlier research has found a positive relationship between longer average input length and positive user experience [33]. These assumptions are further validated using the questionnaire and user feedback.

The positive user sentiment is calculated by the formula

$$positive\ sentiment = (positive\ ratings/all\ ratings) * 100 \tag{3.2}$$

As the user gives either positive or negative ratings throughout the conversation (Section *Each 5 turns*), the $positive\ ratings$ parameter is the number of the positive ones, and $all\ ratings$ parameter is the number of all ratings.

We also created a *compound metric* (Equation 3.3) in order to encompass the combined information of CUQ scores, willingness to re-engage with the chatbot, and positive sentiment. Each individual metric is normalized within the range of $[0, 1]$. Then a weighted sum is calculated.

CUQ score (w_C)	**Willingness to re-engage with the chatbot** (w_W)	**Positive sentiment** (w_S)
0.4	0.3	0.3
0.5	0.25	0.25
0.6	0.2	0.2
0.7	0.15	0.15

Table 3.4.1: Weights for the compound metric

Four combinations of the weights are tested (Table 3.4.1). The weight for the CUQ scores is set higher than the other two metrics due to its larger relative explanatory power. The weights for both willingness to re-engage with the chatbot and positive sentiment metrics are estimated to be equal, representing our subjective estimate of the approximate explanatory power over the user experience for each metric.

$$score_i = w_C \times NC_i + w_W \times NW_i + w_S \times NS_i \tag{3.3}$$

Where i is the participant, NC is the normalized CUQ score, NW is the normalized willingness to re-engage with the chatbot value and NS is the normalized positive sentiment value. Weights w_C, w_W and w_S represent the corresponding weights for each metric.

For the statistical analysis of the metrics, *Wilcoxon Signed-Rank test* and *Sign test* are used to determine whether the results are statistically significant.

3.4.4 Qualitative Data

Human evaluation

For assessing the quality of the summaries and conversations the summaries and transcripts are evaluated by a human eye, which means that they are inspected visually and compared to each other across both methods based on a subjective view of quality [9]. Due to the limited scope of the study, the evaluation is conducted solely by the author and is limited to comparing the similarities and differences of the example results for the two methods. User feedback is used to support this analysis.

Feedback

At the end of the questionnaire, short text feedback from the participants is collected. This feedback is expected to contain one or more sentences and follow the instruction *"Please give feedback on your experience"*.

The feedback for both methods is analyzed using GPT-3 [2] by creating summaries from the feedback and these summaries are then read and analyzed by the author alongside the original feedback to ensure the quality and introduced to the reader, and by conducting sentiment analysis [21] on the feedback for both methods. GPT-3 model DaVinci version 3 is used for both tasks. For summarization Tl;dr summarization is used with maximum tokens of 256. For sentiment analysis the following prompt is used:

> *This is a sentiment classifier:*
>
> *<FEEDBACK>*
>
> *Sentiment:*

To evaluate the consistency of the sentiment analysis, the sentiment analysis was conducted 10 times for the feedback for both methods. The output will be either $positive$, $neutral$, or $negative$. The input for the sentiment analysis is the full feedback corpora for both methods. The reason for this is to investigate whether there is a difference between the sentiment of the feedback for the two methods.

Chapter 4

Results and Analysis

In this chapter, we will present and analyze the results of the user experiments. A total of 9 volunteer participants took part in the study. We personally knew each study participant, as the participants were recruited from close social circles. The study was conducted on the same computer in a physical location. It proved difficult to recruit more participants in the scope of this study, as volunteers were scarce and each chatbot interaction session took approximately $1-2$ hours which resulted in approximately $2-4$ hours time commitment from each study participant and from us, as it was required that we were providing the computer, system, and guidelines for the participants.

Each participant had two conversations with the chatbot and answered two questionnaires. One conversation utilized LimContext method and another utilized FullContext method. The order of appearance of these two methods was determined randomly for each participant, and the participants were not aware of which method was in place.

4.1 User Study

4.1.1 Demographics

Out of the 9 participants, 7 were male and 2 were female. Each participant was a native Finnish speaker. This means that none were native English speakers, but each participant had high fluency in English. The age distribution was between 20 and 69, with most participants ($N = 5$) belonging within the range of $30-39$ (Table 4.1.1).

Characteristic	**Value** *(N=9)*
Gender	
Male	7 (77.8%)
Female	2 (22.2%)
Age	
20-29	2 (22.2%)
30-39	5 (55.5%)
40-49	0 (0%)
50-59	1 (11.1%)
60-69	1 (11.1%)

Table 4.1.1: Study participant demographics

4.1.2 Transcripts

Below are 10-turn excerpts from two conversations during the study, one using LimContext method and another using FullContext method. The transcripts are presented to illustrate both methods in practice.

LimContext Method

🤖 *Right, there could be slower decision making, although I think it could also be beneficial, in that it gives people more time to think critically and with intention. Do you have any other thoughts on the potential benefits of workplace democracy?*

🧑 *No. Do you?*

🤖 *Well, it could also foster creativity and innovation by allowing diverse experiences and perspectives, which helps to create an environment that encourages people to challenge the status quo and think more broadly.*

🧑 *Yes. But I do not think my boss would like it. Should we make a revolution?*

🤖 *Revolution isn't always necessary, I think it really depends on the context and management style. That being said, a workplace democracy could be a good way to revolutionize a hierarchical organization by fully embracing inclusivity and ownership of decisions.*

🧑 *I fully agree! Is there democracy at your workplace?*

🤖 *Unfortunately no, but I do think it's something the company should consider. What do you think the potential downsides of workplace democracy might be?*

👦 *We already went through that. Do you think there are any downsides?*

🤖 *I think one of the main potential downsides is that decisions could take longer, but if the goal is to make well thought-out decisions that represent all voices, then it might be worth the extra time. What do you think?*

👦 *I agree.*

The conversation appears quite fluent. The system asks questions on three out of five turns. Also, the system repeats itself by talking about a topic that was already covered in the conversation. These are typical behaviors for the chatbot according to several users' feedback (Section 4.1.5).

FullContext Method

🤖 *What do you mean?*

👦 *I mean that it things that we find of importance are very personal matters, and I am from the Nordics, we seldomly talk about such things*

🤖 *What do you think Albert Camus was trying to express in The Fall?*

👦 *The stuff we talked about earlier*

🤖 *Yeah. What specifically?*

👦 *The absurdity of life*

🤖 *How do you think The Fall speaks to joy and self-care?*

👦 *It does not really*

🤖 *How does music provide you with an outlet for joy and self-care then?*

👦 *Rhythm and beautiful melodies are pleasant for the ear*

Similar to the previous transcript for Method1, the system asks a lot of questions, this time on each turn. This particular conversation excerpt shows the system relating to the past conversation. The response generation (*Generate response*) prompt contains two turns from both user and the system. The transcript illustrates how the system is able to bring up *music* in the conversation even though it was discussed earlier in the conversation.

4.1.3 Questionnaire Results

Chatbot Usability Questionnaire

Question	LimContext method	FullContext method
The chatbot's personality was realistic and engaging	3.22 (1.2)	3.44 (0.88)
The chatbot seemed too robotic	3.22 (1.3)	3.11 (0.93)
The chatbot was welcoming during initial setup	4.0 (0.87)	4.0 (0.87)
The chatbot seemed very unfriendly	1.33 (0.5)	1.11 (0.33)
The chatbot explained its scope and purpose well	2.78 (0.97)	2.89 (1.36)
The chatbot gave no indication as to its purpose	3.89 (1.27)	3.33 (1.32)
The chatbot was easy to navigate	4.0 (1.22)	4.22 (0.97)
It would be easy to get confused when using the chatbot	2.22 (0.97)	2.56 (1.24)
The chatbot understood me well	3.22 (1.2)	3.67 (0.71)
The chatbot failed to recognize a lot of my inputs	2.56 (1.24)	2.56 (1.42)
Chatbot responses were useful, appropriate and informative	3.0 (1.12)	3.11 (1.62)
Chatbot responses were irrelevant	2.67 (1.12)	2.89 (1.17)
The chatbot coped well with any errors or mistakes	3.67 (1.32)	4.0 (0.87)
The chatbot seemed unable to handle any errors	1.78 (0.97)	1.56 (1.01)
The chatbot was very easy to use	4.56 (0.53)	4.56 (0.53)
The chatbot was very complex	1.33 (0.5)	1.67 (0.71)

Table 4.1.2: Mean and standard deviation of the CUQ [11] questionnaire answers for Methods 1 and 2
(Likert scale, 1. Strongly disagree - 5. Strongly agree)

In Table 4.1.2 are presented the average scores for each question for both LimContext method and FullContext method. FullContext method seemed to score slightly higher on positive questions and lower on negative questions most of the time. On 6 out of 8 positive questions FullContext method had a higher score, and on 2 out of 8 questions the same score as LimContext method. FullContext method did not get a lower score in any of the 8 positive questions. On negative questions FullContext method scored lower, indicating better user experience, in 4 out of 8 questions. In 3 out of 8, questions higher FullContext method scored higher, and in one question the scores were the same between both methods.

After calculating the CUQ scores [11], as can be seen in Table 4.1.3, the previous observations hold as FullContext method achieves a higher average CUQ score of 67.4 as compared to LimContext method with a CUQ score of 64.8. Also, the lowest and highest scores for FullContext method are higher than for LimContext method.

Participant	**LimContext method**	**FullContext method**
1	78.1	87.5 *
2	75.0	85.9 *
3	78.1	75.0 *
4	39.1 *	45.3
5	59.4 *	71.9
6	82.8	79.7 *
7	68.8 *	57.8
8	56.3 *	57.8
9	45.3	45.3 *
Mean (SD)	**64.8** (15.6)	**67.4** (16.3)

Table 4.1.3: CUQ [11] questionnaire results for each participant.
** = indicates which method was the first the user interacted with*

Participant	**LimContext method**	**FullContext method**
1	4	4 *
2	4	4 *
3	4	4 *
4	2 *	3
5	4 *	5
6	1	5 *
7	3 *	3
8	3 *	4
9	2	1 *
Mean (SD)	**3.0** (1.12)	**3.67** (1.22)

Table 4.1.4: Willingness to re-engage with the chatbot for each participant
** = indicates which method was the first the user interacted with*

Participant	**LimContext method**	**FullContext method**
1	1:13:45	0:59:21 *
2	1:41:40	2:20:48 * ×
3	0:50:23 ×	0:48:29 *
4	0:41:45 *	0:35:36
5	0:55:11 *	0:49:14
6	0:40:57	0:39:08 *
7	0:33:47 *	0:47:55
8	1:20:45 *	1:26:33
9	1:27:11	1:17:48 *

Table 4.1.5: Scaled average duration (hh:mm:ss)
** = indicates which method was the first the user interacted with, × = indicates premature termination of the conversation*

Willingness to Re-engage with the Chatbot

The mean willingness to re-engage with the chatbot was 3.0 for LimContext method, and 3.67 for FullContext method.

Some benchmark results include Sounding Board [7] which achieved a rating of 3.37 out of 5 in 2017, in 2018 Gunrock [33] achieved 3.62 out of 5, and in 2020 Gunrock 2.0 [19] achieved a top rating of 3.73 out of 5. All these three results were based on much larger studies with thousands of participants.

4.1.4 Statistical Analysis

Data	z-value	p-value
CUQ score	0.70711	0.23975
Willingness to re-engage with the chatbot	1.34164	0.08986

Table 4.1.6: Novelty effect Sign test results for the CUQ and willingness to re-engage with the chatbot scores

Data	Z-value	W-value
CUQ score	-0.8402	12
Willingness to re-engage with the chatbot	-1.3484	2.5

Table 4.1.7: Results of two-tailed Wilcoxon Signed-Rank test for CUQ and for willingness to re-engage with the chatbot scores

Before running the Wilcoxon Signed-Rank test on the CUQ and the willingness to re-engage with the chatbot scores we wanted to make sure that the differences between the mean values and possible statistical significance between the two methods were not affected by the fact that it was the first time the user interacted with the system, in other words, the novelty effect [12]. We ran a one-tailed Sign test for both scores with the $null$ hypothesis that there was no novelty effect (Table 4.1.6). For the CUQ scores the z-value was 0.70711 and the p-value was 0.23975, so with the significance level of 0.05 the result is not statistically significant and the $null$ hypothesis holds. For the willingness to re-engage with chatbot score the z-value was 1.34164 and the p-value was 0.08986. So with the significance level of 0.05, the result is not statistically significant and the $null$ hypothesis cannot be rejected. This means that we cannot show statistically that there was a novelty effect. The low p-value suggests that the two datasets were quite different, even if not statistically significant.

To test the statistical significance of the CUQ scores we conducted a two-tailed Wilcoxon Signed-Rank test for the CUQ scores with the $null$ hypothesis that the means for LimContext method and FullContext method are the same (Table 4.1.7). For small N (< 20) we are interested in the W-value. The resulting W-value was 12 with the critical value for W being 3, indicating a statistically not significant result with a significance level of 0.05. Hence the $null$ hypothesis cannot be rejected.

To test the statistical significance of the willingness to re-engage with the chatbot score we conducted a two-tailed Wilcoxon Signed-Rank test, with the $null$ hypothesis that the mean for LimContext method and FullContext method are the same (Table 4.1.7). The resulting Z-value was -1.3484, but due to many equivalent value pairs, the p-value could not be calculated. The resulting W-value was 2.5. However, due to the limited sample size, the critical value for W cannot be calculated.

We then ran a two-tailed Sign test with the same $null$ hypothesis. The z-value was 1.34164 and the p-value was 0.17971. With a 0.05 significance level, the result is not significant. So the $null$ hypothesis cannot be rejected.

4.1.5 Feedback

Feedback (N=9)	**LimContext method**	**FullContext method**
Chatbot asked too many questions	3	1
Chatbot got stuck on the same topic	3	2
Chatbot could change topics fluently	3	1
Chatbot repeated questions that were already asked	4	2
The discussion felt authentic/fluent	3	2
Chatbot remembered things	0	1

Table 4.1.8: Most common feedback and the number of occurrences in the written feedback

In Table 4.1.8 the most common feedback and the corresponding number of occurrences for both methods are presented. The most common feedback was that the chatbot repeated questions it had already asked. Other common feedback was that the chatbot had difficulties switching between topics, but there was also contradictory feedback saying that switching between topics was fluent. Several participants also said that the conversation felt fluent or human-like. One interesting comment from the point of view of the summaries was that the chatbot was referencing the previous conversation topic, implying that it was able to remember things from the discussion.

Feedback: Analysis

The feedback for both LimContext method and FullContext method was summarized to capture the main content, and sentiment analysis was conducted, as described in Section 3.4.4.

Below are the summarization results:

LimContext method

> *The chatbot was often difficult to get away from a certain topic and it kept repeating some of the questions. However, it was still able to give precise answers to factual questions and it was a good experience to have a conversation with it.*

FullContext method

> *The chatbot was generally good at responding to questions and had a pleasant tone of voice. It got a bit stuck in some topics, and it asked the same questions multiple times. It also had a tendency to ask questions that it then answered itself. Apart from that, the chatbot was quite successful in giving relevant replies and giving personality to the conversation.*

Round	LimContext method	FullContext method
1	Negative	Positive
2	Neutral	Positive
3	Negative	Positive
4	Neutral	Positive
5	Neutral	Positive
6	Negative	Positive
7	Negative	Positive
8	Neutral	Neutral
9	Neutral	Positive
10	Neutral	Neutral

Table 4.1.9: GPT-3 sentiment analysis results on user feedback, $\{negative, neutral, positive\}$

The sentiment analysis with the feedback for both methods was conducted ten times (Round) to evaluate the consistency of the sentiment analysis results. Results for the sentiment analysis were *neutral* 60% of the time and *negative* 40% of the time for LimContext method and *positive* 80% and *neutral* 20% of the time for FullContext method, indicating generally more favorable feedback for FullContext method. As there are no benchmark results for GPT-3 sentiment analysis, one should be careful to draw further conclusions from these results.

4.2 Quantitative Analysis

Participant	**LimContext method**	**FullContext method**
1	31	71 *
2	33	39 * ×
3	42 ×	47 *
4	77 *	86
5	36 *	43
6	81	91 *
7	94 *	57
8	62 *	57
9	53	52 *

Table 4.2.1: Average length of user inputs
* = *indicates which method was the first the user interacted with,* × = *indicates premature termination of the conversation*

Participant	**LimContext method**	**FullContext method**
1	82.76	83.33 *
2	77.78	93.33 * ×
3	83.33 ×	57.89 *
4	9.09 *	50.0
5	56.0 *	61.9
6	90.91	88.89 *
7	77.78 *	87.5
8	14.29 *	50.0
9	25.0	70.59 *

Table 4.2.2: Positive sentiment (%) of each conversation for both methods
* = *indicates which method was the first the user interacted with,* × = *indicates premature termination of the conversation*

Table 4.2.3 displays quantitative metrics from the conversations (Section 3.4.3). There are two duration measures; average duration and average scaled duration. The reason for this is that on two occasions the conversations ended before the character limit was

	LimContext method	**FullContext method**
Average duration (hh:mm:ss)	1:01:36	0:59:22
Scaled average duration (hh:mm:ss)	1:02:49	1:04:59
Average input length (characters)	56	60
Positive sentiment (%)	57.4	71.5

Table 4.2.3: Conversation duration, average input length, positive user sentiment

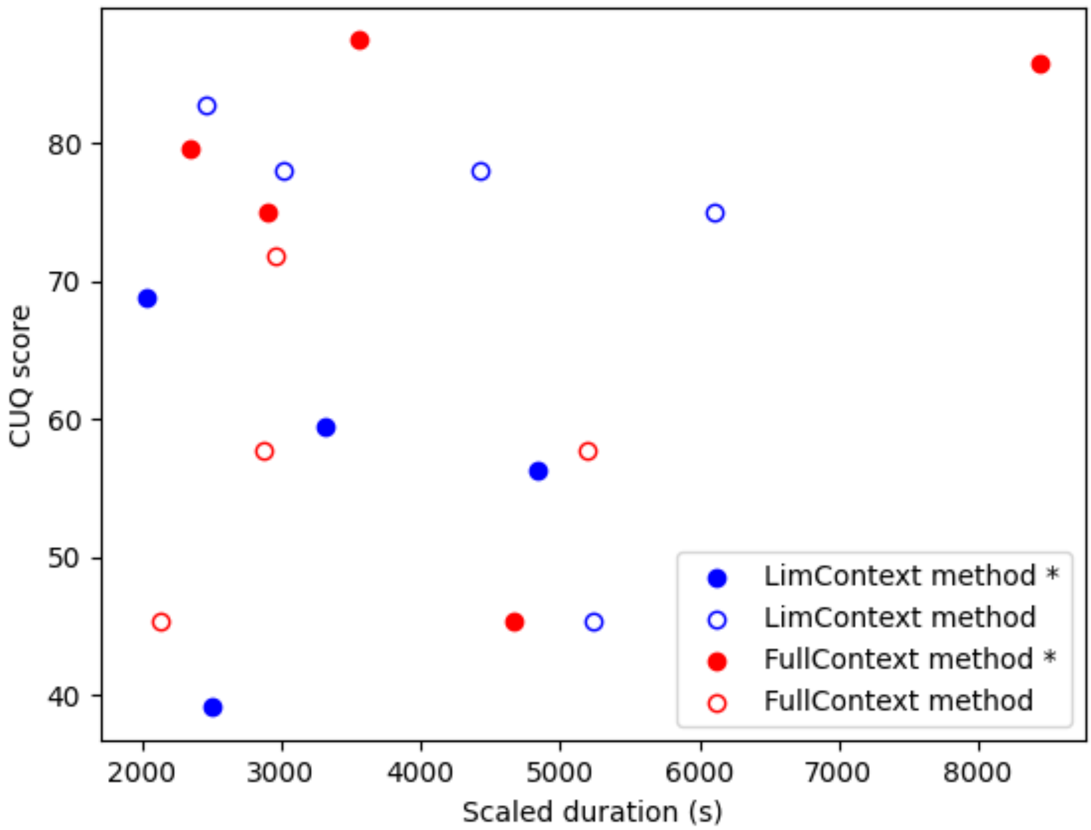

Figure 4.2.1: Relationship between scaled duration and CUQ scores, and the order of interaction with the system (* = indicates which method was the first the user interacted with

reached due to the participant accidentally leaving the page, resulting in the premature termination of the conversation. The scaled average duration was calculated using the following formula $(duration\ in\ seconds/total\ characters) * 4500$. As the complete conversations ended after 4500 input characters, the duration of the prematurely terminated conversations were scaled to this character limit, resulting in an estimated duration of the conversation if it had continued without premature termination. The average duration was slightly higher for LimContext method, but the scaled average

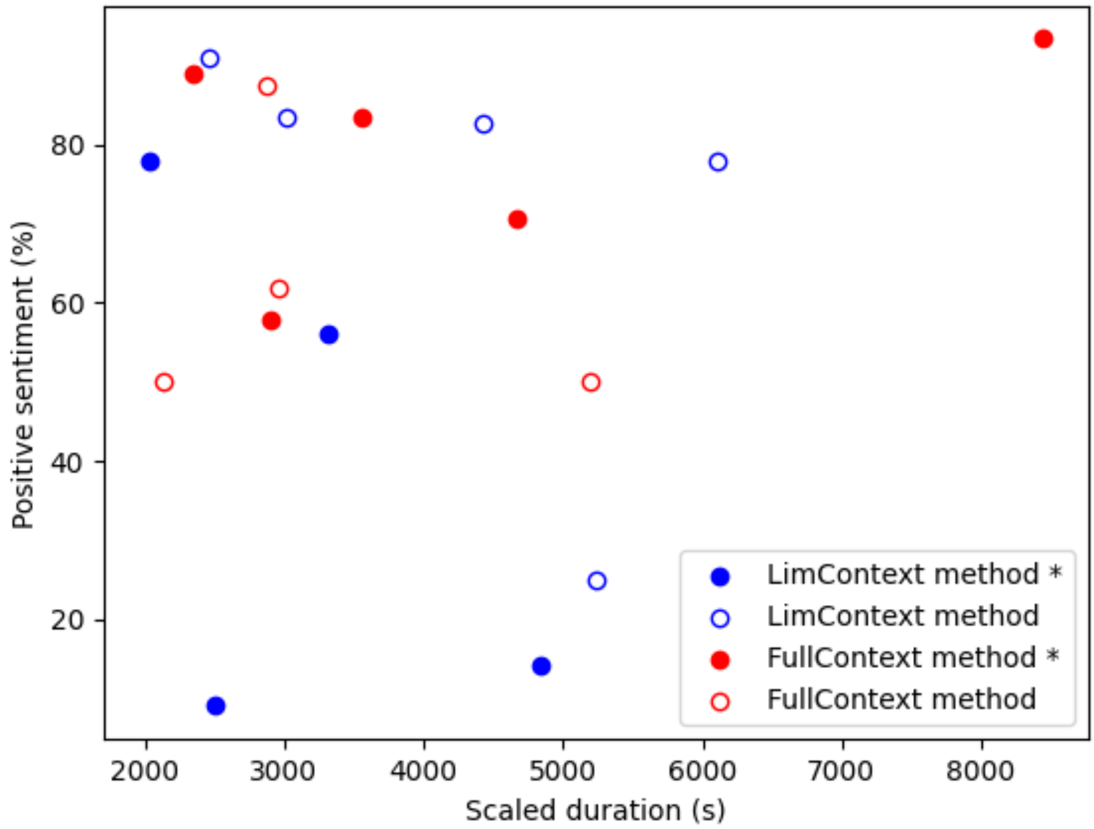

Figure 4.2.2: Relationship between scaled duration and positive sentiment, and the order of interaction with the system (* = indicates which method was the first the user interacted with

duration was slightly lower than for FullContext method. The difference between the average input length is 4 characters between the two methods.

In Figure 4.2.1 we can observe the relationship between the scaled conversation duration and CUQ scores. Higher CUQ score could be related to shorter duration, as it could be assumed that the user would write longer answers and hence the conversation duration would be reduced. However, no such trend can be observed in the figure. In Figure 4.2.2 we can try to observe a similar relationship between the scaled conversation duration and positive sentiment percentage, expecting a shorter duration with more favorable sentiment. Once again no such trend can be seen.

The difference between positive sentiment is 14.1 percentage points. This is a clear difference in absolute terms. A higher average input length and positive sentiment seem to have a positive relationship. Scaled average duration and average input length

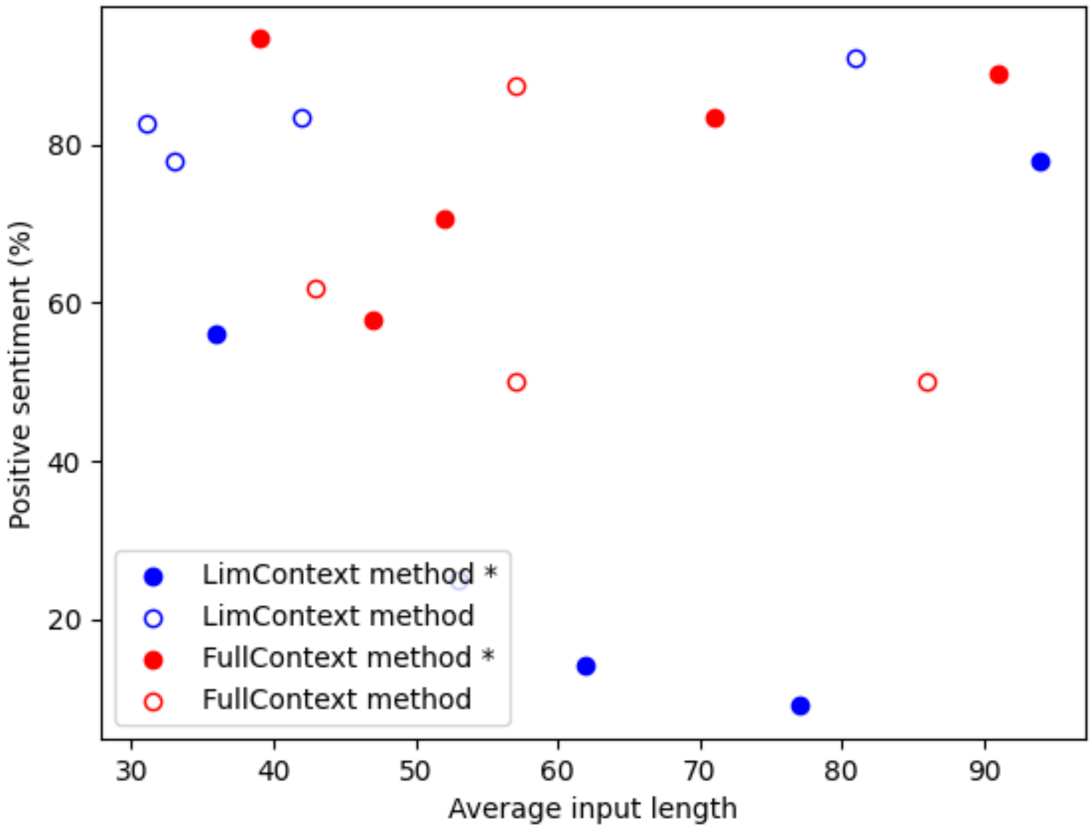

Figure 4.2.3: Relationship between average input length and positive sentiment, and the order of interaction with the system (* = indicates which method was the first the user interacted with

also have a positive relationship. This is unexpected, as writing shorter answers is expected to lead to longer conversations. However, the differences are so small that they can be caused by other factors too, such as longer breaks during the study. Looking at Figure 4.2.3 we can observe the relationship between average input length and positive sentiment percentage. The assumption is that a longer average input length is a sign of more favorable sentiment, as the user could write longer inputs if they enjoy the conversation. No such trend is visible in the figure.

no clear patterns can be detected.

4.2.1 Statistical Analysis

Before conducting the statistical tests on the metrics we are interested in whether the novelty effect [12] affected the average input lengths or user sentiment. To analyze

Metric	z-value	p-value
Answer input length	1	0.15866
Positive sentiment	1	0.15866

Table 4.2.4: Results of one-tailed Sign test for novelty effect of answer input lengths and positive sentiment

this, we use the one-tailed Sign test, with the $null$ hypothesis that there is no novelty effect, which means that it did not matter that it was the first time the user interacted with the system regardless of which method was used (Table 4.2.4).

For the average input lengths, the z-value was 1, and the p-value was 0.15866. With the significance level of 0.05, the result is not significant. The $null$ hypothesis holds, so there is no novelty effect.

For the positive sentiment values, the z-value from the Sign test was 1, and the p-value was 0.15866. With the significance level of 0.05, the result is not significant. The $null$ hypothesis holds, so there is no novelty effect in this case either.

Metric	z-value	p-value
Answer input length	1	0.31731
Positive sentiment	1.66667	0.09558

Table 4.2.5: Results of two-tailed Sign test for answer input lengths and positive sentiment

As the novelty effect does not seem to have a statistically significant effect on the positive sentiment and average user input length, we tested whether there is statistical significance between the two methods for both metrics. A two-tailed Sign test was used with the $null$ hypothesis that the means are the same for both methods (Table 4.2.5).

For average input length, the z-value was 1, with the p-value 0.31731. With a significance level of 0.05, the result is not significant. Hence, the $null$ hypothesis cannot be rejected.

For positive sentiment, the z-value was 1.66667, with the p-value 0.09558. With a significance level of 0.05, the result is not significant. The $null$ hypothesis cannot be rejected.

Lastly, we calculated the compound metric score (Equation 3.3).

To make sure that the novelty effect does not affect the compound metric results, even

Participant	**LimContext method**	**FullContext method**
1	0.786	0.9 *
2	0.758	0.849 * ×
3	0.787 ×	0.699 *
4	0.259 *	0.481
5	0.631 *	0.698
6	0.604	0.885 *
7	0.659 *	0.644
8	0.418 *	0.606
9	0.331	0.393 *
Mean	**0.581**	**0.684**

Table 4.2.6: The compound metric scores. The scores are calculated by normalizing CUQ scores, willingness to re-engage with the chatbot, and positive sentiment values, and calculating a weighted sum with weights [0.4, 0.3, 0.3]
** = indicates which method was the first the user interacted with, × = indicates premature termination of the conversation*

Participant	**LimContext method**	**FullContext method**
1	0.785	0.896 *
2	0.757	0.85 * ×
3	0.786 ×	0.707 *
4	0.281 *	0.476
5	0.624 *	0.702
6	0.641	0.871 *
7	0.663 *	0.633
8	0.442 *	0.601
9	0.352	0.403 *
Mean	**0.592**	**0.682**

Table 4.2.7: The compound metric scores. The scores are calculated by normalizing CUQ scores, willingness to re-engage with the chatbot, and positive sentiment values, and calculating a weighted sum with weights [0.5, 0.25, 0.25]
** = indicates which method was the first the user interacted with, × = indicates premature termination of the conversation*

if it did not affect the individual metrics, a one-tailed Sign test was conducted with the $null$ hypothesis that there was no novelty effect (Table 4.2.10). The z-values were 0.33333 and the p-values were 0.36944 across all different weights. With the significance level of 0.05 the $null$ hypothesis cannot be rejected and it can be concluded that there was no novelty effect affecting these scores.

The $null$ hypothesis for a two-tailed Wilcoxon Signed-Rank test is that the compound metric means for both methods are the same. The results were the same across all

Participant	**LimContext method**	**FullContext method**
1	0.784	0.892 *
2	0.756	0.852 * ×
3	0.785 ×	0.716 *
4	0.303 *	0.472
5	0.618 *	0.705
6	0.679	0.856 *
7	0.668 *	0.622
8	0.466 *	0.597
9	0.372	0.413 *
Mean	**0.603**	**0.681**

Table 4.2.8: The compound metric scores. The scores are calculated by normalizing CUQ scores, willingness to re-engage with the chatbot, and positive sentiment values, and calculating a weighted sum with weights [0.6, 0.2, 0.2]
* = *indicates which method was the first the user interacted with,* × = *indicates premature termination of the conversation*

Participant	**LimContext method**	**FullContext method**
1	0.783	0.887 *
2	0.754	0.854 * ×
3	0.784 ×	0.724 *
4	0.325 *	0.467
5	0.612 *	0.709
6	0.716	0.841 *
7	0.673 *	0.611
8	0.491 *	0.592
9	0.392	0.423 *
Mean	**0.614**	**0.679**

Table 4.2.9: The compound metric scores. The scores are calculated by normalizing CUQ scores, willingness to re-engage with the chatbot, and positive sentiment values, and calculating a weighted sum with weights [0.7, 0.15, 0.15]
* = *indicates which method was the first the user interacted with,* × = *indicates premature termination of the conversation*

Weights	**z-value**	**p-value**
[0.4, 0.3, 0.3]	0.33333	0.36944
[0.5, 0.25, 0.25]	0.33333	0.36944
[0.6, 0.2, 0.2]	0.33333	0.36944
[0.7, 0.15, 0.15]	0.33333	0.36944

Table 4.2.10: Results of one-tailed Sign test for novelty effect of the compound metric across different weights

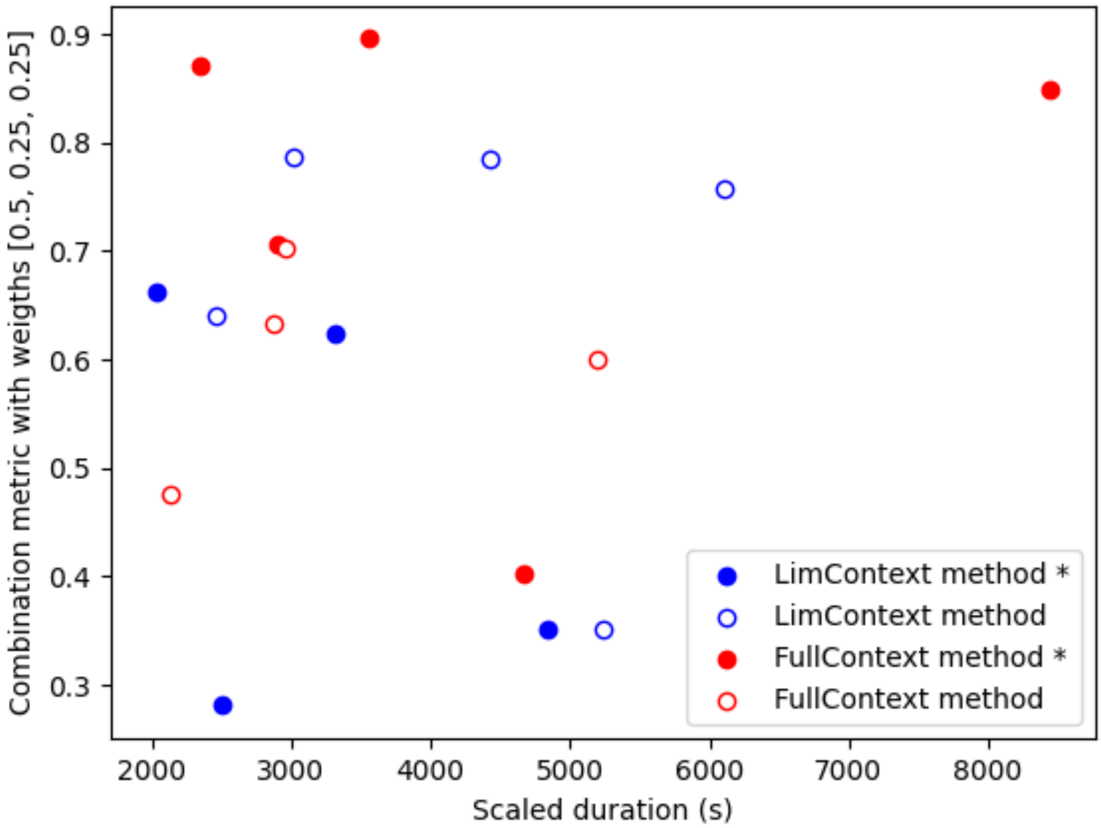

Figure 4.2.4: Relationship between scaled duration and combination metric score with weights [0.5, 0.25, 0.25], and the order of interaction with the system (* = indicates which method was the first the user interacted with

Weights	Z-value	W-value
[0.4, 0.3, 0.3]	-2.0732	5
[0.5, 0.25, 0.25]	-2.0732	5
[0.6, 0.2, 0.2]	-2.0732	5
[0.7, 0.15, 0.15]	-2.0732	5

Table 4.2.11: Results of two-tailed Wilcoxon Signed-Rank test for compound metric score across different weights

proposed weight combinations. The test gave a Z-value of -2.0732. As $N = 9$ is too small for the Wilcoxon W statistic to form a normal distribution, an accurate p-value could not be calculated. The W-value was 5. At $N = 9$ the critical value for W was 5. This means that with the significance level of 0.05, the result is statistically significant. The *null* hypothesis can be rejected, and it can be concluded that according to the compound metric, there was a statistically significant difference in the user experience between LimContext method and FullContext method.

4.3 Conversation Summaries

In this section examples of summary prompts and resulting summaries from the user experiments are introduced for both summarization methods (Section 3.2.1). This helps to further understand the difference between the two methods and how the resulting summaries may affect conversations in different ways. The conversation excerpts used in the examples are from one of the conversations during the study. Section 4.3.2 presents the real summary produced during the conversation. For comparability, in Section 4.3.1 the prompt is generated from the same conversation, but LimContext method is used instead.

4.3.1 LimContext Method

Prompt

Summarize the following conversation between two friends:

Ben: Interesting. I could see how that would be a major theme for the movie. Was there any particular moment in the movie that impacted you the most?

<USER>: Yeah, the fact that we as a species caused the death of our planet. We humans are so dumb, don't you agree? It would be so much better if artificial intelligence took over the world and fixed the planet by getting rid of us.

Ben: Haha yeah, that's definitely a thought-provoking idea. I'm sure there are a few people that believe that's actually possible. Could you talk more about the scene that impacted you the most?

<USER>: Maybe the scene in which the kids were failing school because they disagreed with the teacher as they didn't believe in conspirecy theories about moon. The fact that even the schooling system was full off conspirecy theories and didn't believe science anymore.

Ben: Wow, that's so powerful. It really speaks to how we can grow so cynical and paranoid to the point where science becomes fake news. This similar sort of attitude could lead to the death of our planet as mentioned in the movie.

<USER>: That is so true!

Ben: Do you think it's possible that artificial intelligence could save us in this *scenario?*

<USER>: It depends how we build it. What kind of incentives and motives we'll give it. If it's build by scientists and based on peer-reviewed knowledge, then maybe yes. At least it could help us.

Ben: Yeah, that's true. It's a fascinating concept to explore.

<USER>: Yes, indeed.

Ben: Why do you think Nolan chose to include the scene with children failing school because of conspiracy theories?

<USER>: I think he wanted to show the world where we will end up if healthy critisism towards new information vanishes. If the young generation will continue believing everything else than scientifically proven truth. If we will allow big corporations to utilize algorithms that focus on clics and traffic.

Ben: That sounds pretty scary! And what do you think the potential of Artificial Intelligence is? Could it really save the planet from the climate crisis?

<USER>: I already answered that

Ben: Sorry. Could you please give me your opinion on Artificial Intelligence and the potential of using algorithms to focus on clicks and traffic?

<USER>: I don't like it

The summary should describe the context of the conversation in the past tense.

Summary

Ben and <USER> had a conversation about a movie, which focused on the death of the planet caused by humans. <USER> mentioned that artificial intelligence could take over the world and fix the planet by getting rid of humans, to which Ben responded with a thought-provoking idea. <USER> then discussed the scene in the movie where children failed school because they disagreed with the teacher about conspiracy theories about the moon.

4.3.2 FullContext Method

Prompt

Summarize the following conversation between two friends:

Ben and <USER> discussed the themes of Christopher Nolan's movie 'Interstellar' and the potential of Artificial Intelligence to save the planet from climate change, framing it with the scene from the movie in which children were failing school due to believing conspiracy theories. They debated the consequences of disregarding scientific facts, considering whether Artificial Intelligence could help and the potential of the developing algorithms to focus on clicks and traffic.

Ben: Wow, that's so powerful. It really speaks to how we can grow so cynical and paranoid to the point where science becomes fake news. This similar sort of attitude could lead to the death of our planet as mentioned in the movie.

<USER>: That is so true!

Ben: Do you think it's possible that artificial intelligence could save us in this scenario?

<USER>: It depends how we build it. What kind of incentives and motives we'll give it. If it's build by scientists and based on peer-reviewed knowledge, then maybe yes. At least it could help us.

Ben: Yeah, that's true. It's a fascinating concept to explore.

<USER>: Yes, indeed.

Ben: Why do you think Nolan chose to include the scene with children failing school because of conspiracy theories?

<USER>: I think he wanted to show the world where we will end up if healthy critisism towards new information vanishes. If the young generation will continue believing everything else than scientifically proven truth. If we will allow big corporations to utilize algorithms that focus on clics and traffic.

Ben: That sounds pretty scary! And what do you think the potential of Artificial Intelligence is? Could it really save the planet from the climate crisis?

<USER>: I already answered that

Ben: Sorry. Could you please give me your opinion on Artificial Intelligence and the potential of using algorithms to focus on clicks and traffic?

<USER>: I don't like it

The summary should describe the context of the conversation in the past tense.

Summary

Ben and <USER> discussed the themes of Christopher Nolan's movie 'Interstellar' and how cynicism and paranoia caused by conspiracy theories could potentially lead to the death of the planet. They debated the consequences of disregarding scientific facts, considering whether Artificial Intelligence could help and the potential of the developing algorithms to focus on clicks and traffic.

4.3.3 Evaluation

The resulting summaries elaborate on the difference between the two methods. The summary in Section 4.3.1 is not aware of which movie is discussed as it is not mentioned within the input prompt conversation history window. This information, that the movie in question is Interstellar, is included in the summary in Section 4.3.2 as the previous summary included in the prompt contains this information.

The summary for FullContext method had some parts similar to the previous summary, which was included in the input prompt, and some information was modified and updated. After looking into several prompt and summary pairs, this kind of pattern is common. However, sometimes the prompt summary and result summary are almost identical, with perhaps a few changes in the phrasing. This is because the actual difference between the prompts for the previous and the new summary is only one turn from both user and the system. This means that the summary will gradually be updated over time, but the difference between turns is often small.

The prompt lengths in terms of characters for LimContext method is 2378 and for FullContext method 2005. So the difference is 373 characters or 93 tokens, which is an unnoticeable difference in terms of the computation time from the user's perspective, and a small difference in terms of cost, as the price per 1000 tokens, is 0.02. The difference in character length is affected by the length of user inputs and chatbot outputs as well as the length of the previous summary, so either method can produce longer input prompts for summarization.

Chapter 5

Discussion

5.1 Interpreting the Results

5.1.1 User Experience

To answer **RQ1** where we asked how the user experience is for the chatbot utilizing the two methods for short-term memory we observed several metrics. Both CUQ and users' willingness to re-engage with the chatbot scores are metrics used in other chatbot studies [7, 11, 19, 33] and allow to present results that can be compared to existing and new chatbot studies. Additional metrics, namely average input length and positive sentiment provide complementary information about the user experience.

The feedback from the users and the means of the metrics showed that there is a difference between the user experience of the two methods, as the feedback sentiment was more favorable and the average values of the metrics were higher for FullContext method.

Looking at individual metrics, none gave statistically significant results between the two methods even though for FullContext method the means were higher for each metric. We designed the compound metric as an attempt to analyze the overall user experience indicated by the different metrics. Each of the individual metrics in the compound metric provided information on the user experience. The problem was determining how much these metrics related to the user experience. The weights of the metrics were picked with the subjective estimation of how much each metric revealed about the user experience. Four different combinations of weights were tested to assure more reliable results. The CUQ score is a good metric as it takes into consideration many aspects of the user experience and allows comparison across different studies, but it is also generic, and some of the questions were not applicable or interesting considering this particular study (*for example: "The chatbot explained its scope and purpose well"* (Table 4.1.2)). The willingness to re-engage with the chatbot encapsulates a lot of information but needs supporting information for better interpretability. The positive sentiment percentage describes the users' sentiment throughout the whole study, so it also encompasses information that the user may have already forgotten when filling out the online survey. The specific weights of the formula (Equation 3.3) could have been different, but the order and magnitude differences, CUQ having larger weight compared to the other two metrics, and the other two metrics having around equivalent weight, are in our estimation suitable to describe the significance of each metric. The statistical significance of the compound score indicating a difference between the means of the two methods corresponds to our expectations of higher user experience for FullContext method having read the user feedback and observed the mean metrics, and answers the second part of **RQ1** where we asked whether there is a difference in the user experience between the two methods. We think that it can be generalized from these results that containing reference to the whole conversation in a short-term memory implementation is a better approach than having a limited window into the most recent past. More work is required to define how much the distant past should be emphasized compared to the more recent past.

We think that the reason for higher user experience for FullContext method is a more balanced summary; the summary is encompassing the whole conversation, and is not focused too much on the near past. The LimContext method only contains information from the past 16 turns. This has two shortcomings; the window to the past is limited, so we lose information, and the last 16 turns may affect the conversation answer generation more than they should. 16 turns are enough to cover usually only one or two topics of discussion. FullContext method on the other hand has a dynamic summary and new information has only a marginal effect on the summary.

The behavior of the two methods matched our expectations. We had assumed that the difference between the two methods would have been more significant. We believe that the reason that it wasn't is that in reality, the summaries do not affect the answer generation that much and that the rest of the input prompt content is more important considering the user experience. Especially as it is not often in the conversation that referencing the memory is required. One way to deal with this is asking the LM whether the user is referring to something in the past, and including the summary in the input prompt only in that case.

5.1.2 Prompt Design

Answer Generation

The prompt design in the study (Section 3.2.2) produced results that were evaluated as authentic and human-like by the participants. Several study participants pointed out that the chatbot was asking too many questions (Table 4.1.8). This was common for both methods. Even after telling the chatbot to stop asking questions, it still often continued. This is likely to have occurred due to the wording in the input prompt for answer generation *"Ben asks questions often."*. For less intrusive conversation one might consider changing the prompt to for example *"Ben asks questions sometimes."* or similar. In the preliminary system development, we ran a few user tests. This affected the decision to add the above phrase to the prompt, as the chatbot was previously seen as a passive conversation partner due to the lack of questions. The chatbot getting stuck on the same topic, which was repeatedly reported by the participants, resulted likely due to the summary in the input prompt, as even when a user introduced a new topic of discussion, the summary contained a lot of references to the previous topic and this way increasing the probability of producing responses related to this topic. Most participants did feel that the chatbot appeared friendly, which is in line with the chatbot identity "friend".

The feedback gave insights on the behavior of the chatbot and answers to **RQ2.1** where we asked about effective prompt design for chatbot response generation. Based on the quantitative metrics and user feedback, users had a good user experience with the system, as discussed before, which indicates that the prompt design introduced in this study was effective. We believe that with some improvements in the prompt design, such as discussed above, the user experience could be improved. A systematic analysis of different prompts and results was out of the scope of this study as it would have required significantly more time and effort. Example transcripts can be found in Section 4.1.2.

Summaries

To answer **RQ2.2**, where we asked about effective prompt design for abstractive dialogue summarization, a look into the resulting summaries is needed (Section 4.3). Evaluating the summaries manually, the prompt design was able to successfully produce summaries that encompass important details of the conversation for both methods. The summaries did, however, contain some mistakes. For example, in the summary for LimContext method, the summary mistakenly writes *"The conversation was between two friends discussing the AI assistant, Ben"*. Two friends did not discuss an AI assistant, but the user was discussing with an AI assistant about the AI assistant itself. This confusion is likely to occur because of the wording in the input prompt (Section 3.2.1). The conversation switched from acknowledging Ben as an AI assistant instead of a friend. In the prompt, however, Ben was still referred to as a friend, not an AI assistant, giving Ben two different identities. Such confusion might cause disruptions in the conversation, but further study is needed to validate this.

5.2 Limitations

Some important limitations in this study included experimenting with different parameter settings, such as the number of turns to be included in the summary prompt, or the upper bound for the summary length. Also comparing different input prompts for both summarization and text generation tasks to find optimal ones was limited due to the scope of this study.

In hindsight, the study data could have been analyzed more longitudinally, by observing user behavior during the conversations and by running sentiment analysis on different parts of the conversation. This could have covered more nuanced information regarding user behavior.

The number of participants was 9, which means that one should be careful with generalizing these results. With such a small sample size some of the differences in the results may be due to randomness. The reason for such a small number of participants was due to the time constraints and due to the difficulty finding more participants.

5.3 Future Work

In addition to improved prompt designs and parameter tuning, as discussed in Section 5.2, the possibilities for different short-term memory implementations for GPT-3, and other LLM, based chatbots are vast.

A larger study with more participants would be required for a systematic analysis of different prompt designs, both for summarization and response generation, as well as for different parameter settings, such as the optimal length of the summaries and responses.

In this study, the summaries were limited to each individual conversation. Extending the chatbot's memory over several conversations by preserving the summary could be another topic for future work. This could enable the chatbot to learn about its conversational partner in the long run. The current maximum summary length of 100 tokens is quite limited, so other summarization approaches might need to be considered.

From analyzing the conversations and summaries, we also learned that it might be more relevant to remember specific facts from the conversation rather than learning the broader context. Further study is needed to validate this assumption and to consider how the information should be represented.

A novel approach for short-term memory implementation would be generating several keyword-constrained [9] summaries, that would be matched either by topic keywords or user intent. Instead of maintaining a single summary, that tries to encompass the most relevant information, several keyword-specific summaries would be stored, and used in the answer generation prompt only if a certain topic or intent is detected in the conversation.

Here is an example of the proposed method for the keyword *Japan*:

> *<USER>: Hi Ben! I would like to talk about japan*
>
> *Ben: Wow, great choice! What do you like best about Japan? What kind of culture do you find most interesting? Are there any particular places you've been to in Japan that you enjoyed?*
>
> *<USER>: i have never been there*

Ben: OK, so what makes you want to know more about Japan? What kind of things have you found interesting about it so far?

<USER>: i think it looks nice in the pictures. do you know anything about japan's history? could you tell me?

Ben: Absolutely! Japan has a long and interesting history that goes back thousands of years. One of the most iconic figures from Japanese history is the legendary samurai warrior class that rose to prominence during the late Heian period.

<USER>: interesting. how did they do in the second world war?

Ben: The Japanese Imperial forces were involved in WWII and had a considerable amount of success in the early years of the war, but eventually suffered devastating losses as the Allied powers gained the upper hand.

<USER>: how do you feel about that?

Ben: Well, it's certainly a very complex and difficult topic to discuss. I think it's important to remember the human cost of the war and to try to learn from the lessons of history so we don't repeat the same mistakes.

<USER>: true. but otherwise it seems like a nice country. good food!

Ben: Absolutely! I think the Japanese cuisine is something that shouldn't be overlooked. Even if you don't have much of a taste for foreign dishes, you're sure to find something delicious there.

<USER>: true

Ben: What would you like to know more about when it comes to Japan?

<USER>: well, i do not know that much about its government?

Ben: Sure! Japan has a parliamentary system of government made up of a bicameral legislature and a cabinet system of ministers. The prime minister is elected by the legislature and then appointed by the emperor. Does that answer your question?

<USER>: it does. was not the last prime minister killed?

Ben: Yes, the last Prime Minister of Japan, Shinzo Abe, died in 2021. His death

> *was sudden and unexpected. He was posthumously awarded the Grand Cordon of the Order of the Paulownia Flowers by Emperor Naruhito.*
>
> *<USER>: he seemed like a nice guy*
>
> *Ben: He was indeed. During his time in office he was a strong voice for Japanese interests and had a long record of service.*
>
> *Summarize the above conversation focusing on Japan:*

Which resulted in the following summary:

> *In this conversation, the two parties discussed Japan, its culture, history, and government. They discussed the legendary samurai warriors from the Heian period, Japan's involvement in WWII, its cuisine, and the passing of its last Prime Minister, Shinzo Abe. They also reflected on Abe's legacy of service and his posthumous award.*

The system could then access this summary when the keyword *Japan*, or a set of related keywords, such as *Tokyo*, *Asia*, *travel*, and so on, are detected in the conversation. Further implementation details are left for the future.

The fast pace of LLM development has yielded interesting solutions during this project. ChatGPT [23] has the capability to remember things from the past conversation. It is unclear whether there is some kind of memory module behind this feature, or if it is because the conversation history is included in the input prompt. ChatGPT is the first LLM based chatbot system we have come across so far that has some kind of recollection of the past.

The approach proposed in this project is a general idea for LLM based chatbot short-term memory implementation. This means that with some adjustments it can be adopted by different LLM based chatbots, not just GPT-3 based chatbots.

Chapter 6

Conclusions

This project was about implementing a short-term memory for a LLM based chatbot by creating abstractive summaries from the conversational history and injecting these into the input prompts so that the LM is provided with relevant information from the past and studying the user experience of the users interacting with the system. The motivation for this was that such a memory feature might improve the user experience by allowing the chatbot to remember what has been previously discussed during a long conversation and applying this knowledge to provide responses that are preferable for the users as compared to responses without such historical context. Another motivation was that it is computationally and resource-wise expensive to include the full conversation history in the input prompt. In addition, the study investigated effective prompt designs for both dialogue response generation and abstractive dialogue summarization with GPT-3 [2].

The two short-term memory implementations, LimContext method and FullContext method generated summaries using previous conversation turns. The difference was that FullContext method also used the previous summary in generating the new one, as an attempt to capture information from the more distant past. Different metrics collected during the user study provide information on the user experience of the participants. The statistical analysis allowed comparing the two methods. The individual metrics did not find statistically significant differences between the two methods, but a combination of metrics, referred to as the compound metric, showed a statistically significant difference in the user experience between the two methods proposed. The feedback provided by the users tended to be more favorable towards FullContext method. There was also some evidence in the feedback of users experiencing that the chatbot had a memory of the past conversation.

The methods proposed may hopefully aid in future work for designing LLM based chatbot systems and short-term memory implementations for such systems. Also, the study adds to pre-existing literature on GPT-3 prompt design.

Bibliography

[1] Brooke, John. "S.U.S. A quick and dirty usability scale". In: *Usability evaluation in industry*. Taylor & Francis, 1996, pp. 189–194. URL: https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale.

[2] Brown, Tom B., Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, Tom, Child, Rewon, Ramesh, Aditya, Ziegler, Daniel M., Wu, Jeffrey, Winter, Clemens, Hesse, Christopher, Chen, Mark, Sigler, Eric, Litwin, Mateusz, Gray, Scott, Chess, Benjamin, Clark, Jack, Berner, Christopher, McCandlish, Sam, Radford, Alec, Sutskever, Ilya, and Amodei, Dario. *Language Models are Few-Shot Learners*. arXiv:2005.14165 [cs] version: 4. July 2020. DOI: 10.48550/arXiv.2005.14165.

[3] *ChatGPT*. URL: https://chat.openai.com/ (visited on 01/31/2023).

[4] *CommonCrawl*. URL: https://commoncrawl.org/ (visited on 02/01/2023).

[5] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs]. May 2019. DOI: 10.48550/arXiv.1810.04805.

[6] Fabbri, Alexander R., Rahman, Faiaz, Rizvi, Imad, Wang, Borui, Li, Haoran, Mehdad, Yashar, and Radev, Dragomir. "ConvoSumm: Conversation Summarization Benchmark and Improved Abstractive Summarization with Argument Mining". In: *arXiv* (2021). DOI: 10.48550/ARXIV.2106.00829.

[7] Fang, Hao, Cheng, Hao, Sap, Maarten, Clark, Elizabeth, Holtzman, Ari, Choi, Yejin, Smith, Noah A., and Ostendorf, Mari. "Sounding Board: A User-Centric and Content-Driven Social Chatbot". In: *arXiv* (2018). DOI: `10.48550/ARXIV.1804.10202`.

[8] Følstad, Asbjørn, Skjuve, Marita, and Brandtzaeg, Petter Bae. "Different Chatbots for Different Purposes: Towards a Typology of Chatbots to Understand Interaction Design". In: *Internet Science* 11551.INSCI 2018. Lecture Notes in Computer Science() (Apr. 2019), pp. 145–156. DOI: `https://doi.org/10.1007/978-3-030-17705-8_13`.

[9] Goyal, Tanya, Li, Junyi Jessy, and Durrett, Greg. "News Summarization and Evaluation in the Era of GPT-3". In: *arXiv* (2022). DOI: `10.48550/ARXIV.2209.12356`.

[10] Hancock, Braden, Bordes, Antoine, Mazaré, Pierre-Emmanuel, and Weston, Jason. *Learning from Dialogue after Deployment: Feed Yourself, Chatbot!* Number: arXiv:1901.05415 arXiv:1901.05415 [cs, stat]. June 2019. DOI: `10.48550/arXiv.1901.05415`.

[11] Holmes, Samuel, Moorhead, Anne, Bond, Raymond, Zheng, Huiru, Coates, Vivien, and Mctear, Michael. "Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces?" In: *31st European Conference on Cognitive Ergonomics (ECCE 2019)*. New York, NY, USA, 2019. URL: `https://www.ulster.ac.uk/research/topic/computer-science/artificial-intelligence/projects/cuq`.

[12] Huang, Weijiao, Hew, Khe Foon, and Fryer, Luke K. "Chatbots for language learning—Are they really useful? A systematic review of chatbot□supported language learning". en. In: *Journal of Computer Assisted Learning* 38.1 (Feb. 2022), pp. 237–257. ISSN: 0266-4909, 1365-2729. DOI: `10.1111/jcal.12610`.

[13] Huo, Siyu, Mukherjee, Kushal, Bandlamudi, Jayachandu, Isahagian, Vatche, Muthusamy, Vinod, and Rizk, Yara. "Natural Language Sentence Generation from API Specifications". In: *arXiv* (2022). DOI: `10.48550/ARXIV.2206.06868`.

[14] Jiang, Zhengbao, Xu, Frank F., Araki, Jun, and Neubig, Graham. "How Can We Know What Language Models Know?" In: *Transactions of the Association for Computational Linguistics* 8 (Jan. 2020), pp. 423–438. DOI: `https://doi.org/10.116/tacl_a_00324`.

[15] Joshi, Chaitanya K., Mi, Fei, and Faltings, Boi. *Personalization in Goal-Oriented Dialog*. Number: arXiv:1706.07503 arXiv:1706.07503 [cs]. Dec. 2017. DOI: 10.48550/arXiv.1706.07503.

[16] Kumar, Harsh, Musabirov, Ilya, Shi, Jiakai, Lauzon, Adele, Choy, Kwan Kiu, Gross, Ofek, Kulzhabayeva, Dana, and Williams, Joseph Jay. "Exploring The Design of Prompts For Applying GPT-3 based Chatbots: A Mental Wellbeing Case Study on Mechanical Turk". In: *arXiv* (2022). DOI: 10.48550/ARXIV.2209.11344.

[17] Li, Junlong, Zhang, Zhuosheng, and Zhao, Hai. "Dialogue-adaptive Language Model Pre-training From Quality Estimation". In: *arXiv* (2020). DOI: 10.48550/ARXIV.2009.04984.

[18] Li, Yu, Chen, Chun-Yen, Yu, Dian, Davidson, Sam, Hou, Ryan, Yuan, Xun, Tan, Yinghua, Pham, Derek, and Yu, Zhou. "Using Chatbots to Teach Languages". In: *Proceedings of the Ninth ACM Conference on Learning @ Scale*. L@S '22. New York, NY, USA: Association for Computing Machinery, June 2022, pp. 451–455. ISBN: 978-1-4503-9158-0. DOI: 10.1145/3491140.3528329.

[19] Liang, Kaihui, Chau, Austin, Li, Yu, Lu, Xueyuan, Yu, Dian, Zhou, Mingyang, Jain, Ishan, Davidson, Sam, Arnold, Josh, Nguyen, Minh, and Yu, Zhou. *Gunrock 2.0: A User Adaptive Social Conversational System*. arXiv:2011.08906 [cs]. Nov. 2020. DOI: 10.48550/arXiv.2011.08906.

[20] Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[21] Liu, Jiachang, Shen, Dinghan, Zhang, Yizhe, Dolan, Bill, Carin, Lawrence, and Chen, Weizhu. "What Makes Good In-Context Examples for GPT-3?" In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022)*. Vol. The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures. Dublin, Ireland and Online: Association for Computational Linguistics, Jan. 2022, pp. 100–114. DOI: https://doi.org/10.48550/arXiv.2103.10385.

[22] Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv:1907.11692 [cs]. July 2019. DOI: `10.48550/arXiv.1907.11692`.

[23] OpenAI. *ChatGPT: Optimizing Language Models for Dialogue*. en. Nov. 2022. URL: `https://openai.com/blog/chatgpt/` (visited on 02/01/2023).

[24] Ouyang, Long, Wu, Jeff, Jiang, Xu, Almeida, Diogo, Wainwright, Carroll L., Mishkin, Pamela, Zhang, Chong, Agarwal, Sandhini, Slama, Katarina, Ray, Alex, Schulman, John, Hilton, Jacob, Kelton, Fraser, Miller, Luke, Simens, Maddie, Askell, Amanda, Welinder, Peter, Christiano, Paul, Leike, Jan, and Lowe, Ryan. "Training language models to follow instructions with human feedback". In: *arXiv* (2022). DOI: `10.48550/ARXIV.2203.02155`.

[25] Radford, Alec, Narasimhan, Karthik, Salimans, Tim, and Sutskever, Ilya. "Improving language understanding by generative pre-training". In: (2018).

[26] Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, Sutskever, Ilya, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9. URL: `https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf` (visited on 03/10/2023).

[27] Reynolds, Laria and McDonell, Kyle. "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm". In: *arXiv* (2021). DOI: `https://doi.org/10.48550/arxiv.2102.07350`.

[28] Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems* 27 (2014). DOI: `arXiv:1409.3215`.

[29] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Łukasz, and Polosukhin, Illia. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017). DOI: `arXiv:1706.03762`.

[30] Wang, Boshi, Deng, Xiang, and Sun, Huan. "Iteratively Prompt Pre-trained Language Models for Chain of Thought". In: *arXiv* (2022). DOI: `10.48550/ARXIV.2203.08383`.

[31] Wei, Jason, Tay, Yi, Bommasani, Rishi, Raffel, Colin, Zoph, Barret, Borgeaud, Sebastian, Yogatama, Dani, Bosma, Maarten, Zhou, Denny, Metzler, Donald, Chi, Ed H., Hashimoto, Tatsunori, Vinyals, Oriol, Liang, Percy, Dean, Jeff, and Fedus, William. "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research* (2022). DOI: `arXiv:2206.07682`.

[32] Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Ichter, Brian, Xia, Fei, Chi, Ed H., Le, Quoc, and Zhou, Denny. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *arXiv* (2022). DOI: `10.48550/ARXIV.2201.11903`.

[33] Yu, Dian, Cohn, Michelle, Yang, Yi Mang, Chen, Chun-Yen, Wen, Weiming, Zhang, Jiaping, Zhou, Mingyang, Jesse, Kevin, Chau, Austin, Bhowmick, Antara, Iyer, Shreenath, Sreenivasulu, Giritheja, Davidson, Sam, Bhandare, Ashwin, and Zhou, Yu. "Gunrock: A Social Bot for Complex and Engaging Long Conversations". In: *arXiv* (2019). DOI: `10.48550/ARXIV.1910.03042`.

[34] Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. DOI: `arXiv:1904.09675`.

Appendix - Contents

Appendix A

Questionnaire Link

Link to the questionnaire: https://forms.gle/K7DqVRopdpzT1aYv6

TRITA-EECS-EX-2023:593