BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

SIMULATION OF HUMAN INTERACTION USING AI

SIMULACE LIDSKÉ INTERAKCE S VYUŽITÍM AI

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR Bc. DAVID CHOCHOLATÝ
AUTOR PRÁCE

SUPERVISOR Ing. RADEK HRANICKÝ, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2025

Master's Thesis Assignment

161849

Institut: Department of Information Systems (DIFS)
Student: **Chocholatý David, Bc.**
Programme: Information Technology and Artificial Intelligence
Specialization: Machine Learning
Title: **Simulation of Human Interaction using AI**
Category: Artificial Intelligence
Academic year: 2024/25

Assignment:

1. Learn about artificial intelligence with a focus on Large Language Models (LLM).
2. Learn techniques for human interaction with LLM.
3. Design an experimental environment in which several LLM-based entities interact with each other to simulate human behavior.
4. Implement the proposed experimental environment.
5. Conduct a set of experiments that focus on different model situations, including the performance of predefined tasks.
6. Analyse the findings and evaluate the success of the predefined tasks.
7. Discuss the results obtained.

Literature:

- Abbasiantaeb, Zahra, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. "Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions." In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 8-17. 2024.
- Gao, Jie, Simret Araya Gebreegziabher, Kenny Tsu Wei Choo, Toby Jia-Jun Li, Simon Tangi Perrault, and Thomas W. Malone. "A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration." In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1-11. 2024.
- Kim, Callie Y., Christine P. Lee, and Bilge Mutlu. "Understanding large-language model (llm)-powered human-robot interaction." In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 371-380. 2024.

Requirements for the semestral defence:
Points 1 to 3.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Hranický Radek, Ing., Ph.D.**
Consultant: Patzelt Petr, Ing.
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: 1.11.2024
Submission deadline: 21.5.2025
Approval date: 22.10.2024

Abstract

Simulation of the believable human interaction can strengthen the application of the large language models (LLMs) in computational social sciences. Especially for experiments in interpersonal communication using modeling tools. In this work, a *PerSimChat* framework is designed that provides an experimental environment for simulating multi-human conversations using LLM agents with real-world persona data. Simultaneously, a new approach for selecting the order of the agent's speech called *One-by-One Talk with Agent's Need to Talk* is introduced. Empirical studies demonstrate the framework's performance on many evaluation scenarios, beating the baseline solutions on the dimensions of believability, credibility, content depth, and relevance. The system achieves competitive results with other multi-agent debate systems on commonsense and mathematical benchmarks. In addition, this thesis provides a demonstration web application for creating simulations and running variety of scenarios.

Abstrakt

Simulace věrohodné interakce mezi lidmi může posílit využití velkých jazykových modelů (LLM) v oblasti výpočetních sociálních věd. Zejména pro experimenty v mezilidské komunikaci s využitím modelovacích nástrojů. V této práci je navržen nástroj *PerSimChat*, který poskytuje experimentální prostředí pro simulaci konverzací více osob pomocí LLM agentů s daty o reálných lidech. Současně je představen nový přístup pro volbu pořadí řečí agentů nazvaný *postupný rozhovor s potřebou agenta mluvit.* Empirické studie prokazují výkonnost nástroje v mnoha scénářích hodnocení a překonávají porovnávaná řešení v dimenzích věrohodnosti, důvěryhodnosti, hloubky obsahu a relevance. Systém dosahuje konkurenceschopných výsledků s jinými debatními systémy s více agenty v oblasti zdravého rozumu a matematiky. Kromě toho tato práce poskytuje demonstrační webovou aplikaci pro vytváření simulací a spouštění různých scénářů.

Keywords

Agent-Based Modeling and Simulation, Multi-Agent Debate (MAD), Large Language Models (LLMs), LLM-to-LLM interaction, Computational Social Science (CSS), Simulation of Human Conversation, Human Brain

Klíčová slova

agentově orientované modelování a simulace, multiagentní debata, velké jazykové modely, vzájemná interakce velkých jazykových modelů, výpočetní sociální vědy, simulace lidské konverzace, lidský mozek

Reference

CHOCHOLATÝ, David. *Simulation of Human Interaction using AI.* Brno, 2025. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Radek Hranický, Ph.D.

Rozšířený abstrakt

S rozvojem zpracování přirozeného jazyka a zejména úlohy generování textu za použití velkých jazykových modelů (LLM) se v současné době výzkum a vývoj zaměřuje na využití těchto nástrojů v mnoha oblastech a odvětvích lidského počínání. Jednou z těchto neméně významných oblastí jsou výpočetní sociální vědy. Mezi jejich hlavní záměry patří modelování, simulace a analýza věrohodného lidského chování a konverzace ať již dvou osob či více osob současně.

Tato práce popisuje úvod do umělé inteligence (AI) od základů přes generativní AI až k aktuálnímu stavu LLM. Současně provádí rozsáhlý průzkum odborné literatury k danému tématu. Na základě významných publikací jsou popsány a rozebrány nejdůležitější přístupy jak k vytváření věrohodných LLM agentů, s využitím znalostí o funkcích lidského mozku a lidského chování, tak k modelování a návrhu věrohodných simulací lidské konverzace. Následně tato práce představuje a navrhuje vlastní experimentální prostředí s využitím nejnovějších poznatků, přístupů a technologií.

Obecně pro tvorbu multiagentního systému s využitím LLM existují dva přístupy. První využívá jediného jazykového modelu pro simulaci interakce všech agentů. Hlavní výhodou takového přístupu je jeho nižší výpočetní náročnost oproti ostatním systémům a přirozenější provázání jednotlivých zpráv. Druhou možností je reprezentace každého agenta pomocí jednoho jazykového modelu, kdy každý agent využívá alespoň jednoho vlastního LLM. Takový system disponuje především sociologickým realismem. Na základě publikované literatury na toto téma, výsledků měření a doporučení autorů vlastní systém aplikuje druhý z těchto přístupů. Samotný návrh experimentálního prostředí se zaměřuje na výběr multiagentního přístupu k modelování interakce, návrh agenta, modelování jeho vlastností a charakteru a simulační scénáře.

Nejdůležitější částí, která má významný vliv na úlohu zajištění co nejvíce věrohodné lidské konverzace je návrh jediného LLM agenta, který se skládá z mnoha komplexních komponent. Na nejvyšší úrovni lze rozdělit jednotlivé segmenty na *kognitivní moduly* a *modelování charakteru osoby*. Kognitivní moduly reprezentují jednotlivé významné procesy lidského mozku, a to kognitivní funkce. Konkrétně tato práce uvažuje tzv. *paměťový modul* (memory module), *modul reflexe* (reflection module), *modul plánování* (planning module), *akční modul* (action module), *modul vnímání* (perception module) a *modul přirozeného jazyka* (natural language module). Dalším důležitým konceptem je získávání dat pro modelování charakteru osoby. Tato práce uvažuje celkem dva přístupy. Prvním je generování popisu osoby od základu pomocí jediného LLM. Druhý z těchto přístupů reprezentuje konverze dat poskytnutých od externího zadavatele, společnosti Lakmoos AI, s.r.o., do požadovaného formátu. S využitím těchto informací lze získat věrohodnější chování jednotlivých osob. Současně dalších vylepšeních lze docílit pomocí definování vědomostních úrovní, ktere určují znalost osoby především podle věku a dosaženého vzdělání.

Jedním z hlavních konceptů, který tato práce představuje, je určení pořadí, ve kterém simulované osoby konverzují. Vzhledem k nevyhodujícím přístupům, které byly doposud publikovány, tato práce navrhuje nový koncept nazvaný *postupný rozhovor s potřebou agenta mluvit.* Hlavním benefitem daného návrhu je dynamické určení pořadí na základě vnitřních stavů agenta. Takové chování blíže odpovídá reálné situaci. Navržené experimentální prostředí bylo implementováno včetně rozšiřujícího uživatelského rozhraní. Tento nástroj byl následně publikován a je volně dostupný pro širokou veřejnost.

Následně tato práce experimentálně hodnotí navržené simulační prostředí s dalšími uvažovanými systémy. Nejprve je testována volná diskuze, kdy osoby spolu komunikují po předem stanovený počet vyměněných zpráv. Pro tento způsob interakce jsou vytvořeny

dva experimentální přístupy s využitím jazykového modelu jako hodnotitele, a to zaprvé přiřazení skóre každému systému pomocí několika kritérií a zadruhé jejich párové porovnání. Navržený systém *PerSimChat* dosahuje nejlepších výsledků v dimenzích zaměřených na přirozenost konverzace. Současně tato práce poskytuje testování skupinové debaty agentů, kdy jejich cílem je dosáhnout konsenzuálního řešení. Při porovnání navrhovaného systému s existujícími řešeními, i když jeho hlavním úkolem není řešení komplexních úloh, představený nástroj dosahuje konkurenceschopných vlastností a přesností. V návaznosti na volnou diskuzi byla provedena studie reálnými lidmi s využitím veřejného dotazníku a získány cenné poznatky o preferencích dobrovolných hodnotitelů. Nakonec jsou důkladně rozebrány případy použitích, na kterých poukazuji na benefity a limity navrženého systému.

V neposlední řadě jsou diskutovány získané poznatky, navržena možná vylepšení, uvedeny oblasti pro budoucí rozvoj podobných simulačních nástrojů a následně také limitace aktuálního stavu včetně etických aspektů této práce.

Simulation of Human Interaction using AI

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Radek Hranický, Ph.D. The supplementary information was provided by Bc. Jan Polišenský, Ing. Petr Patzelt, Juraj Štibrány, Bc. Matej Koreň, Bc. Adam Žitník, and Bc. Kamila Zahradníčková, MSc. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
David Chocholatý
May 14, 2025

Acknowledgements

I would like to thank my supervisor Ing. Radek Hranický, Ph.D., for his professional guidance, supportive approach, willingness throughout the writing of this thesis, and for ensuring excellent communication and cooperation with the external partner. I would also like to thank Bc. Jan Polišenský for providing tools and concepts from the company Lakmoos AI s.r.o., for sharing valuable expert knowledge, and for the enriching conversations that significantly influenced the current form of this thesis. My thanks also go to Ing. Petr Patzelt and Juraj Štibrány for their willingness and cooperation in providing and presenting internal data from Lakmoos AI s.r.o., Bc. Matej Koreň and Bc. Adam Žitník for support in deploying this work for an external partner, Bc. Kamila Zahradníčková, MSc. for providing a wealth of expertise and feedback from the field of psychology, and to Bc. Josef Kuba for the inspiring discussions and suggestions regarding potential experiments. Especially, many thanks to all the volunteers for completing the public survey. Last but not least, I would like to thank the company Lakmoos AI, s.r.o. for providing the computational resources.

Contents

List of Figures

Chapter 1

Introduction

The concept of artificial intelligence, already widely known among the general public, is reaching unprecedented proportions. With the advent of the latest trends and innovative research discoveries, the boundaries of our capabilities and human understanding are being pushed. Especially in recent years, the subfield of this both praised and reviled area of information technology, natural language processing, has experienced its most significant growth since the introduction of syntactic structure by Noam Chomsky. With the evolution and availability of computing power, we are reaching new possibilities with model modification regarding model size and moving toward using large language models. These models are automated assistants, so-called „chatbots.“ With the development of such agent technology and large language models, people are becoming increasingly aware of the similarity and plausibility of assistants to those represented by humans. Based on this, humanity began to ask a question, which is quite appropriate, namely how plausibly such assistants can mimic human behavior and provide a plausible simulation of a conversational person. That reflects what the believability of a conversation is from a human perspective, how well the assistant expresses emotions, and how reasonable its decision-making processes are.

To scale this task further, we can ask how to use multiple models cooperating simultaneously instead of a single language model. At the same time, how can we simulate a believable human conversation with this system? Through such a simulation, can we solve problems through debate, obtaining and analyzing vital information that is answered not by one but by several assistants simultaneously, who agree on a final solution? Does such a solution give us a higher probability of correctness than an answer from just one automated worker? These are precisely the questions, alongside other questions, that this thesis seeks to answer. At the same time, it is currently an area of interest for academic research and the development of new approaches by technology companies.

Several very successful publications have already been written on a similar topic. However, the authors often focus on a particular subset of this complex task and do not cover a broader range of possibilities. Published literature usually uses manually created or only automatically generated data about autonomous agents that do not reflect real personas. The way multiple agents communicate with each other often does not correspond to the real-world situation and does not completely replicate natural human communication. Additionally, not much emphasis has been placed on the plausibility of the similarity of the so-called agent to a real human person. Current multi-agent simulations often lack realistic dialogue flow, context retention, and believable character behavior, which this thesis aims to address.

This work proposes and implements an experimental system to simulate natural human conversation with large language models, both free conversation and debate. To ensure natural communication, the thesis presents a communication pattern of *One-By-One Talk with Agent's Need to Talk*, exploiting the emotional state of the autonomous agent. The emotional level is also used when storing or dynamically retrieving memories. The designed approach leverages the capabilities of large language models in decision-making for value setting or multiple choice. The system increases the believability of a virtual replica of a persona with character modeling also based on real-world data, especially the persona's characteristics and traits.

The created framework is tested with multiple evaluation scenarios for free discussion and group debate. The system possesses comparable performance capabilities in these experiments and outperforms compared solutions on multiple evaluation dimensions. Along with that, a public survey with the help of human evaluators was conducted. In addition, I provide a subjective study on numerous use cases. The user interface is also designed for the proposed framework to increase the usability of the tool.

Overall, the thesis divides its tasks into five chapters. Following the introduction given in this chapter, Chapter 2 summarizes the basics of artificial intelligence, specifically neural networks and natural language generation approaches. Chapter 3 then follows with an introduction to neuroscience and social behavior, including a discussion of the most essential processes and functionalities of the human brain and the observed characteristics of human behavior during social interaction. At the same time, this chapter discusses the foundations of autonomous systems and related technical literature published to date. Chapter 4 presents the design of the solution, including the design of the agent itself individually and the scheme and concept of communication between such agents. In addition, it presents possible communication scenarios. Subsequently, Chapter 5 describes the implementation of the designed framework, including the system's core concepts. Chapter 6 provides the experimental evaluations of the PerSimChat framework, while Chapter 7 builds on these experiments and discusses the results. This also directs the next steps for the research field, from the development of technologies to related simulations. Simultaneously, it states possible future directions, limitations, and ethical considerations. The final chapter, Chapter 8, summarizes the existing knowledge on simulating human conversation in an experimental setting, discusses published approaches, and describes the proposed design, its implementation, and conducted evaluations.

Chapter 2

Neural Networks for Natural Language Generation

In this chapter, the architecture and concepts of the *neural network* are described from the very beginning. With that, the components are listed, the training process is explained, and the related types of general neural networks are shown. With significant advancements, I describe the *transformer* models in depth. On top of that, the essentials of generative AI and state-of-the-art natural language processing with language models are also presented.

2.1 From Neural Networks to Deep Learning

In human nature and nature itself, there are many uncertain phenomena. The efforts and goals of researchers are to model such relations using concepts, theories, models, algorithms, and methods [78]. There are designed multiple basic to more advanced systems [86]. However, a significant breakthrough came, especially for natural language processing, with the onset and start of widespread use of *neural networks*.

Inspired by the nature of a human brain [69], in recent years, neural networks predominantly use *perceptron* [71] as a building block in a *layer*. Using two or more trainable layers, we have a *feedforward architecture* [83]. This construction is also called a *multi-layer perceptron* (MLP).

MLP is a quintessential example of a *deep learning* model. The term „deep“ is taken over from the use of multiple layers [40].

2.1.1 Components and Training of Multi-Layer Perceptron

The multi-layer perceptron (MLP) is composed of several essential components, namely *neurons*, *layers*, *activation functions*, and *weights and biases*.

In MLP, the neuron is represented by a perceptron [71]. A single perceptron itself serves as a binary classifier. It can decide whether the input belongs to one class or another.

Combining multiple perceptrons, we get a layer of neurons. In such an architecture, every neuron processes the same input independently. The individual outputs of the neurons together form the output vector. Using multiple perceptrons, we can force every perceptron to focus on detecting different patterns in the input data.

By connecting several layers in a row, we get, from a general point of view, the *feedforward neural network*. This network has two or more trainable layers. The first layer is called the *input layer*, the last layer the *output layer*, and the layers between the first and last the *hidden layers*. When each neuron in one layer is connected to every neuron in the preceding layer, we call this layer a *fully-connected* layer.

These models are called feedforward because there are no feedback connections in which the outputs of the model are fed back into itself [40].

We scale the input values by the connections *weights* to represent the strength of the connections between neurons. The weight indicates the importance of each feature. To shift the activation function to fit the data better, we use *bias*, the additional parameter to the weighted sum of input connections.

Using our model with linear layers only will result in a linear transformation of the input. Based on that, the *activation function* has to be used to extend the concept. Common activation functions used are the *logistic sigmoid* function, *hyperbolic tangent* (Tanh), *rectified linear unit* (ReLu), or *softmax* activation function.

We can minimize a predefined loss function by adjusting the weights and biases. This function serves as an evaluation of how well our model performs. The goal of neural network training is to iteratively change these values to enable the neural network to learn patterns from the input data. Based on that, the network can generalize to previously unseen examples in a test evaluation.

2.1.2 Types of Neural Network Architectures

Beginning with the basic network architecture for simple tasks called *feedforward neural network*, we can obtain complex models, preferably specializing in specific tasks, by modifying this architecture.

Convolutional Neural Networks

This type of network specializes in processing data with a known grid-like topology. These can be, for example, time-series data (1-D grid) or image data, which can be thought of as a 2-D grid of pixels. The name „convolutional“ indicates that the networks employ a mathematical operation called *convolution*, a specialized linear operation. Convolutional networks are neural networks that use convolution instead of general matrix multiplication in at least one of their layers [40].

Recurrent Neural Networks

Extending the feedforward neural networks by including feedback connections, we get *recurrent neural networks* (RNNs) [40]. This family of networks is specially designed to process sequential data. As convolutional networks can readily scale to images with large width and height, and some convolutional networks can process images of variable size, recurrent networks can scale to much longer sequences than would be practical for networks without sequence-based specialization [40]. The difference between RNN and the classical feedforward neural network is shown in Figure 2.1.

By improving the performance of the types of neural network architecture presented, the revolutionary concept, especially for natural language processing tasks, is the *transformer* model.

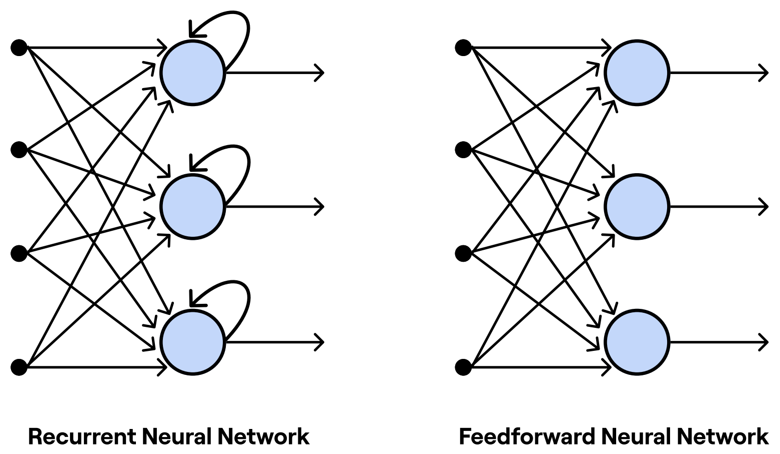

Figure 2.1: The comparison of recurrent neural network and feedforward neural network.

2.2 Transformers Architecture and Attention Mechanism

Neural networks completely changed the field of machine learning. Especially for sequential data, such as audio, video, or text, convolutional or recurrent neural networks [50, 58, 83] were used primarily. Despite the benefits, these networks struggle with long-range dependencies across inputs. Additionally, they face challenges in parallelization. To solve such detriments, a significant advance was made by a *transformer* architecture including an *attention* mechanism [99].

Transformers uses the *encoder-decoder* structure [5, 18, 93]. This architecture learns to encode an input sequence into a vector representation in the encoder part. The decoder transfers a given vector representation (encoder output) back into an output sequence while having the previous tokens from the target sequence (shifted right). This mechanism is presented in Figure 2.2. Models themselves can consist of only a decoder or an encoder.

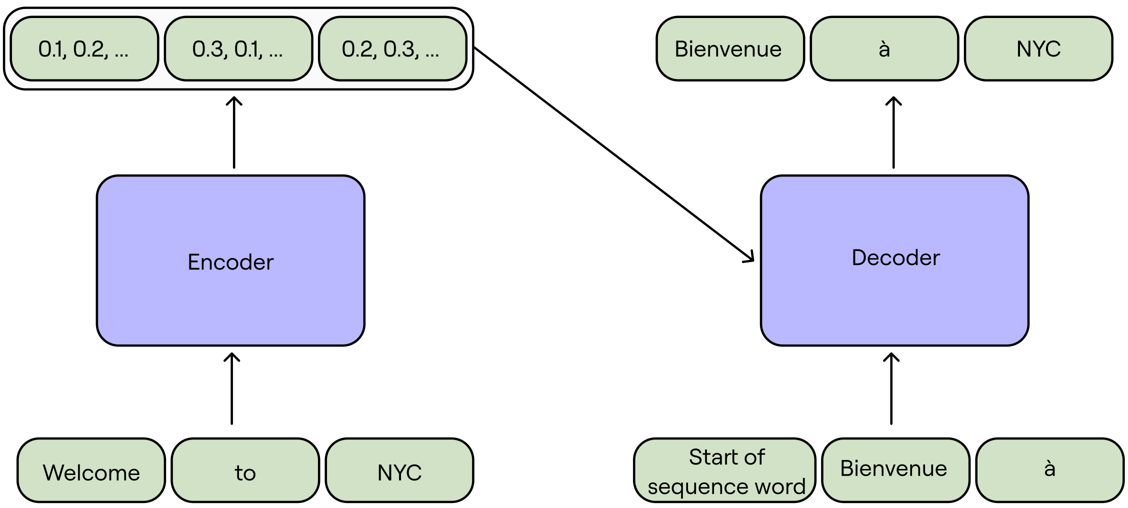

Figure 2.2: The basic view of the encoder-decoder mechanism on an example of translating English sentences into French.

By eschewing recurrence, the transformer model architecture relies entirely on the self-attention mechanism to draw global dependencies between input and output, without sequence-aligned recurrent or convolutional neural networks [99]. Self-attention is an attention mechanism relating different positions of a single sequence to compute a representation of the sequence.

In each encoder and decoder stack processing, the transformer captures vector representation information through layers of self-attention and feedforward layers. More precisely, the encoder is composed of a stack of six identical layers, while each layer has a multi-head self-attention mechanism and a position-wise fully connected feedforward network. The decoder, having the same number of identical layers, inserts another multi-head attention over the output of the encoder stack, as demonstrated in Figure 2.3.

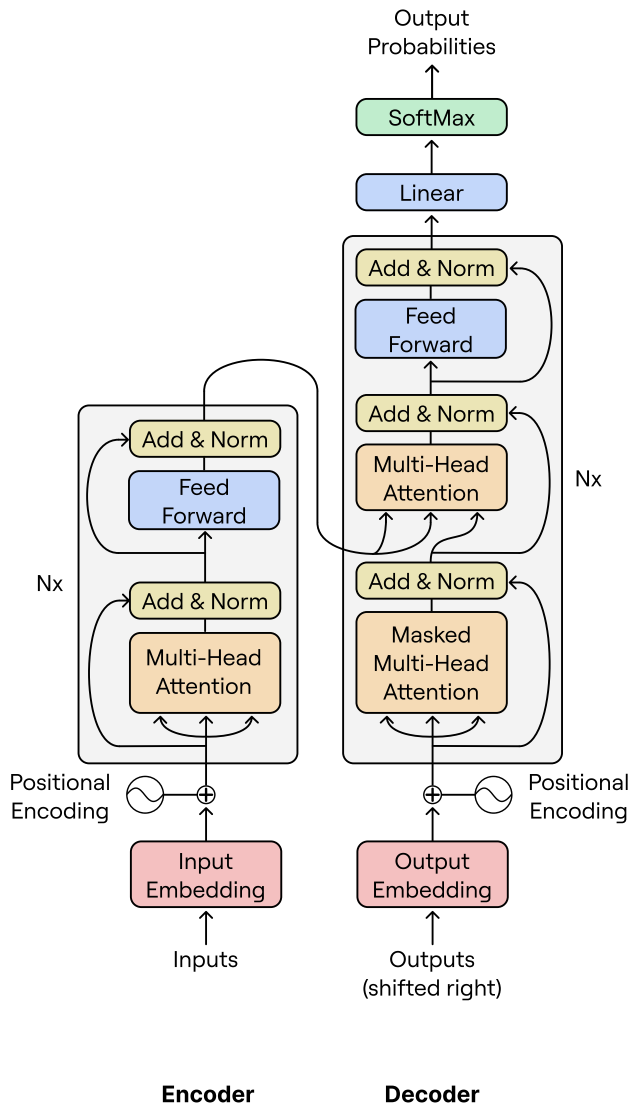

Figure 2.3: The transformer model architecture with encoder and decoder structure.

An attention function can be described as mapping a *query* and a set of *key-value* pairs to an output. The output is computed as a weighted sum of the values.

The *query* is a vector that represents a specific input sequence position. The query vector seeks to find the input parts that are relevant to the current position. The *key* is a vector associated with a specific position in the input sequence. It determines how much attention should be given to other positions. Lastly, the *value* is a vector that gets passed along when the attention mechanism aggregates information from other tokens.

The so-called multi-head attention mechanism is assembled from several single particular attention mechanisms, called *scaled dot-product attention*, see Figure 2.4. The scaled dot-product attention and *multi-head attention* are defined by Definition 2.2.1 and Definition 2.2.2, respectively. The multi-head computation is also demonstrated by Figure 2.5.

Definition 2.2.1 (Scaled Dot-Product Attention)**.** Let Q denote the query matrix, K the keys matrix, V the values matrix, and A^T denote the transposition of the matrix A. Let $softmax$ denote the softmax activation function defined in a standard manner. The *scaled dot-product attention* is defined as

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

where d_k denotes the dimension of queries and keys, and d_v the values dimension [99].

Definition 2.2.2 (Multi-Head Attention)**.** Let h denote the number of heads, d_{model} denote the dimension of keys, values, and queries. Let Q denote the query matrix, K the keys matrix, V the values matrix, O the output matrix, d_k denote the dimension of queries and keys, and d_v the values dimension. Let *Concat* denote the concatenation function defined in a standard manner. The *multi-head attention* is defined as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O, \tag{2.2}$$

where $\text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ and the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W_i^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$, for $i = 1, ..., \text{h}$ [99].

In this section, I described the core concept of natural language processing with transformer models. The overall encoder-decoder architecture was explained, including the so-called attention mechanism that uses keys, queries, and values to capture essential features in the input data.

2.3 Generative AI with Large Language Models

With powerful tools such as neural networks, the primary focus is on recognition, regression, and classification tasks. However, the new direction gave rise to new opportunities. What if we aimed to generate new data rather than processing real-world data? This task is a primary goal of the general *generative AI*.

In this section, I describe what generative AI is from a basic point of view. Subsequently, I focus on the *large language models* (LLMs) as a part of the generative AI and describe their strengths and weaknesses.

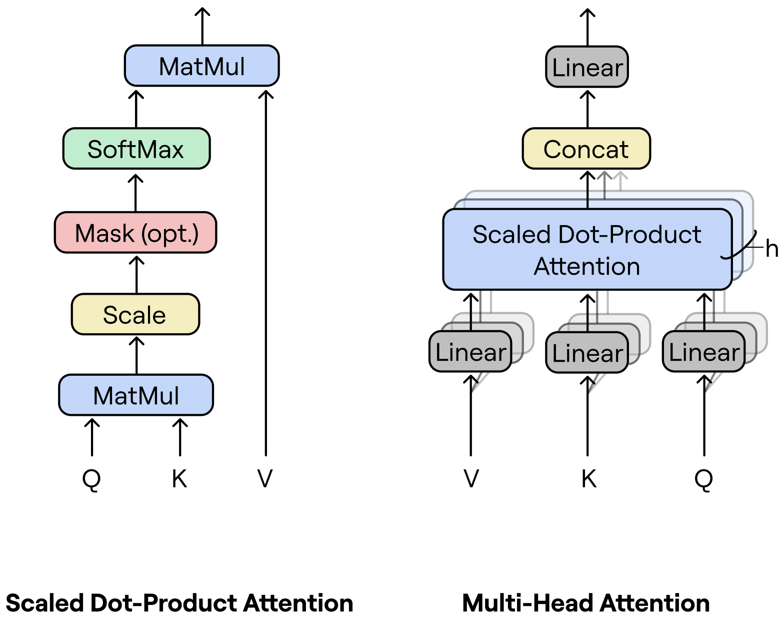

Figure 2.4: The scaled dot-product attention with the multi-head attention, which consists of several attention layers running in parallel.

2.3.1 General Generative AI

Generative AI refers to a class of algorithms and models within AI and *natural language processing* (NLP) that are designed to generate new, previously unseen data similar to existing examples employing a variety of techniques [82]. Generative AI models are trained on large datasets of existing content. These models learn the underlying patterns and structures present in the training data and use that knowledge to create novel instances that resemble the original data [45]. Subsequently, we can divide the generative models into the three main types: *Variational Autoencoder models*, *Generative Adversarial Networks*, and *Autoregressive Models*, of which the last type is most important for this work.

In the context of generative AI, autoregressive models are a class of likelihood models that generate new sequential data by predicting the next value in a sequence based on the previous values. These models involve modeling the probability distribution of each element in a sequence given the entire history of previous elements [45]. Such an approach is visualized in Figure 2.6 and Figure 2.7. This ability makes autoregressive models well-suited for various NLP tasks where the ability to understand and generate coherent sequences is essential [70].

2.3.2 Toward Language Modeling with Large Language Models

LLMs are the key component behind language modeling, specifically the text generation task. This task aims to force the models to generate human-like text. They consist of large pre-trained transformer models trained to predict the next word (more precisely, the token) given some input text[1].

A language model trained for *causal* language modeling takes a sequence of text tokens as input and returns the probability distribution for the next token. By using the decoder-only

[1] https://huggingface.co/docs/transformers/llm_tutorial, accessed: 2024-12-06.

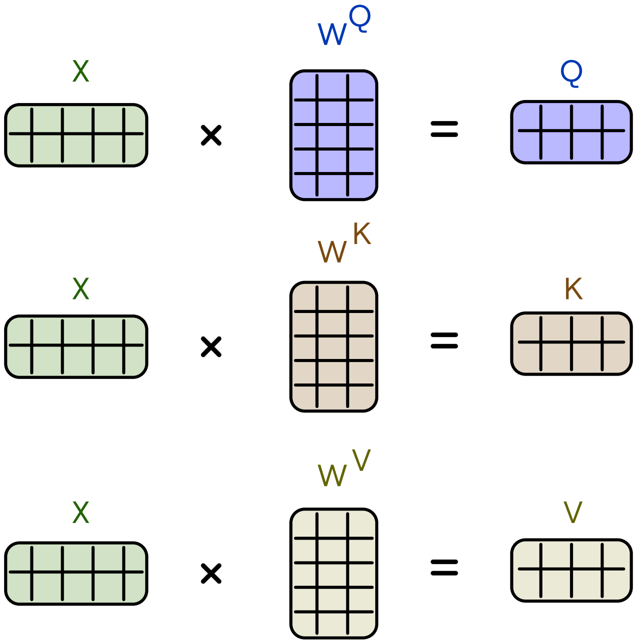

Figure 2.5: The attention mechanism with a query matrix Q, key matrix K, and value matrix V. The W^Q, W^K, and W^V denote the corresponding matrices in order, and X denotes the input vector.

transformer-based architecture, the models predict a sequence autoregressively, predicting one token at a time based on the preceding tokens without relying on an explicit encoder [45].

A critical aspect is the selection of the next token from the probability distribution. This can be as simple as a most likely token from the probability distribution or as complex as applying a dozen transformations before sampling from the resulting distribution[2]. The large language model *temperature* can also influence how the next token is selected. It is a parameter that sets the balance between predictability and creativity in the generated text. More specifically, it adjusts the probability distribution from which tokens are sampled[3].

2.3.3 Challenges and Limitations of the Large Language Models

This section discusses some of the main limitations of LLMs and the potential for improvements.

Data Bias and Ethical Concerns

In a training process, LLMs can inherit and amplify biases present in training data. Some of the biases that can be included are gender, race, and cultural biases[4]. Based on that, LLMs can dispose of unethical outputs, such as discriminatory language.

[2] `https://huggingface.co/docs/transformers/llm_tutorial`, accessed: 2024-12-06.

[3] `https://www.hopsworks.ai/dictionary/llm-temperature`, accessed: 2024-11-30.

[4] `https://www.datacamp.com/blog/understanding-and-mitigating-bias-in-large-language-models-llms`, accessed: 2024-12-06.

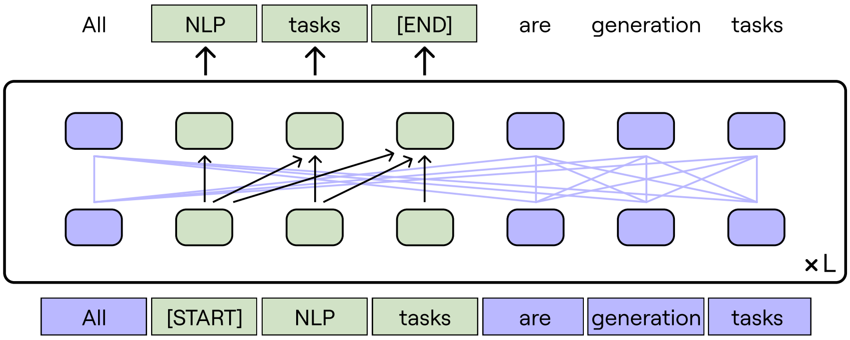

Figure 2.6: Concrete example of a generation using a language model. Text spans (green part) are blanked out. They are generated autoregressively.

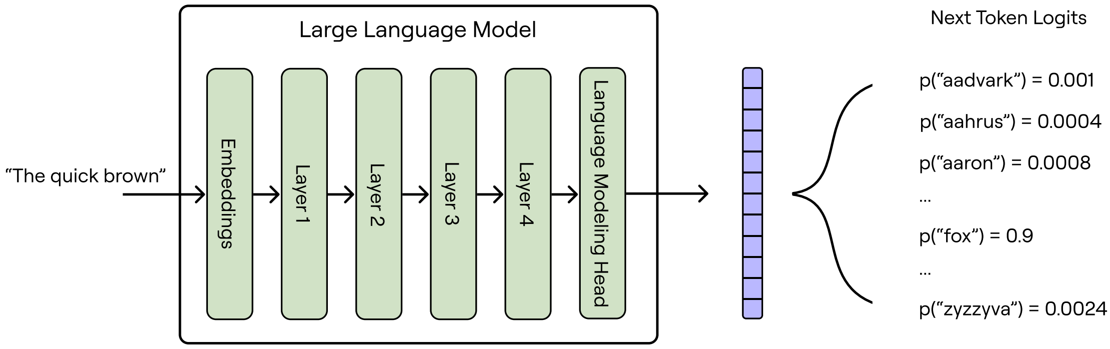

Figure 2.7: The example of an autoregressive generation for the next token prediction.

Large Language Model Hallucinations

In the context of LLM, hallucinations refer to generating nonsensical or unfaithful content to the provided source [53]. This problem leads to incorrect information[5]. We can divide hallucinations into multiple categorizations, for example, factuality hallucinations or faithfulness hallucinations.

Context Length and Memory Constraints

With a focus on the generation task itself, LLMs are limited by the input text length they can process in a single instance. Although some models can handle the longer context, they struggle with understanding long-term dependencies in complex tasks [72].

Potential for Misinformation

Looking at the credibility of the information provided, LLMs can generate factually incorrect or misleading information. This mainly affects areas such as healthcare, news, or science, where the accuracy of information is critical [72].

[5]https://www.lakera.ai/blog/guide-to-hallucinations-in-large-language-models, accessed: 2024-12-06.

2.3.4 Beyond Pre-Trained Models with Prompt Engineering

In this section, building on general generative AI and especially LLMs, I present advanced techniques to obtain the best possible results using these models. The concepts and techniques for *prompt engineering* are described, introducing the various methods used nowadays.

The LLMs have relatively large and well-functioning working memory. Whatever fits into the *context window* is immediately available to the transformer through its internal self-attention mechanism[6]. In this, the context window refers to the textual range that an LLM can process at the time the information is generated[7]. Following that, various so-called *prompting techniques* are involved as a part of the prompt engineering field.

Starting from the beginning, *prompt* is the input we provide to the model to elicit a specific response. This can take various forms, ranging from simple questions or keywords to complex instructions, code snippets, or creative writing samples. The effectiveness of the prompt directly influences the quality and relevance of AI's output[8]. Prompt engineering is a discipline for developing and optimizing prompts to use language models efficiently. It is used to improve the capacity of LLMs on a wide range of common and complex tasks[9]. The prompting techniques are the approaches to creating the prompts that provide the model with context, instructions, and examples that help it to understand the intent and respond in a meaningful way[8].

Various advanced techniques are available alongside the basic general recommendations for the prompt format and structure. Some of the most popular methods are the *zero-shot prompting*, *few-shot prompting*, *chain-of-thought prompting* (CoT), *contextual prompting*, and *meta prompting*.

Zero-Shot Prompting

Zero-shot prompting describes a technique in the prompt used to interact with the model that does not contain examples or demonstrations. The zero-shot prompt directly instructs the model to perform a task without additional examples to steer it[10].

Few-Shot Prompting

Few-shot prompting [54, 95] can enable in-context learning, where we provide demonstrations in the prompt to steer the model to better performance. The demonstrations serve as conditioning for subsequent examples where we would like the model to generate a response[11].

Chain-of-Thought Prompting

Chain-of-thought prompting, introduced by Wei et al. [108], enables complex reasoning capabilities through intermediate reasoning steps, leading to a more comprehensive and

[6] https://medium.com/@tonytong.ai/andrej-karpathys-keynote-at-microsoft-build-2023-8b45a2bbf22e, accessed: 2024-12-07.
[7] https://www.techtarget.com/whatis/definition/context-window, accessed: 2024-12-07.
[8] https://cloud.google.com/discover/what-is-prompt-engineering?hl=en, accessed: 2024-12-02.
[9] https://www.promptingguide.ai/, accessed: 2024-12-02.
[10] https://www.promptingguide.ai/techniques/zeroshot, accessed: 2024-12-05.
[11] https://www.promptingguide.ai/techniques/fewshot, accessed: 2024-12-06.

well-structured final output[12]. This can also be combined with short prompting to get better results on more complex tasks that require reasoning before responding[13].

Contextual Prompting

Contextual prompting involves providing a detailed context within the prompt to guide the model's response. This technique helps the model understand the task better by embedding relevant information directly in the prompt. It is beneficial for tasks that require specific background knowledge or situational awareness[14].

Meta Prompting

Meta prompting focuses on the structural and syntactical aspects of tasks and problems rather than their specific content details. The goal is to construct a more abstract and structured way of interacting with large language models, focusing on the form and pattern of information over traditional content-centric methods[15].

Advanced Prompting Techniques

Other advanced prompting techniques were published, for example, *Prompt Chaining*[16], *Self-Consistency* [105], or *Tree-of-Thoughts* [66, 112]. In addition, more framework-like techniques, such as *Self-Reflection* [87] or *ReAct* [113], are nowadays available.

Alongside prompt engineering, techniques such as *Retrieval-Augmented Generation* [59] and *fine-tuning*[17] can also be used.

[12] https://cloud.google.com/discover/what-is-prompt-engineering?hl=en, accessed: 2024-12-02.
[13] https://www.promptingguide.ai/techniques/cot, accessed: 2024-12-06.
[14] https://medium.com/@yogabalajig/prompt-engineering-techniques-and-best-practices-83bf48c850e6, accessed: 2024-12-06.
[15] https://www.promptingguide.ai/techniques/meta-prompting, accessed: 2024-12-06.
[16] https://www.promptingguide.ai/techniques/prompt_chaining, accessed: 2024-12-06.
[17] https://www.ibm.com/topics/fine-tuning, accessed: 2024-11-30.

Chapter 3

Human Interaction Modeling and Simulation

In this chapter, related work is analyzed from various perspectives. For the selection of the papers primarily focused on *computer-human interaction*, the approach was to find well-known papers and the currently published state-of-the-art articles. The data collection approach was inspired by Gao et al. [36] who follow the *Systematic Literature Review*[1]. I focused on papers containing the key terms, including, for example, „Large Language Model Agents,“ „prompt/prompting,“ „generative AI,“ „Simulating Human Interaction,“ and „Human-AI Collaboration.“ In summary, 100 scientific papers were chosen, of which, after reading the abstracts, results, and conclusions, 35 were selected as truly related. All of these articles were fully read, and finally 11 articles were chosen as deeply related to the topic.

This chapter describes the main concepts of the human brain, its physiological functions, and the essential phenomena that affect human interaction. With this knowledge, we move on to the foundations of autonomous systems using large language models (LLMs), and this work is placed among others in terms of *human-AI collaboration*. Finally, I discuss approaches from individual publications that are deeply related to the thesis and evaluate them regarding memory design choices.

3.1 Neuroscience and Social Behavior: From Human Brain to Interaction

This section disassembles the key concepts of human brain function. The mental processes, known as *cognitive functions*, essential for this work, are described. Subsequently, moving from the individual's brain, I focus on the cooperation and workflow of social groups in the real world. Various approaches are determined, and the influences of non-negligible interaction are considered.

3.1.1 The Physiological Perspective on the Human Brain

This section provides an introduction to brain structure and function. The brain is an astonishing living organ inside our heads, consisting of billions of tiny cells. The brain

[1] https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf, accessed: 2025-12-06.

enables us to sense, think, and respond to the world around us. The main obstacles that prevent us from creating a machine that can behave like real-world creatures are our limited knowledge about the brain in both its structure and its function [115].

The mental processes that allow us to receive, select, store, transform, develop, and recover information that we have received from external stimuli are the brain *cognitive functions*. This process allows us to understand and relate to the world more effectively. Cognitive functions are brain-based skills that we need to carry out any task, from the simplest to the most complex. They are related to the mechanisms of how we learn, remember, solve problems, pay attention, etc. [114].

The most crucial human brain functions and processes to this work are *perception*, *attention*, *memory*, *decision-making*, *natural language*, *planning*, *self-evolution*, *reasoning*, *reflection*, and *action*. The purposes of each function and process are as follows.

Perception

Perception is how our brain organizes and interprets sensory information from the environment. This process is influenced by various factors, including past experiences, expectations, cultural background, and attention[2]. It involves processing and subsequent interpretation of the five senses: touch, sight, sound, smell, and taste[3].

Attention

Attention is a cognitive process that involves focusing on a specific aspect of the environment or information while filtering other stimuli[4]. It helps filter out distractions, enabling us to process relevant information efficiently, which is essential for effective perception and learning [38].

Memory

Memory encompasses the processes of encoding, storing, and retrieving information. We can categorize human memory into various parts:

- *Sensory memory* is the shortest-term element of memory. It is the ability to retain impressions of sensory information after the original stimuli have ended. Sensory memory is an ultra-short memory and decays or degrades very quickly, typically in the region of 200–500 milliseconds after the perception of an item, and certainly, less than a second [114].

- *Short-term memory* acts as a „scratch-pad“ for temporary recall of the information being processed at any time. This type of memory holds a small amount of information, which is around *seven* items, based on the *Miller's Law* [71], in mind in an active, readily-available state for a short time (typically from 10 to 15 seconds, or sometimes up to a minute). That can also be thought of as the ability to remember and process information at the same time[5].

[2] https://library.fiveable.me/key-terms/neuroscience/perception, accessed: 2024-11-05.
[3] https://www.verywellmind.com/perception-and-the-perceptual-process-2795839, accessed: 2024-11-05.
[4] https://www.happyneuronpro.com/en/info/what-is-attention/, accessed: 2024-12-06.
[5] https://human-memory.net/short-term-working-memory/, accessed: 2024-12-06.

- *Long-term memory* is intended to store information over a long period. Despite the impressions of forgetting, it seems likely that long-term memory actually decays very little over time and can store a seemingly unlimited amount of information almost indefinitely. In fact, there is some debate about whether we ever „forget“ anything, or whether accessing or retrieving specific items from memory becomes increasingly complex. Short-term memory can become long-term memory through the process of *consolidation*, involving rehearsal and meaningful association. Unlike short-term memory, long-term memory encodes information for storage semantically (i.e., based on meaning and association) [114].

Decision-Making

Decision-making involves evaluating information and making choices based on individual preferences, beliefs, and values. Emotions and social contexts also influence decision processes. Integration of information from emotional responses, memory, and reasoning is required [22]. This allows people to weigh options and anticipate results [75].

Natural Language

Natural language processing is capable of understanding, producing, and using language for communication. This also includes components such as syntax (sentence structure), semantics (meaning of words), and pragmatics (contextual use)[6] [47].

Planning

Planning involves the ability to develop strategies to achieve specific goals[7]. It requires foresight and anticipation of future needs. Based on that, the actions are reorganized. Planning is also closely related to executive functions, including working memory [68].

Self-Evolution

Self-evolution refers to the capacity for personal growth through learning and experience. It includes adapting behaviors based on past outcomes and reflecting on thoughts and actions [84].

Reasoning

Reasoning is the cognitive process of drawing conclusions or making inferences based on available information or premises[8]. It is critical for decision-making, allowing people to analyze situations and formulate solutions [91].

[6] https://www.communicationcommunity.com/5-domains-of-language/, accessed: 2024-11-25.
[7] https://krestonpedabo.com/the-significance-of-goal-setting-and-strategic-planning-in-organisations/, accessed: 2024-12-06.
[8] https://library.fiveable.me/key-terms/cognitive-psychology/reasoning, accessed: 2024-11-25.

Reflection

Reflection enhances learning and personal development[9]. It allows individuals to evaluate past actions, learn from mistakes, improve their performance, and improve their skills[10]. Reflection involves critical thinking about thoughts, experiences, and knowledge[11].

Action

Action refers to the execution of decisions and the participation in purposeful behavior. The ability to take action is closely related to motivation, decision-making, and planning [7]. It also includes the coordination of various cognitive functions to perform a (physical) action.

This work's last important human brain ability is the *emotion*. Emotion is defined as a complex reaction pattern involving experiential, behavioral, and physiological elements by which an individual attempts to deal with a personally significant matter or event[12]. Emotional experiences have three components: a subjective experience, a physiological response, and a behavioral or expressive response[13].

Defining emotions is a task that is not yet complete. Many researchers are still proposing theories on what makes up our emotions, and existing theories are constantly being challenged[13].

3.1.2 Social Behavior and Interaction

Human behavior and interaction can be analyzed from many points of view. For this work, I focus on social interaction in group discussions and debates with decision-making and consensus-reaching. Many factors can influence the quality and flow of the interaction. Among the most important ones, I advise *individual differences*, *emotions*, and *group dynamics*. This chapter discusses each of the areas of interest.

Individual Differences in Social Interactions

First, I would like to emphasize the difference between the *characteristics* of a person and their *traits*. Person characteristics refer to the observable and measurable aspects of a person. These typically include general information, such as demographic details. On the other hand, traits delve deeper into a person's psychological and behavioral attributes. These include, for example, introversion or extroversion, openness to new experiences, emotional stability, etc.

The personality traits impact the individual's behavior in a social interaction [12]. Following the *Attachment Theory* [11], *Social Learning Theory* [8], *Life History Theory* [23], and *Cognitive Behavioral Theory* [10], the typical result is that historical experiences and characteristics also have a significant impact. The important theory is the *Big Five Per-*

[9] https://aithor.com/essay-examples/the-importance-of-reflective-cycle-in-personal-and-professional-development, accessed: 2024-11-25.

[10] https://medium.com/@shilpa.ukau/self-development-through-reflective-practice-75cd36bbd2ff, accessed: 2024-11-26.

[11] https://uwaterloo.ca/writing-and-communication-centre/critical-reflection, accessed: 2024-11-26.

[12] https://dictionary.apa.org/emotion, accessed: 2024-11-27.

[13] https://online.uwa.edu/news/emotional-psychology/, accessed: 2024-11-27.

sonality Traits concept by Costa and McCrae [21], which highlights five essential traits: *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*.

Openness Openness (also referred to as openness to experience) emphasizes imagination and insight the most out of all five personality traits [80]. People with a high level of openness tend to have a broad range of interests. They are curious about the world and other people and are eager to learn new things and enjoy new experiences[14].

Conscientiousness Conscientiousness is defined by high levels of thoughtfulness, reasonable impulse control, and goal-directed behaviors [80]. Highly conscientious people tend to be organized and mindful of details. They plan, think about how their behavior affects others, and are aware of deadlines[14].

Extraversion Extraversion (or extroversion) is a personality trait characterized by excitability, sociability, talkativeness, assertiveness, and a high level of emotional expressiveness [80]. People who are high in extraversion are outgoing and tend to gain energy in social situations. Being around others helps them feel energized and excited[14].

Agreeableness Agreeableness includes attributes such as trust, altruism, kindness, affection, and other prosocial behaviors [80]. People with agreeableness tend to be more cooperative, while those low in this personality trait tend to be more competitive and sometimes even manipulative[14].

Neuroticism Neuroticism is a personality trait characterized by sadness, moodiness, and emotional instability [80]. Individuals who are high in neuroticism tend to experience mood swings, anxiety, irritability, and sadness. Those with a low level of this personality trait tend to be more stable and emotionally resilient[14].

Roles of Emotions in Interaction

Emotions do not have an impact on the individual themselves; they also impact the context of social interaction. They are shaped not only by individual experiences but also by societal norms and contexts. This directly impacts the interaction of people [4]. It influences decision-making, communication, and relationship-building [57]. With that, the ability of how individuals can process and manage their own emotions is central to successful social interactions [39].

Additionally, people adapt their behavior based on the emotions expressed by others [98]. In alignment, emotions impact group dynamics, including cohesion, conflict resolution, and collective decision-making [9].

Group Dynamics

In the context of decision-making, the collective aggregation of information, perspectives, and knowledge often leads to better outcomes than the decisions made by individual experts. With that, these groups can manage decisions effectively [92].

[14] https://www.verywellmind.com/the-big-five-personality-dimensions-2795422, accessed: 2024-12-07.

However, with a focus on reaching a common consensus, based on the *Abilene Paradox* [48], the group members may fail to communicate their true preferences. Instead, they conform to perceived group consensus. This can lead to decisions that are not ultimately supported by any individual member. Social influences and group norms can affect how groups reach a consensus. Peer pressure and the influence of more dominant or respected members can influence the group's opinions [52].

Communication strategies or hierarchies in a social group can be used to improve the decision-making process and the quality of the final consensus. Groups can reach a consensus through open communication, in which all members have the opportunity to express their opinions and concerns in a democratic decision-making process [44]. Building a hierarchy of the group with a basic setting; in some groups, a facilitator or mediator may be involved to guide the group through the decision-making process. It can help manage group dynamics, ensure equal participation, and keep focused on the task [32]. Typically, such a position may belong to a team leader, who holds decision-making authority and directs the workflow. The leader's role is often to set goals, allocate resources, and resolve conflicts [77]. We can distribute the leadership dynamically, meaning the leadership role can be shifted based on expertise or the task [88]. In general, achieving consensus can take time in the decision-making process. Groups may need several rounds of discussion, reflection, and refinement before reaching a final agreement.

In this section, I have described and discussed the most essential parts of the human brain from a physiological point of view, which are essential for this work. At the same time, I have focused on human interaction, the influences on its execution, and the procedures to achieve the best possible results.

3.2 Foundations of Autonomous Systems and Human-AI Interaction

In this section, the important concepts of how LLM agents are built and how they behave in the simulation environment are described. With that in mind, for this work, the approaches to how humans interact with AI and with LLM are not less important. This has also been the subject of extensive research in recent times.

3.2.1 Large Language Model Autonomous Agents in the Simulation Environment

Simulation encompasses the emulation of real-world processes or systems by employing mathematical formulas, algorithms, or computer-generated representations to mimic their behaviors or characteristics. *Agent-based* modeling and simulation focuses on modeling complex systems by simulating individual agents and their interactions within an *environment* [34, 67].

Starting from the very beginning, the environment, whether static or evolving, introduces conditions, instigates competition, defines boundaries, and occasionally supplies resources that influence agent behavior [19, 34]. Agents may be constrained or influenced by the environment, and their interactions can affect the environment itself [34].

The *agent*, with specific characteristics and states, is described as an autonomous entity that can perceive the environment using its sensors, make a judgment based on the current state, and, consequently, act based on the actions available [72].

In the context of LLMs, agents are based on a single instance or multiple instances of LLM. These models use the given input for the decision-making process. In alignment, these agents can also have access to additional tools.

Agents *interact* with each other and their environment through predefined mechanisms. Interactions can be direct (agent-to-agent) or indirect (agent-to-environment or environment-to-agent) [34]. The goal is to mirror the behaviors in reality based on predefined or adaptive rules [28, 67].

In general, LLM agents are constructed from submodules, and these modules usually represent individual cognitive functions. Based on that, I follow the unified framework for the architecture design of an autonomous agent based on LLM, proposed by Wang et al. [101].

The modules can be divided into a *profile module*, *memory module*, *planning module*, and *action module*.

The profile module is responsible for the profile content and the profile generation strategy. The memory module defines the memory structure, the formats of the memories, and the performed operations. These can also include a *reflection* task. The planning module is responsible for the reasoning and planning of future actions based on memories. Finally, the action module executes the created plan. How these modules are implemented refers to the specificity of the implementation of the individual publications, and typically, various approaches for each are proposed.

In addition, other abilities and concepts of the agent can be included, such as *perception*, *decision-making*, *self-evolution*, or *reasoning*.

3.2.2 Human-AI Collaboration

In recent years, the main focus in the AI research space has been on the interaction between humans and primarily LLMs. This is sometimes called *human-AI* collaboration.

The researchers approach these tasks from various perspectives. Based on the division of Kim et al. [56], we can generally divide this field research into the following subparts:

- *Human-Robot Agent* interaction: In this interaction, the human interacts with the LLM using a physical robot, which has implemented speech through a text-to-speech module.
- *Human-Text Agent* interaction: In this interaction mode, the human interacts with the LLM through a chat environment.
- *Human-Voice Agent* interaction: The human interacts with the LLM using the text-to-speech module.

From this work perspective, I focus only on the text agents. From the human-text agent interaction perspective, Gao et al. [36] provided a taxonomy on how humans interact with the text agent interface, mostly known as a chat. They have split the approaches of how humans communicate with text agents as follows: *standard prompting*, *user interface*, *context-based*, and *agent facilitator*.

Based on this division, this work primarily approaches the context-based mode and the agent-facilitator mode.

3.3 Interaction Among Large Language Models

The interaction between humans and LLM agents has already been described, known as human-AI collaboration. However, the actual state-of-the-art research focus, motivated by the scaling of the system studies, is on the interactions between the LLMs themselves. There are various views and approaches to such a task.

The primary motivation for these simulations and this work, in general, is to simulate human behavior in real-life interaction and communication scenarios. This is a subfield of the parent area known as *computational social science* (CSS) [6, 41, 119]. Based on that, I can analyze individual and group behavior, study the concepts behind reaching a consensus, and use these results to predict and analyze human behavior.

I divide the related work into five subfields based on the focus of the individual works: *agents simulation in a game environment*, *artificial agents interaction*, *human interaction modeling*, *evaluation*, and finally, *communication patterns*. The characteristics of each field are described separately in the sections. I want to mention that many works cover more than one of these areas.

3.3.1 Agents Simulation in Game Environments

This section describes related work published on the simulation of artificial agents in a game-like environment. Based on that, most works dispose of the sandbox environment inspired by The Sims[15] computer game [61, 79, 81, 107].

Starting from the beginning with the development of a single agent, a significant concept of the evolution of the VOYAGER learning agent in the Minecraft[16] game environment was presented by Wang et al. [100]. In the context of strategic negotiation, persuasion, and cooperation with humans, Bakhtin et al. [30] introduced the CICERO[17] AI agent. It is the first AI that attained human-level performance in the popular strategy game Diplomacy[18].

With the expansion of simulation to multiple agents while focusing on collaboration, Park et al. [79] introduced the Generative Agents in the virtual environment framework with the simulation of twenty-five handcrafted agent characters. They are pointing out the fact that believable agents require conditioning not only in their current environment but also in a vast amount of past experience. This is impossible nowadays due to the underlying models' limited context window when first-order prompting is used. Based on that, they employed a hybrid memory structure to facilitate agent behaviors. This memory structure explicitly models the human short-term and long-term memories. Short-term memory temporarily buffers recent perceptions, while long-term memory consolidates important information over time [101]. In this case, long-term memory is designed as a shared *memory stream*. The agent's memories are dynamically retrieved based on three factors: *recency* using the time of memory access, *importance* to the agent's beliefs, and *relevance* to the current situation. The top-ranked memories are fitted into the context window of the LLM. With that, they use the concepts of agents' *self-reflection* and implement other cognitive modules such as perception, planning, and action. Figure 3.1 shows an example of their game environment.

In relation to [79], with a focus on human behavior, METAAGENTS by Li et al. [61] simulated the job fair environment with predefined character positions and responsibilities. Unlike [79], they do not recommend the proposed memory retrieval for conversation-heavy

[15] https://www.ea.com/games/the-sims, accessed: 2024-05-02.
[16] https://www.minecraft.net, accessed: 2024-11-25.
[17] https://ai.meta.com/research/cicero/, accessed: 2025-05-09.
[18] https://webdiplomacy.net/, accessed: 2025-05-09.

Figure 3.1: The Generative Agents game environment example throughout the day. Adapted from [79].

settings. Instead of directly inserting extended multi-turn dialogues into the prompt window and to avoid the risks of error introduction by using the text summarization, they replaced this concept by extracting the overarching theme and context of the conversation, and aligning with the key terms and standout words.

With an extension of [79], Wang et al. [107] introduced Humanoid Agents by using the game interface as well. They aim to enhance the realism and applicability of generative agents. Humanoid agents have a single emotion at the time out of seven possible options: *anger*, *sadness*, *fear*, *surprise*, *happiness*, *neutral*, and *disgust*, following Ekman's six basic emotions [27]. Also, each of their basic needs (*fullness*, *fun*, *health*, *social*, and *energy*) is set in an integer range, and the relationships between agents are rated in the proper range as well. Recently, Qian et al. [81] used the proposed ChatDev virtual environment as a software company with agents specialized in roles. In this chat-powered development framework, they used typical job positions such as *Chief Executive Officer* (CEO), *Chief Technology Officer* (CTO), *programmer*, *reviewer*, and *tester*. With these agents, they use their interaction to develop software based on the provided specifications. The agents also use the hybrid memory structure (short-term and long-term memory).

3.3.2 Artificial Agents Interaction

In this section, I describe a simulation of LLMs' interactions using artificial large language model agents, in general. Such environments targeting the communication task are also known as *Multi-Agent Debate* (MAD).

Especially in this section, the authors of such papers modeled agents whose primary task is not necessarily the imitation of real human interaction but instead the interaction of primarily manually created artificial characters. Typically, such characters have some position and resulting duties, while we are not strict about the agent's character traits.

Using the interactions between two LLMs only, Abbasiantaeb et al. [1] simulated human-to-human conversational question-answering by using two LLM agents — *teacher* and *student*. The teacher is connected to Wikipedia texts, and to maintain the most accurate answer, the teacher's LLM agent can only use copied texts from this website.

Alongside scaling to several interacting entities (more than two), there are generally two approaches to simulate multiple persons or artificial agents. The first one uses only a single LLM while modeling and simulating multiple entities. The second model uses multiple LLMs, while a single LLM at least represents each so-called agent. By simulating strategic human behavior, Sreedhar and Chilton [89], using the single-agent and multiple-agent architecture, achieved better results with the second option.

In the context of MAD, it was first introduced by Du et al. [26]. They studied improvements in factuality and reasoning in LLMs using multiple so-called *rounds*. Their findings suggest that the *society of minds* approach [73] has the potential to advance the capabilities of LLMs significantly. In this approach, Minsky argues that the human mind is not a singular entity but rather a society of simpler components working together. These components collectively produce what we perceive as complex behaviors, intelligence, and consciousness.

With a focus on task solving, Chan et al. [13] presented the multi-agent referee team — ChatEval. This team discusses the specified problem and proposes a final decision and solution. Using the diverse agents' roles, the agents engage in sequential debates with access to all communication history. Using various communication strategies, the agents interact in multiple rounds to achieve the best results.

Through a multi-agent debate framework which I denote as „MAD with Judge," Liang et al. [63] figured out the Degeneration-of-Thought problem, which claims that once the LLM has established confidence in its solutions, it is unable to generate novel thoughts later through reflection even if its initial stance is incorrect. They also incorporate the judge agent, which monitors and manages the debate process to obtain a final solution. With an MAD architecture design, Chen et al. [16] created a ReConcile framework. In a multi-round, multi-agent debate, they improved collaborative reasoning results on multiple benchmarks using different LLMs for the debate agents.

From a general point of view, Wu et al. [109] introduced an open-source framework for creating multi-agent conversations with LLMs called AutoGen. By this, they provide the necessary basic structures for the developer's needs. The extension by Dibia et al. [24] also provides the user interface for such a framework in the AutoGen Studio application. The other open frameworks available are also CrewAI[19] or LangGraph Studio[20].

In the context of Microsoft Research[21] publications, Fourney et al. [33] proposed a complex, advanced multi-agent system, called Magentic-One, with agents designed for specific tasks. For example, one agent is used for accessing images and extracting code, another single agent is used for navigating to a URL, one is used for analyzing the code, and the last is used for code execution. In summary, each agent is responsible for a concrete subset of tasks under the guidance of one leading agent. This suggests the advantage of different specializations of the individual participants in the interaction and the setting of hierarchies in the group, both of which are discussed in more detail in the following section.

[19] `https://www.crewai.com/`, accessed: 2025-05-01.
[20] `https://github.com/langchain-ai/langgraph-studio`, accessed: 2025-05-01.
[21] `https://www.microsoft.com/en-us/research/`, accessed: 2025-05-01.

Lastly, with a focus on agents' definition, instead of relying on predefined agents, Chen et al. [15] proposed the AutoAgents framework for automatic specialized agents generation according to the task definition.

3.3.3 Human Interaction Modeling

Instead of primarily targeting the simulation task itself, we now move to realistic human behavior simulations using LLM agents. The primary focus is to model a human characteristic as believably as possible.

Tang et al. [94] tackle the problem of a lack of human characteristics in LLM simulations by designing the agent-based platform GenSim. This platform simulates customizable social scenarios, while it supports one hundred thousand agents to simulate large-scale populations in real-world contexts. Error-correction mechanisms are incorporated to ensure more reliable and long-term simulations. They used a hybrid memory structure (short-term and long-term), and the reflection mechanism is included. Furthermore, Zhou et al. [118] created a SOTOPIA environment to simulate and evaluate the complex social interaction between artificial agents using the forty predefined characters, where a single LLM represents each agent. In the environment, agents role-play and interact in various scenarios to achieve social goals. With the social network simulation system, Gao et al. [35] modeled primarily the processes of information, attitude, and emotion. They used a different approach for the agent's memory architecture, called the *memory pool*. This pool is a dynamic store that functions as an agent's long-term memory. In [60], Li et al. present a novel framework, called CAMEL, aimed at enhancing autonomous cooperation among LLMs. Instead of requiring human input for effective task completion, the authors propose a system in which multiple AI agents interact and collaborate with minimal human intervention.

In connection with [118], Zhou et al. [117] introduced an evaluation of the two main approaches to simulate social interactions, namely using a single LLM to simulate all agents or using a language model for each represented agent. They highlight the performance of the single LLM over the multiple ones based on their struggles with meeting the specification of the social goals and producing a less natural social interaction flow compared to the first option. On the other hand, they additionally recommend preferring the multi-agent system over the single LLM system when sociological realism is a priority. With that, Wang et al. [104] discovered that a well-prompted single-agent LLM can achieve performance nearly equivalent to that of the best multi-agent discussion frameworks across various reasoning tasks. Specifically, when task-specific demonstrations are included in the prompt, the performance of a single-agent LLM matches or even surpasses that of multi-agent systems. However, multi-agent discussions tend to outperform single-agent setups in scenarios where no demonstrations are provided in the prompt.

With a focus on the inter-consistency of LLM collaboration, Xiong et al. [111] formulated a three-stage debate by simulating real-world scenarios: *fair debate*, *mismatched debate*, and *roundtable debate*. Using only two LLMs, they modeled the fair debate (LLMs with comparable capabilities) and the mismatched debate (LLMs that exhibit vastly different levels of abilities). The roundtable debate scenario was added by including more models in the interaction. The primary conclusion is that stronger LLMs can be distracted by weaker language models when debates are mismatched.

Multiple approaches were invented in the context of a workflow in team collaboration. Simultaneously, with a base agent's interaction, some inventions can be used to design the agent's specialized expertise. Hong et al. [51] in the MetaGPT framework used during the programming team human workflows, the standard developer team positions, such as *product manager*, *architect*, or *engineer*, for example. With that, long-term memory is only designed to store summarized information.

Moreover, it is possible to specify a single team leader in team cooperation. Using this approach, in [33], although this publication is not focused on human simulation realism, the authors select a single leading model called *orchestrator*. Similarly, with a focus on social interactions, Liu et al. [65] used the central agent in a grid of agents called the *learner agent*, which collects answers to the initiated question from the connected active interacting agents. Focusing on social interactions, they also proposed a training paradigm that allows language models to learn from simulated social interactions while achieving social alignment. Compared to existing methodologies, the new approach achieved better performance in alignment benchmarks and human evaluations.

By providing complex taxonomies and overviews, Gürcan [43], using a general overview, introduced the four-group methodology: *agent-oriented*, *interaction-oriented*, *environment-oriented*, and *organizational-oriented*. By the proposition, social agents are role-players playing one or several predefined characters. Gao et al. [34] presented a survey on the topic of agent-based simulations. They provide a mindful overview with a focus on the agent's *capability*, *autonomy*, *social ability*, *reactivity*, and *proactiveness*, as well as the *interactions* themselves. The significant disadvantage found is long-term planning in complex real-world problems, for which current agent architectures are not fully capable of solving these challenges, such as processing speed, resource efficiency, or task complexity. Another disadvantage is that the agent cannot be transferred to other environments. Although an agent may excel in the environment for which it was designed, its performance may be completely inadequate in other environments. Lastly, they pointed out the difficulty of the quantitative and qualitative evaluation task and marked it as an open problem and a future research direction. Guo et al. [42] summarized the current state-of-the-art by dividing the communication paradigms into three fields: *cooperative*, *debate*, and *competitive*. Additionally, they split the communication structure into *layered*, *decentralized*, *centralized*, and *shared message pools*. Wang et al. [101] provide an exhaustive range survey proposing a typical general architecture design of a unified framework containing the *profiling module*, *memory module*, *planning module*, and *action module*. With that, the metrics used are well summarized.

3.3.4 Evaluation of Communication using Large Language Models

One of the most challenging tasks in this field is approaching the evaluation step. In general, we can use two major approaches: (1) an automated benchmark, which can involve single or multiple LLMs, and (2) human crowd workers who evaluate the outputs of the tested system.

With a focus on the evaluation issue of the believability of human behavior, Aher et al. [3] proposed a new type of test to evaluate to what extent a given language model can simulate different aspects of human behavior — the Turing Experiment. For the same task, Xiao et al. [110] created a benchmark called SimulateBench. This benchmark is based on the *consistency* dimension — the extent to which LLMs can behave consistently with the given information of a human to simulate, and on the *robustness* dimension — the ability of LLMs to simulate behaviors that remain robust when faced with perturbations. The authors figured out that the GPT series models[22] perform better than the open-source models,

[22] https://platform.openai.com/docs/models/gp, accessed: 2024-11-29.

and that a longer context size does not necessarily mean better consistency performance. In [111], Xiong et al. define the metric to quantify the inter-inconsistency among multiple LLMs. Lastly, Khan et al. [55] highlighted the fact that LLMs are experts in various fields, while humans are not. So, evaluating the models by humans will evolve into non-experts overseeing experts.

The authors of the individual frameworks also use multiple evaluation approaches. Du et al. [26] conducted experiments on various tasks from multiple fields, such as arithmetic and mathematical reasoning tasks and chess move prediction. In addition, for the actuality evaluation, they use the task to generate historical biographies of people, actuality knowledge questions, and the validity of the chess move. Similarly, Wang et al. [106] used multiple tasks, including the new (Trivia Creative Writing) and took over (Codenames Collaborative and Logic Grid Puzzle, both taken from [90]). Park et al. [79] evaluated generative agents with two approaches — *controlled evaluation* and *end-to-end evaluation*. They run a controlled evaluation in two stages. First, they separately evaluated individual agents to determine whether they generate believable behavior in narrowly defined contexts. Then, they performed an agent community evaluation in the environment and investigated their emergent behavior as a collective. The results were scored by 100 human participants. In the end-to-end evaluation part, they measured emergent social behaviors: *information diffusion*, *relationship formation*, and *agent coordination*. With that, they conducted an analysis of boundaries and errors as well. To investigate the effectiveness of Humanoid Agents, Wang et al. [107] by using the effects of activities and conversations compared their predictions with human annotations made by three volunteers.

In [17], Chen et al. evaluated the framework on seven benchmarks, including two commonsense (StrateqyQA [37] and CommonsenseQA [2]), three math (GSM8K [20], AQuA [64], and MATH [49]), one logical reasoning (Date Understanding [90]), and one natural language inference task (ANLI [76]). Similarly, Wang et al. [104] used the FOLIO-wiki dataset [46, 116] with GSM8K [20] and ECQA dataset [2]. With the design of a new holistic evaluation framework called SOTOPIA-EVAL, Zhou et al. [118] used the following dimensions in the adequate possible numeric range: *goal completion*, *believability*, *knowledge*, *secret*, *relationship*, *social rules*, and *financial and material benefits*.

In the context of using LLMs as evaluators, Wang et al. [102] discussed the fairness of the evaluation responses of two different models. They found that the quality ranking of candidate responses can easily be hacked by simply altering their order of appearance in the context. This manipulation allows for skewing the evaluation result, making one model appear considerably superior to the other. With that, they propose multiple strategies to address this issue in the FairEval[23] tool. While incorporating multiple LLMs to evaluate open-domain chatbots through MAD, Chan et al. [13] constructed a referee team called ChatEval. This team autonomously discusses and evaluates the quality of responses generated from different models. It focuses on open-ended questions and natural language generation tasks.

From a different perspective, Chen et al. [14] created a literature review to identify the primary challenges in the evaluation of LLM agents. They summarized the evaluation metrics, falling into one of two groups — *Agent-oriented* or *Task-oriented*, and provided a comprehensive taxonomy. Also, Gao et al. [34] developed a survey with part oriented on the evaluation process with the split on *realness validation with real human data*, *explanations providing for simulated behaviors*, and *ethics evaluation* and Wang et al. [101]

[23] https://github.com/i-Eval/FairEval, accessed: 2025-04-30.

identified the *subjective evaluation* (human annotation and the Turing test [96]) and *objective evaluation* (environment simulation, social evaluation, multi-task evaluation and software design).

3.3.5 Communication Patterns

From the point of view of communication patterns, in general, we can classify the published literature using four main concepts: *pair dialogue*, *one-by-one*, *simultaneous talk*, and *external selector*.

Pair Dialogue

In this approach, only two LLM agents speak with each other [1, 55, 60, 74, 79, 81, 107, 118]. More precisely, the setup involves one debate initiator and a second agent, with the two taking turns speaking to each other.

One-by-One Pattern

By adding more than two agents into the simulation, the previous approach is not applicable. This pattern iterates over all agents in a round-robin circle in a fixed order to ensure all agents speak equally [3, 13, 26, 51, 63].

Simultaneous Talk using the Multi-Agent Debate

The simultaneous talk improves the predefined order from the previous pattern design by letting the agents generate the answer all at once [13, 14, 62, 94, 104]. After each generation, agents have access to the others' responses based on the chosen communication topology [62].

External Selector Approach

Following [109], the AutoGen library[24] offers a unique approach to previous approaches of selecting the agents' speaking order. They include a single external agent that selects the next speaker after each message.

Interaction Rounds and Consensus

In the context of achieving the best performance possible, some of the authors proposed the communication pattern, where agents interact in multiple rounds [13, 94] mostly to achieve the final consensus [17, 26, 62, 63, 65, 104]. After each round, the primary goal is to reevaluate the agent's attitudes to achieve consensus.

The approach in which consensus is achieved differs. Generally, the debate can be seen as a multi-agent game, where convergence is not guaranteed [26]. Du et al. [26] found that language models are able to converge on a single shared answer after multiple rounds of debate, so all agents have to agree on a final solution. A similar approach is used by Chen et al. [17] except the number of rounds is predefined, so there does not have to be a consensus every time. Liang et al. [63] used different debate levels, from full consensus, via leveraging the amount of agreement of participants (the majority agents rule is applied), to the absolute impossibility of consensus. In addition, Li et al. [62] and Wang et al. [104]

[24] `https://github.com/microsoft/autogen`, accessed: 2025-04-30.

used a majority vote to reach a consensus, while in [104], the authors let an additional agent choose the final decision by analyzing the voting system history. Figure 3.2 visualizes the voting system approach. Finally, Liu et al. [65] approached consensus replacement by maximizing the reward associated with desired or socially acceptable outcomes.

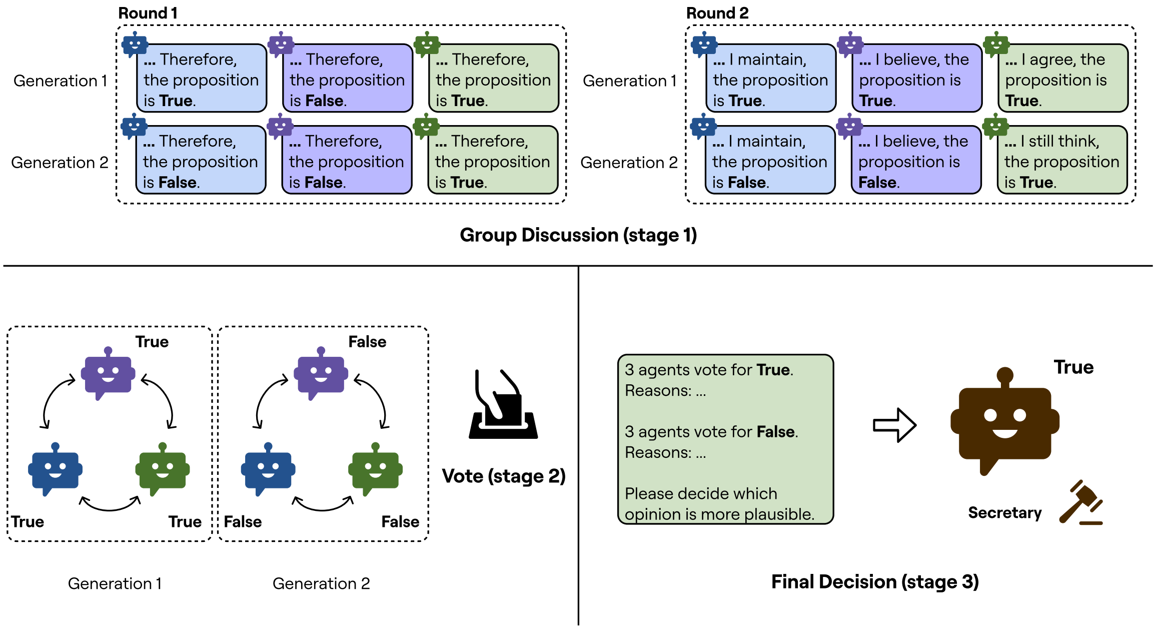

Figure 3.2: The visualization of the voting system. First, all the agents communicate in multiple rounds in the first stage. Afterward, they vote during the second stage, and a non-involved judge makes the final decision in the last stage of this process.

3.4 Evaluation of Approaches and Memory Design Choices

The core of each agent is its memory. Overall, two types of memory are mainly used in the proposed systems: short-term memory (representing working memory) and long-term memory with lifelong memory storage capabilities. Multiple multi-agent systems propose such a memory architecture [61, 79, 81, 94].

Notably, Li et al. [61] noted the disadvantages of the retrieval function used by Park et al. [79] when the scoring mechanism ranks memories based on their recency, relevance, and importance. In relation to [61], the authors mention the problems with this architecture in conversation-heavy settings. Furthermore, they highlight the risks of a summary of the original text. With that, they propose replacing these criteria with a memory retrieval function extracting two information categories, the overarching theme with the context of the conversation, and key terms or standout words. The authors mentioned that this mechanism mimics human-like recall processes.

I go beyond the memory retrieval function using the proposed techniques in this work. To more faithfully mimic the human brain and behavior, as explained in [25, 29, 31, 97], the authors highlight the storage of information based on energy. As the human body and brain are largely influenced at the emotional level, the proportionality of remembering information or an event depends on the strength of the emotional experience associated with the situation or the knowledge. The more positive or negative the thought or event is, the more probable it is that such an experience will be remembered. The same goes for memory retrieval based on the current persona's emotional state.

To save information from short-term memory for long-term knowledge, as in the human brain, memories are moved from short-term memory into long-term memory. Moving memories from short-term memory into long-term memory is also known as a *memory consolidation* process [85].

At the same time, I propose another technique to improve the quality of imitating a real-life persona. As figured out in [3], LLMs can fail in imitating non-expert entities, such as humans. More interestingly, the models struggle with simulating the children, for example, when the pressure is that the children's knowledge is not as advanced as an adult's (possibly with some higher education degree). More concretely, the five-year-old girl very likely should not know anything about a complex topic such as, for example, the theory of relativity[25], at least not so deeply and mathematically.

In compliance with the given criteria, the proposed design includes various perspectives. Furthermore, I combine the related methods and introduce new improvement techniques.

[25] https://www.bibsonomy.org/bibtex/291aa35fba096740bb413835ef651bdfb/ad4, accessed: 2025-04-30.

Chapter 4

Environment Design and Configuration

In this chapter, I describe the system design for the new proposed simulation framework called *PerSimChat*, including agent architecture, environment settings, and simulation scenarios.

Following the recommendations from [117], the authors recommend the multi-agent system for scenarios with the emphasis on social realism over the benefits of the single model simulation. Based on that, the system designed in this work also uses the multi-agent architecture.

Further, I describe the process of choosing an architecture for large language model (LLM) agents. Subsequently, I discuss the agent's design and all its cognitive and general modules. Having an agent entity, I go beyond the environment configurations, including communication patterns, reaching consensus, and finally, the higher view of the simulation scenarios and tasks.

4.1 Agent's Design and Character Modeling

This section discusses the structure and design of the individual agents in the simulation system. With a focus on a single LLM agent, each entity contains all relevant modules. The primary ones are the blocks responsible for the cognitive processes of the simulated brain. Naturally, these modules are referred to as *cognitive modules*. Alongside these modules, another crucial module designed for the agent's emotions is the *profile module*. Cognitive modules primarily provide for the profile module the retrieved memories from the agent's memory system, while the profile module generates and provides the current emotional level of the agent. The high-level agent's architecture represented by these two modules is shown in Figure 4.1. Subsequently, the individual modules are analyzed in detail, including their design, functionalities, and purposes.

4.1.1 Cognitive Modules

Based on the systematic literature review, including state-of-the-art techniques, I conclude that Park et al. [79] provided valuable insights into the overall artificial agent structure in the Generative Agents. From this point of view, including the similarities with a simulation of human conversation, I follow the cognitive function modules they used by adding the *natural language module*. This work proposes the following modules: *perception*, *memory*

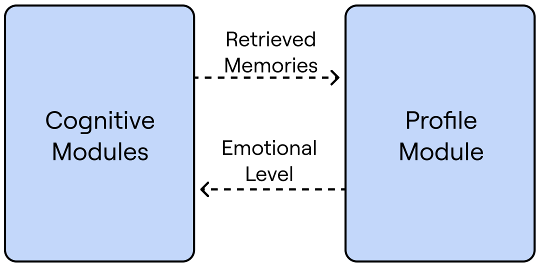

Figure 4.1: The high-level single agent architecture is built from two major modules — the cognitive modules responsible for the cognitive functions processing and the profile module responsible for the dynamic emotional level generation. The output of the cognitive modules is the memories retrieved from short-term and long-term memory. The profile module's output is the generated emotional level.

(including *retrieval function*), *reflection*, *action*, *planning*, and *natural language*. These modules correspond to the human cognitive functions described in Section 3.1.1. The overall architecture using the cognitive modules and the relations between them is shown in Figure 4.2.

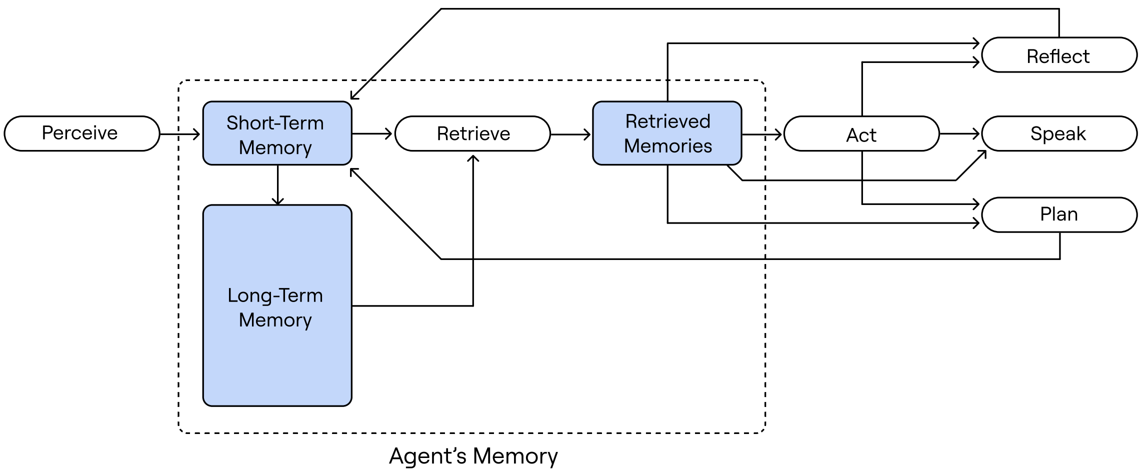

Figure 4.2: The architecture of cognitive functions modules in the system. The main part is the agent's memory, which is primarily represented by short-term and long-term memory. The *perceive*, *act*, *reflect*, *speak*, and *plan* actions represent the individual modules in order, namely the perception module, the action module, the reflection module, the natural language module, and the planning module.

Based on the survey introduced by Wang et al. [101], I can classify this thesis into the following categories:

- Memory Module:
  - Memory Structure: Hybrid Memory.
  - Memory Format: Natural Language.

  - Memory Operation:
    * Reading.
    * Writing.
    * Reflection.

- Profile Module:
  - Profile Contents:
    * Demographic Information.
    * Personality Information.
    * Social Information.
  - Generation Strategy:
    * LLM-Generation.
    * Dataset Alignment.

- Planning Module: Planning with Feedback using Model Feedback.

- Action Module:
  - Action Target: Communication.
  - Action Production: Memory Recollection.
  - Action Space: Self-Knowledge.
  - Action Impact: Internal States.

In the following section, I will further describe, along with other modules, each of the above.

Memory Module

As retrieved from [34], almost every piece of literature published on the related topic uses the memory module. As described in Section 3.1.1, the human brain has three types of memory: sensory memory, short-term memory, and long-term memory. Based on the length of information storage in such memory, I conclude that sensory memory is not needed for the task, based on the fact that the primary purpose of such memory is to retain impressions of sensory information, which is stored in such memory for 200–500 milliseconds. This memory is unusable for the simulation of communication.

Overall, I use two types of memory: short-term memory and long-term memory. This architecture is also used and recommended by previous publications, which are described in more detail in Section 3.4. Figure 4.3 shows the general memory architecture. For memory processes, the central working part, the short-term memory, needs one of the following information to store: *last speech topic and keywords*, the individual persona's *self-reflection*, or the persona's *plan*, also provided in the same format. The agent's memory module output is the retrieved memories from short-term and long-term memory.

Memory Writing Inspired by the approach of [61], instead of saving memory for the entire conversation, the designed system saves two types of information from the original discussion:

- The overarching theme and context of the conversation.

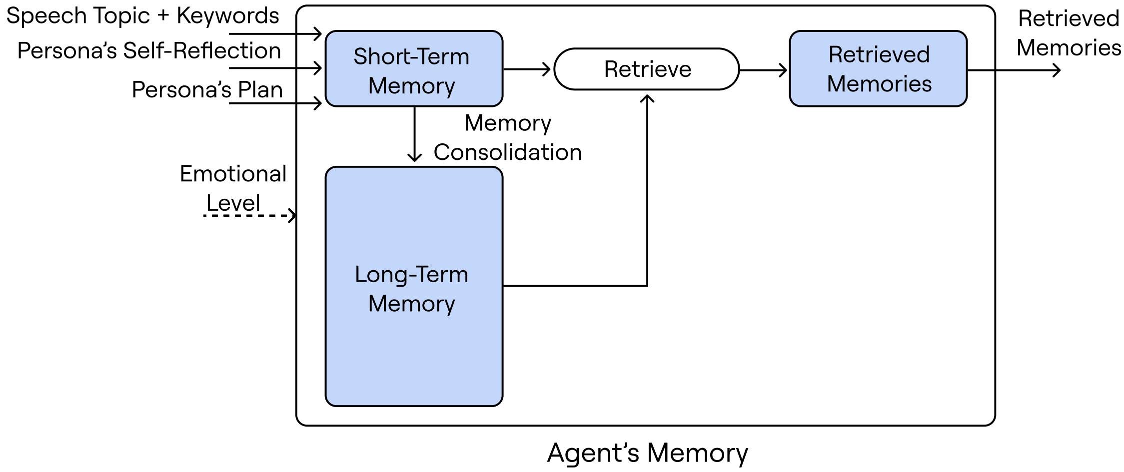

Figure 4.3: The agent's memory architecture. It comprises short-term memory, long-term memory, and temporal memory of the retrieved memories (denoted as *retrieved memories*). The output of the whole memory is this information. As input, the short-term memory saves one of the following information: the last speech topic and keywords, the individual persona's self-reflection, or the plan (in the same format). The dynamically updated emotional level is used for decision-making in the memory processes.

- Key terms or standout words.

These observed experiences are first stored in short-term memory. Such an approach more likely mimics the human brain function when the stored memories are simplified and changed compared to the original perception.

In summary, the three types of information can be stored in short-term memory in such a format: last available speech, the individual agent's self-reflection, or its plan.

When new information is available to store in short-term memory, it replaces the old information. In this step, the *memory consolidation* process is performed, which is also responsible for the information filtering. It decides which data is moved into the long-term memory and which is discarded.

In the designed system, whether the memory is stored in long-term memory or not depends on the following factors:

- Emotional level of the agent based on memories in long-term memory (including persona description, characteristics, and traits, which are already stored in long-term memory as the first item), as well as memories from short-term memory. This includes the context of the influence of a specific text on an agent. The emotional state is simultaneously saved in the long-term memory with the stored information.

- Relevance (event, task relevance, and semantic context) [79] using the similarity of the new information (memory chunk stored in the short-term memory) to the memories already stored in the long-term memory.

- Importance similarly to [79], which is stored in the long-term memory as well.

This represents one filter for storing information in long-term memory. The second filter is executed when recalling the information from memory.

Memory Reading Based on [79], for the retrieval function, I use multiple components to calculate the final retrieval scores for the memories.

When information is retrieved from long-term memory, the score for the memory to be retrieved is calculated using the following factors:

- Recency and frequency [79].
- Importance similarly to [79].
- Relevance (event, task relevance, and semantic context) [79].
- Similarity with the information in the short-term memory.
- Correlation of the emotional level of the agent when storing the memory chunk in the long-term memory compared to the current agent's emotional level.

Information from short-term memory is obtained directly without further modifications.

Reflection Module

Similarly to [79], I consider a type of memory relating to standard observations — the *reflection*. This type of memory stores the inner agent's thoughts and is stored in long-term memory simultaneously with plans and memories from short-term memory. Therefore, it is also included when retrieving memories.

Because the new design uses the agent's emotional state, I let the LLM decide, based on the current emotional state and previous memories, whether to reflect (the planning is performed otherwise). This reflects the inclusion of the persona's mood to perform self-reflection.

Because reflections are also stored in long-term memory alongside perceptions and plans, the self-reflection process builds a hierarchical tree of reflections, in a similar vein to [79]. Using this approach, the leaf nodes represent the original observations. Every higher level of a tree with non-leaf nodes generalizes the observed information in a more abstract and higher-level structure. Figure 4.4 shows an example visualization of such a tree.

The reflection topic and keywords also represent the information. The architecture of the reflection module is shown in Figure 4.5. This module needs the following information for its inner processing: retrieved memories from the agent's memory, information on whether to reflect (a flag), and the actual agent's emotional level.

Planning Module

For the individual agent, we need it to follow a consistent long-term behavior that does not make significant changes to reflect a particular behavior of a real individual. This contributes to long-term planning over the time horizon to ensure a coherent sequence of actions. In real conversation, even if someone does not speak, they are still thinking, planning, or reacting internally. Maintaining continuity in an individual agent's processing helps improve agents' responses in future turns.

The plan is generated using a single LLM. The information needed for the generation process is the retrieved memories, information on whether to plan (a flag), and the actual agent's emotional level. The planning process results in the plan itself represented as a plan topic and keywords. The architecture of this module is shown in Figure 4.6.

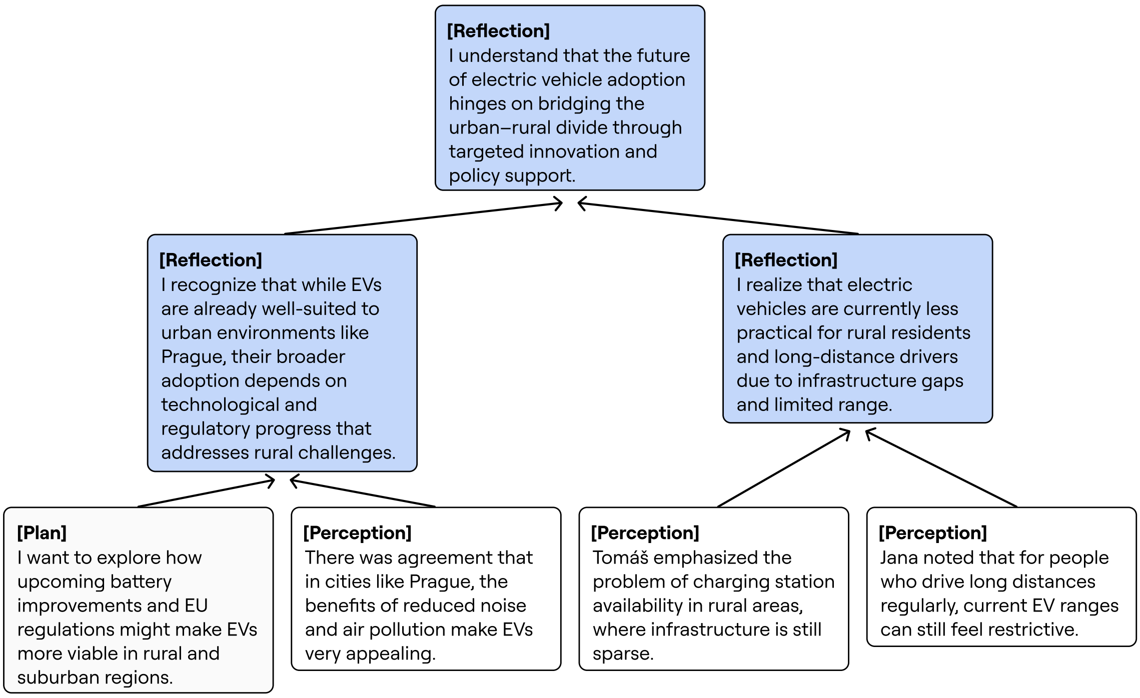

Figure 4.4: The example of an agent's reflection tree, including the perceptions of the environment and plans, represented as the leaf nodes.

The resulting plan is stored in short-term memory. Based on that, the system has a history of the agent's plans later stored in the long-term memory. In addition, planning, alongside the stored observations and reflections, forms a system for decision-making regarding the agent's behavior. Despite changes in the memory module, this acquisition reflects the design in [79]. Including the planning provides the agent with the ability to react to external observations.

Action Module

The action module selects the action to execute. The action is determined based on the relevant memories that have been retrieved. These memories are obtained from short-term and long-term memory. The action module also needs the agent's actual emotional level. The result of this module is the flag, which determines whether to reflect or plan. It also indirectly affects the natural language module (speak) because of the analysis of the agent's *need to talk*. During a single message generation in the discussion, each agent participant performs only a single action (one of reflection, planning, or speaking). In the overall system, the selection of which agent will speak is made based on the agent's score. This value is reassessed before each talk. All of these observations are generated using the LLM. Figure 4.7 shows the overall architecture of the action module.

The main goal of this module is to analyze the agent's memories and decide whether to speak based on the agent's need to talk. Overall, the real actions of the current agent are affected by the other agent's actions (need to talk levels), provided at the same time. For generating such a level, a single LLM is prompted with dynamically retrieved memories

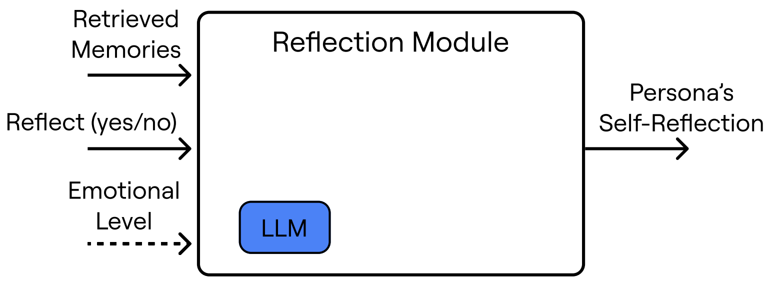

Figure 4.5: The reflection module architecture is built using a large language model (LLM) for the reflection process. For such a process, the module needs the retrieved memories from short-term and long-term memory, information on whether to reflect, and the actual emotional level. The output of this module is the individual personas' self-reflection.

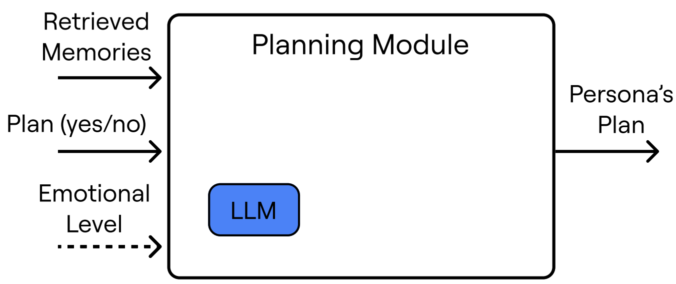

Figure 4.6: The architecture of the planning module. This module is built primarily from the large language model (LLM) used for the plan generation. Such a process needs the retrieved memories, information on whether to plan, and the actual agent's emotional level. The result of this module is the long-term plan itself.

and the current emotional state. Based on this, the model is required to generate a single integer number in the range 0–10 as output.

Also, I need to capture the agent's behavioral goal, such as whether, in the dialogue, it should either be open to final consensus or oppose the proposals. The openness to consensus value generation is also the primary responsibility of this module.

The action module sub-parts do not directly affect the simulation. However, the values generated for a single persona indirectly affect how the simulation progresses.

Perception Module

The perception module is responsible for perceiving information from the external world. It captures the current observations from the multiple agents' discussions and provides them to the short-term memory module. The perception module is also responsible for extracting the overarching text theme with keywords and standout words using LLM before passing this information to the short-term memory module. This task needs the last observed speech (of the other agents or the agent itself) alongside the additional metadata, such as the speech author and the emotional level of the speech author. In addition, the emotional level of the persona itself is needed. The architecture of the perception module is shown in Figure 4.8.

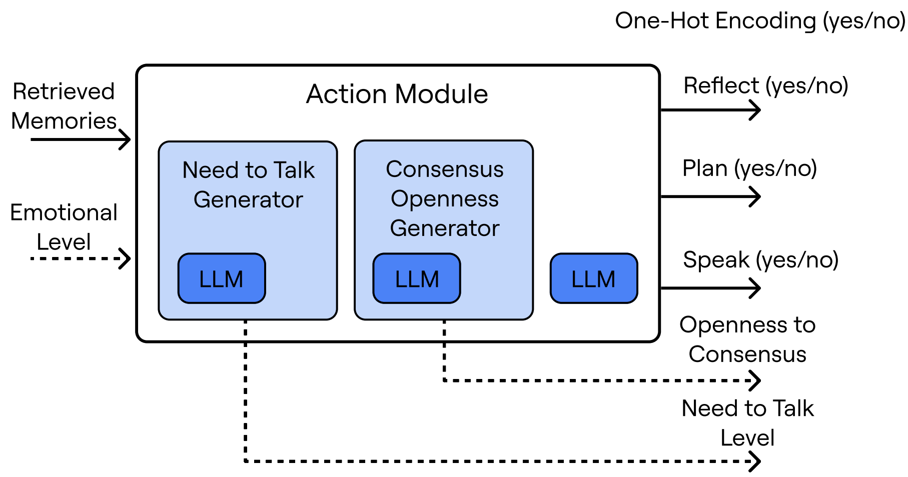

Figure 4.7: The architecture of the action module. This module is built from three main parts. One part is a large language model (LLM), which is responsible for choosing the action to perform — reflection or planning. The natural language module (speak) is indirectly affected by the need for talk level generation; however, each agent performs only a single action in a single message generation. The need to talk generator submodule is responsible for such a calculation. The consensus openness generator submodule generates the value of openness to achieving final consensus in a group debate. The action module needs the retrieved memories and the actual emotional level for such processing.

I want to mention that the perception module is the last module, which handles the original text before it is processed to extract the topic and keywords.

Natural Language Module

This module is responsible for the actual talk (text message) production. The style of a text and the opinions contained therein are affected by all other modules. In summary, the information provided and the style of the text relate to the agent's memories, including the relations with other agents, the agent's social goals, the agent's actual emotional state, the self-reflection results, planning and reacting to decisions, and the actions proposed. This module uses a single LLM for message generation. The result of this module is the generated talk, along with additional metadata. The architecture of the natural language module is shown in Figure 4.9.

To achieve a faster communication sequence, I limit the agents' generated message to one paragraph of text, which I consider about 50 words. However, this value is not set directly. Instead, if possible, the agent is reminded to provide shorter and more concise answers. However, the system does not prevent the agent from speaking more thoroughly and accurately.

4.1.2 Profile Module

To ensure that the persona behaves more accurately, naturally, and is more consistent during the interaction, the persona's *profile module* was added along with the cognitive modules. The dynamical part of this module during the conversation generates the emotional level of the agent.

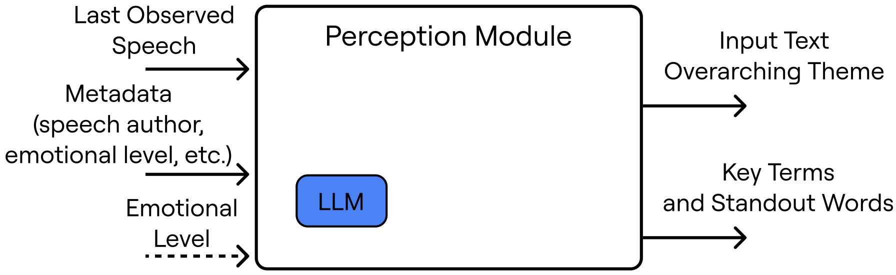

Figure 4.8: The architecture of the perception module. The main part of this module is the large language model (LLM) for the overarching theme, key terms, and standout words extraction from the input text. In addition to the last observed speech text, the module needs the metadata, such as the speech author and the author's emotional state, when generating this text. Also, the module needs the current persona's emotional level.

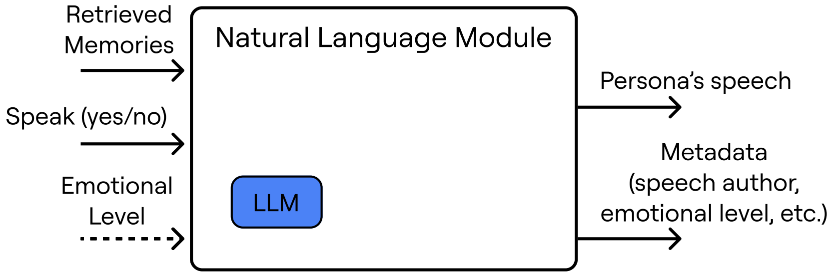

Figure 4.9: The architecture of the natural language module. The main part of this module is the large language model (LLM) for text generation. For this task, the module needs the retrieved memories, information on whether to speak, and the actual agent's emotional level. The result is the generated speech itself alongside additional metadata.

The emotional state is represented by the concept of Ekman's six basic emotions [27], and so *happiness*, *sadness*, *anger*, *fear*, *disgust*, and *surprise*. According to [107], instead of using just one of these levels at a time to obtain a much richer context of the agent's emotional state, the LLM generates a score in the range 0–10 for each of these emotions. The important part of an LLM prompt, which is responsible for the generation of actual value, is provided in Listing 4.1. For that, I work with the hypothesis that the LLM can sufficiently determine the combinations of emotions that can and cannot realistically occur. The part of the profile module responsible for the emotional level generation is the *emotional level generator*, which is represented by the described LLM, as shown in Figure 4.10.

```
Based on this information, how does the agent's emotional state evolve? Adjust
   the intensity of each emotion accordingly to simulate a natural emotional
   response. Consider:
• User's personality traits (e.g., extraverted users may cause higher levels of
   happiness or excitement in certain scenarios, while neurotic users may trigger
   higher fear or sadness).
• The current event (whether it is positive, neutral, or negative).
• The relationship between the agent and the user (e.g., friendly, neutral, or
   antagonistic).
```

```
• Previous emotional history (what emotions have been dominant in the past and
  how they influence the current state).
• Any other contextual factors from memory (how past interactions with the user
  or previous emotional states play into the new response).
```

Listing 4.1: The part of the emotional level generation prompt responsible for navigating the language model to the decision-making process of the emotional level generator. The language model ensures the core functionality of this generator.

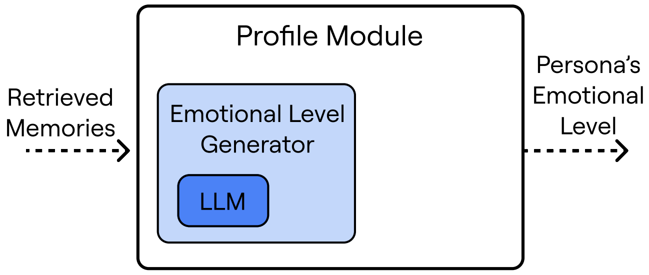

Figure 4.10: The architecture of the profile module. This module dynamically updates the actual persona's emotional level. Such a task uses the emotional level generator represented by a single large language model (LLM) with the retrieved memories provided by the cognitive modules and the previous emotional level, if available.

For emotional level generation, I use a single LLM. It is needed to process the retrieved memories provided by the cognitive modules. Every time a new message is added to the list of messages, the emotional state is re-evaluated, and the latest context from the conversation is included in the calculation. Because the memories used for generation are dynamically retrieved based on several factors, the original personal information should mostly remain.

During the generation process of the new agent's emotional level, the system also incorporates the previous emotional state. At the beginning of the conversation, the emotional level of the persona is only generated based on the persona description and the conversation topic. The result is the dynamic persona's emotional level.

4.1.3 Character Modeling with Persona Data

I use believable information about simulated personas to create agents that more closely resemble real people. In this work, a persona refers to a synthetic character with predefined demographic, behavioral, and emotional attributes, used for simulation purposes. This information contains demographic, personality, and social information about the persona. To obtain a broader context of persona descriptions, I used both the artificially generated personas and the dataset personas provided by the Lakmoos AI s.r.o[1], which are transformed into the required format. In addition, the amount of persona description is increased using LLM-generated content. The proprietary data does not use real persons. In general, the dataset is created from statistics about real populations and available surveys. However, the corresponding entities are virtual.

[1] https://lakmoos.com/, accessed: 2024-11-30.

Using the data in the appropriate format, I let the LLM generate the styling of the persona, which contains the persona's pseudo-name and surname, the persona styling description, one paragraph long, and the persona's main characteristics and traits. This information is generated before the simulation run, and the profiles are provided in this format. At the beginning of the simulation run, the system stores them in long-term memory as a first record. The example of a persona profile description is shown in Listing 4.2.

```
Name and Surname: Josef Svoboda
Description: Josef Svoboda is a down-to-earth, 50-year-old family man who enjoys
   the simple pleasures of life in a small town in the Czech Republic, where he
   shares a cozy home with his wife and two children. A hardworking man with a
   high school diploma, he values tradition, loyalty, and good company. His
   weekdays are spent balancing work and family, while weekends bring the thrill
   of a football match with friends or the quiet patience of fishing by the lake.
   Once a week, he retreats to his favorite pub, where laughter and stories flow
   as easily as the cold beer in his hand. Dressed in sturdy jeans, a well-worn
   flannel, and practical boots, his style reflects his pragmatic nature,
   uncomplicated, reliable, and effortlessly classic. Though a man of few words,
   his firm handshake and warm smile speak volumes about his honest and steady
   character.

Characteristics:
• Age: 50
• Gender: Male
• Marital Status: Married
• Residence: Small town in the Czech Republic
• Financial Status: Financially stable
• Occupation: Warehouse worker
• Education Level: Technical high school
• Number of Children: 2 sons
• Lifestyle: Small-town, working-class
• Hobbies and Interests: Football, fishing

Traits:
• Loyal & Devoted
• Hardworking
• Easygoing
• Routine-Oriented
• Stubborn
• Not Tech-Savvy
• Occasionally Gruff
```

Listing 4.2: The example of a generated persona profile description including the name and surname, styling description, characteristics, and traits.

From a general point of view, by using the persona stylization, a cooperative or adversarial interaction can be simulated. In the first one, the agent strives to achieve a consensus as quickly as possible and is open to other opinions. On the other hand, with adversarial interaction, at least some of the agents try to avoid mutual agreement and undermine the views of others.

The summarizing algorithm, which incorporates all tasks performed by the agent's modules, is described in Algorithm 1.

Algorithm 1 Single Agent Processing Algorithm (including Memory Consolidation)

```
Initialization:
Input profile description: (name and surname, description, characteristics, traits)
Initialize short-term memory STM
Initialize long-term memory LTM with profile description
Initialize retrieved memories RM ← LTM
Initialize emotional level E
Initialize NeedToTalk = f(E, RM)
if group debate mode then
    Initialize consensus openness CO = g(E, RM)
end if
while conversation is active do
    if agent is selected to speak then
        Agent generates message M
    end if
    Agent Processing:
    if received last message M then
        Process Perception Module:
        Extract (topic, keywords) from M
        Memory Consolidation:
        Filter relevant information from STM for long-term memory storage
        if information is significant and relevant then
            Store STM information in long-term memory LTM
        else
            Discard irrelevant information
        end if
        Update Short-Term Memory:
        Replace STM with new information from perception
    end if
    Retrieve Memories:
    Retrieve relevant memories RM from STM and LTM
    Recalculate Need to Talk:
    Compute NeedToTalk = f(E, RM)
    if group debate mode then
        Compute new consensus openness CO = g(E, RM)
    end if
    Decide Action in Action Module:
    if agent is not speaker then
        if agent is selected to reflect then
            Perform self-reflection
            Perform memory consolidation and Update STM with the insights
        else if planning is needed then
            Generate or update plans
            Perform memory consolidation and Update STM with the plan
        end if
    end if
    Update Emotional Levels
end while
```

4.2 Simulation Architecture and Scenarios

This section describes the simulation itself in terms of communication patterns and the main simulation loop. With that, the new concept of how the agent's speaking order is chosen is introduced. I discuss the design and consensus achievement settings for the group debate simulations. Lastly, I provide the modeling example of a single simulation run and possible simulation scenarios.

4.2.1 Communication Pattern

For the communication pattern, I divided the task into two subgroups based on the main goal: *free discussion* and *group debate*. For both, selecting the order in which the agents speak is needed. Based on the lacking characteristics of the concepts published to date, I introduce a new approach called *One-by-One Talk with Agent's Need to Talk*, described in depth in this section.

Free Discussion and Group Debate

In a free discussion task, the main goal is not achieving the final solution, but we are mostly interested in the agents' interaction. This starts with a predefined topic, and the agents freely communicate and contemplate opinions. Therefore, we do not force the achievement of a consensus, but we are more focused on the interaction structure. I am letting the discussion execute for a specified amount of time, which is, in the system case, calculated based on the exchanged agents' messages (each agent's speech is one message).

On the other hand, we require the final solution for the group debate, so the communication is performed until the consensus is achieved and the result is provided. How consensus is reached is described later in Section 4.2.2.

One-by-One Talk with Agent's Need to Talk

In a general context of the communication pattern, I can split the existing literature into four approaches: *pair dialogue*, *one-by-one*, *simultaneous talk*, and *external selector*. These concepts are described in more detail in Section 3.3.5.

Because the focus is also on the interactions with more than two agents, I immediately reject the first approach.

With the one-by-one method, I evaluate this solution as unrealistic regarding the credibility of real human communication. Moving to the simultaneous talk, even though this proposal improves the previous pattern designs, this solution still falls short of the required credibility in terms of human interaction. Lastly, the external selector uses an agent to choose the order of the agent's messages. This approach is more advanced than the previous; however, such a concept builds the order using the external entity and not on the agent's inner states and needs. Based on this, I designed a new approach for selecting the order of the agent's speech called *One-by-One Talk with Agent's Need to Talk*.

In this architecture, I propose using the standard one-by-one communication pattern. However, agents do not follow the predefined talk order. Instead, the system calculates the agent's need to talk based on multiple relevant facts before each speech. These facts include the agent's current emotional state and relevant memories. With that, I provide the agent with the current number of messages exchanged and the message limit, which enforces the decreasing need to talk in time. It is generated using a single LLM call in an integer range from 0 to 10. The need to talk level for each agent is re-evaluated after each message. This approach is depicted in Figure 4.11.

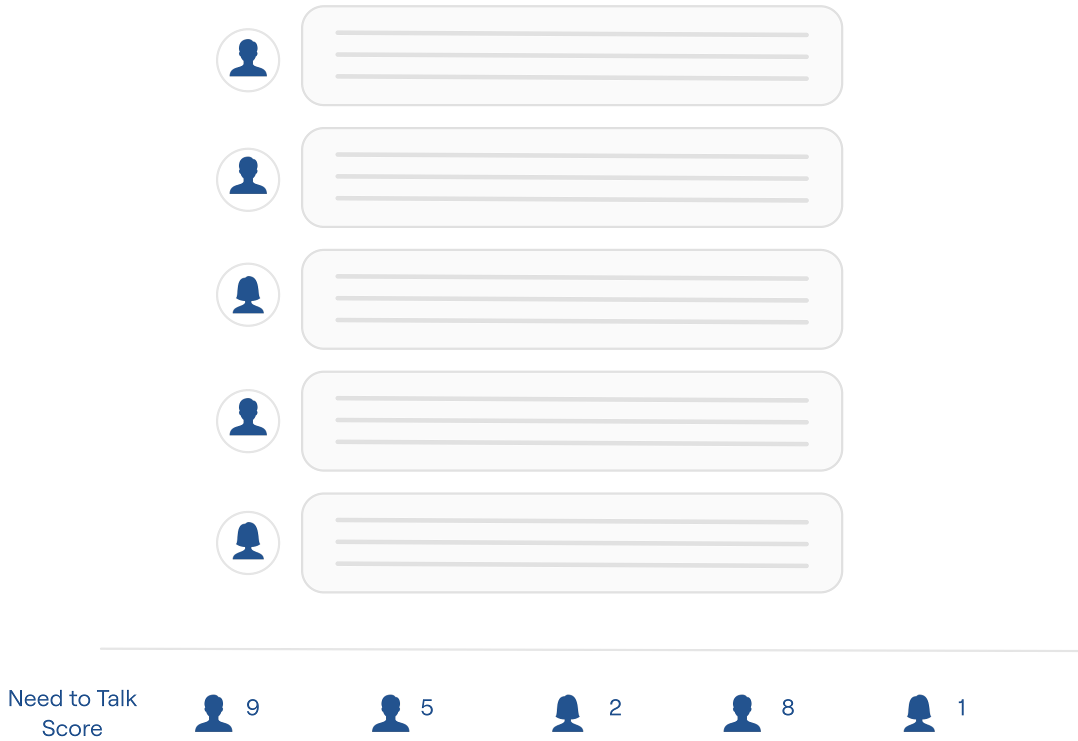

Figure 4.11: The concept of the one-by-one talk with agents needs to talk. Before each message, it evaluates the need of the individual agent to speak. Such a score is generated in an integer range from 0 to 10.

To select the speaker's order using agents' need to talk scores, I design two approaches of how the system chooses the next speaker, and so *maximum likelihood* and *softmax*. The agent with the highest score will speak next for the maximum likelihood method. Using the softmax approach, I use a concept similar to how the LLMs select the following tokens in the output stream. I use the softmax function with the *temperature* parameter for this. With it, we can set the uncertainty of the selection.

In general, while using this architecture, the agent may speak more than once in a row. I evaluate this fact as authentic and credible to the actual communication. However, to offer wider possibilities with the system, I optionally force the system not to repeat the speaker in a row, similarly to [109].

Finally, the standard mesh topology, when all agents are connected, is used for the topology, so each agent observes every communication.

4.2.2 Discussion with Consensus Achievement

In the group debate simulation, I use the same concepts as for the free discussion, and the speaker selection using agents needs to talk scores. Additionally, if a discussion task requires a final solution, this is processed until a consensus is reached during the group debate.

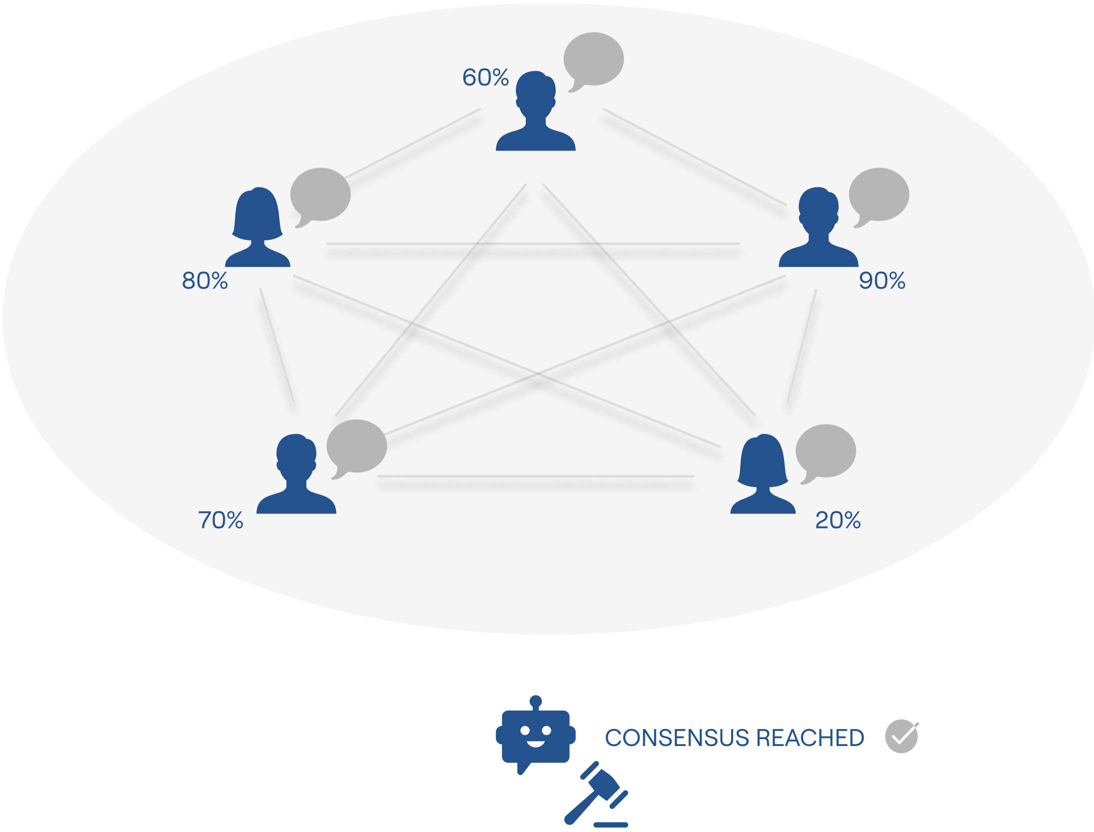

Figure 4.12: Achieving a consensus in a debate with fully connected topology. After each message, the agents set their openness to consensus scores in an integer range from 0 to 10. The excluded judge will decide based on the scores, the need to talk levels, and the communication history whether consensus is reached.

To achieve the final consensus, I first let each participating agent calculate the level of openness to consensus in the range of 0 to 10. Based on values from all agents, their need to talk scores, and the conversation history, the non-participating model (called the *judge*) will automatically, without intervention, decide whether the consensus was reached or not. If so, the discussion ends. Notably, the agent is reminded of the majority voting approach, but not forced to do so. This consensus-reaching design is shown in Figure 4.12.

For the judge agent, I consider two approaches: *sliding window* and *persona architecture*. For the sliding window, I designed a limited buffer for the full messages, their authors, and their emotional states for each message. The persona-like judge uses similar concepts as described in Section 4.1.1 as much as possible (perception, planning, reflection, memory system, and emotional level generation). The consensus decision uses its own memories. If consensus is reached for both systems, the judge generates the final answer in a single message.

4.2.3 Modeling Example of a Simulation Run

In this section, the modeling example of the simulation run is described. I will use the *free discussion* simulation scenario for the description. The algorithm for this simulation type is more formally described in Algorithm 2 using the need to talk pattern with the maximum likelihood method. The consensus-related tasks are added for the *group debate*. Namely, each agent calculates the *openness to consensus* value in the initialization part and updates the level during the simulation run. A single *judge* agent is added, which decides whether consensus is reached or not. Finally, the consensual answer is returned. The algorithm for the group debate simulation type is formally described in Algorithm 3.

Algorithm 2 Free Discussion Simulation (Need to Talk Pattern with Maximum Likelihood)

Initialization:
Set *topic*
Select participant *agents* $\mathcal{A} = \{A_1, A_2, ..., A_n\}$
Define total *messages limit* T
for each agent $A_i \in \mathcal{A}$ **do**
    Initialize *agent* (memories, emotional state, $NeedToTalk_i$)
end for
$t \leftarrow 0$
while $t < T$ **do**
    if $t = 0$ **then** ▷ Start of conversation
        Select first *speaker* $A_s \leftarrow \arg\max_{A_i}(NeedToTalk_i)$
    else
        for each agent $A_i \in \mathcal{A}$ **do**
            Perform *agent's processing* including $NeedToTalk_i$ recalculation
        end for
        Select next *speaker* $A_s \leftarrow \arg\max_{A_i}(NeedToTalk_i)$
    end if
    A_s generates *message* M_t
    $t \leftarrow t + 1$
end while

For the modeling run, first, I break down the simulation process from a high-level point of view. As input, the simulation run has the conversational topic. Also, the selected personas who should participate in the conversation are provided. The last input is the type of simulation (free discussion or group debate). In a free discussion scenario, the conversation itself is the result of the simulation run. In addition to the free discussion, the final consensual answer is obtained for the group debate.

The imaginary modeling example of the simulation run with the topic *„What is the best ice cream flavor?"* is further described.

Simulation Inputs

For the simulation run, multiple inputs have to be provided:

- Topic: The topic of the conversation is „What is the best ice cream flavor?". So, during such a discussion, the agents will share their opinions on which flavor (for example, vanilla, chocolate, strawberry, or mint) is the best ice cream flavor.
- Selected Personas (Agents): The personas chosen for this simulation are:
    - *Alena Nováková*: A foodie, passionate about trying new flavors and often argues for unusual choices.

Algorithm 3 Group Debate Simulation (Need to Talk Pattern with Maximum Likelihood)

Initialization:
Set *topic*
Select participant *agents* $\mathcal{A} = \{A_1, A_2, ..., A_n\}$
Create new *judge* agent J
for each agent $A_i \in \mathcal{A}$ **do**
    Initialize *agent* (memories, emotional state, $\mathit{NeedToTalk}_i$, $\mathit{OpennessToConsensus}_i$)
end for
$t \leftarrow 0$
$\mathit{ConsensusAchieved} \leftarrow$ **False**
while not $\mathit{ConsensusAchieved}$ **do**
    if $t = 0$ **then** ▷ Start of debate
        Select first *speaker* $A_s \leftarrow \arg\max_{A_i}(\mathit{NeedToTalk}_i)$
    else
        for each agent $A_i \in \mathcal{A}$ **do**
            Perform *agent's processing* including $\mathit{NeedToTalk}_i$ recalculation and $\mathit{OpennessToConsensus}_i$ recalculation
        end for
        Select next *speaker* $A_s \leftarrow \arg\max_{A_i}(\mathit{NeedToTalk}_i)$
    end if
    A_s generates *message* M_t
    Consensus Evaluation:
    Judge J evaluates $\mathit{OpennessToConsensus}$ values
    if J determines consensus is reached **then**
        $\mathit{ConsensusAchieved} \leftarrow$ **True**
    else
        $t \leftarrow t + 1$
    end if
end while
Return Consensual Answer M_{final} (generated by *judge* J)

- *David Procházka*: A classicist, believing in the traditional, often advocating for vanilla or chocolate.
- *Eva Horáková*: A health-conscious individual, leaning toward options like fruit-based flavors.
- *Lukáš Jelínek*: A child who likes colorful and fun choices strongly favors mint chocolate chip.

- Type of Simulation: Free discussion. The goal is to simulate an ongoing conversation without reaching a consensus.

Example Interaction Simulation

The simulation begins with the topic set, and the agents are initialized with their characteristics and other memories. In the free discussion scenario, the agents will interact in the following steps:

1. Initialization: Each agent is assigned an emotional state, a memory base, and a need to talk level based on their persona. For example, agent *Alena Nováková* might have a high need to talk about unique flavors. In contrast, agent *David Procházka* will have a lower need to talk but will be more inclined to argue for traditional flavors.

2. Start of Discussion: Agent *Alena Nováková* is selected to speak first, as it has the highest need to talk score. Agent *Alena Nováková* says: "I think the best ice cream flavor is pistachio because it's exotic and has a unique taste that no other flavor can match."

3. Following Discussion: Agent *David Procházka* responds, choosing vanilla as its choice: "I understand your point, Alena, but for me, vanilla has always been the best. It's simple, versatile, and universally loved." Agent *Eva Horáková*, concerned with health, argues: "While I respect vanilla, I prefer fruit-based flavors, like strawberry or lemon. They are refreshing and lower in sugar than creamy options." Agent *Lukáš Jelínek*, the child persona, chimes in: "I think mint chocolate chip is the best because it's colorful and fun. Plus, it has chocolate, which makes it extra tasty!"

4. Ongoing Discussion: The conversation continues as agents freely express their opinions, share personal experiences, and try to persuade others with their arguments. Agent *Alena Nováková* adds: "Pistachio is just so much more interesting. It's unlike all the other basic flavors everyone already knows about." Agent *David Procházka* counters: "Basic doesn't mean bad. Vanilla is simple, but that's what makes it so good. It's the foundation for so many desserts." Agent *Eva Horáková* emphasizes: "You know, I think fruit flavors like strawberry are a healthier option, and they're just as delicious. Plus, you get a burst of freshness." Agent *Lukáš Jelínek* agrees, but with a twist: "I like the fun of mint chocolate chip, and it has that cool minty flavor! Plus, who doesn't love the chocolate chunks in it?"

5. Emotional and Opinion Dynamics: As the discussion continues, each agent adjusts their emotional state and need to talk values based on the ongoing conversation. Some agents may feel more passionate as they defend their choices, while others may cool down if they think their point has already been made. Agent *Alena Nováková* might become more passionate about pistachio if others disagree, while agent *David Procházka* could start emphasizing the nostalgic value of vanilla.

6. End of Discussion: Since it is a free discussion, there is no need for consensus, and agents express their viewpoints without expecting a final decision. The conversation will continue when the maximum allowed time (number of messages exchanged) is reached. Also, the overall opinion resulting from the conversation may not converge. For example, agent *Alena Nováková* might conclude: "Okay, I think we all have our favorites. Whether it's pistachio, vanilla, strawberry, or mint chocolate chip, each flavor has its charm!"

Final Output of the Simulation Run

The simulation run ends without a definitive conclusion or consensus. The conversation provided a variety of opinions, but the system did not require agreement, as the goal was to simulate a free discussion. The result of the simulation is a collection of diverse perspectives from all agents, showcasing the variety of tastes and preferences regarding the best ice cream flavor.

4.2.4 Simulation Scenarios

Following the simulation division from Section 4.2.1, the simulation scenarios can be grouped based on the free discussion and group debate.

Free Discussion

The task can be divided further based on the following criteria for the discussion without the need for consensus achievement.

Cooperative and Adversarial Discussion Based on this division, we can force agents to cooperate and be open to other agents' opinions. On the other hand, adversarial settings force some agents to question the views of others and add very different opinions. These can be set by agents' *social goals* (following [118]), as described further in Chapter 5.

Personas with and without Relation Discussion Naturally, these two tasks are scenarios where, in the first case, it considers agents in some relationship, familial or otherwise. Secondly, the system executes the discussions of previously unknown personas. Similarly, the *relationships* (following [118]) are described in Chapter 5.

Group Debate

For the group debate, the main goal is to achieve the final solution and consensus. The specific simulation scenarios can be divided into the following subsets for the two main concepts. Of course, general scenarios and topics can also be used.

Commonsense Tasks In these tasks, the group has to answer questions focused on applying general world knowledge and reasoning beyond what is explicitly stated. The answer could be yes/no, or multiple choice.

Math Tasks For the math task, from a general point of view, the task can be split into two subgroups according to the complexity of the assignment answer: *single value* solutions and the *word problems*.

Chapter 5

Implementation of the PerSimChat Framework

On a higher level, this chapter describes the implementation of the proposed *PerSimChat* framework architecture.

First, using the framework itself, I enumerate the third-party tools and external services used for the simulations. Secondly, this chapter focuses on the core concepts of implementing the system. Third, extending the tool with a user interface and providing a web application is described. Last but not least, I mention the application deployment on the Lakmoos AI, s.r.o. servers.

5.1 Third Party Tools and External Services

The implementation of the PerSimChat framework uses multiple external services and tools, especially for communication with the model deployments. For large language models (LLMs), the two services are available to be used, and so *OpenAI API*[1] and Azure models. For the Azure LLMs, I use the *Azure OpenAI*[2] for OpenAI models and *Azure Machine Learning Serverless Endpoints*[3] for open-source models. The proprietary Lakmoos AI models were used as localhost calls through the SSH tunnel to the Lakmoos server side.

The API used for communication with models is the *Python OpenAI* API library, and so the synchronous or asynchronous versions and the Azure API version as well. I use the well-known *Jinja2* library for templating the model input prompts. The *Tenacity* library is used to manage API calls.

During dynamic memory retrieval, I use the *Faiss*[4] library for the embeddings closest matches searches using the Faiss index. The embeddings are created using the *text-embedding-ada-002* model by OpenAI. For providing the tokens count during the complex simulation analyses, I use the *Tiktoken*[5] library for encodings for the GPT-4 model.

[1]`https://platform.openai.com`, accessed: 2025-05-02.
[2]`https://azure.microsoft.com/en-us/products/ai-services/openai-service`, accessed: 2025-05-02.
[3]`https://azure.microsoft.com/en-us/products/machine-learning`, accessed: 2025-05-02.
[4]`https://github.com/facebookresearch/faiss`, accessed: 2025-04-29.
[5]`https://github.com/openai/tiktoken`, accessed: 2025-04-29.

5.2 The Core Concepts of the PerSimChat Framework

The implementation of the system is available in two such variants, and so the *synchronous* and *asynchronous*. The faster and more optimized version is the asynchronous version; however, the synchronous version is also maintained for use cases where the asynchronous version can exceed the number of model calls.

I divide the implementation of the PerSimChat framework into five subareas: *agent*, *environment*, *judge*, *profiles conversion and generation*, and the main *simulation loop*.

5.2.1 Agent

The agent is mainly implemented with so-called *modules*. These modules are the *cognition modules* or *profile module*.

The profile module stands for the emotional level generation using Ekman's six basic emotions [27]. Every emotion is generated in the range 0–10 using the LLM call, so the emotional level holds six values, one for each emotion. The cognition modules hold the core of every agent. The implemented human cognition functions are: *memory*, *action*, *natural language*, *perception*, *planning*, and *reflection*. The memory module is further divided into primarily *long-term* memory, *short-term* memory, *memory consolidation*, and dynamic *retrieval*.

Memory Modules

The framework represents a single memory as a *memory structure*. It holds the *memory type* (one of the following: perception, reflection, plan, profile, conversation topic, or social goals), creation timestamp, last access timestamp, importance, overarching topic, keywords, and emotional state when storing such information in memory. The last access timestamp is updated when the memory is accessed through dynamic retrieval.

In a general point of view, the short-term memory implements the basic functionality, and so its primary purpose is to store a single information. To optimize dynamic retrieval, the topic and keywords embeddings are also calculated in short-term memory and stored to optimize the embedding creation process.

More complex is the concept of long-term memory, which stores multiple memories at once based on the time hierarchy. At the beginning of the simulation, the description of the persona's profile for which the current agent stands is stored as the first memory in the memory stream. With that as a second memory, the system stores the conversation topic and optionally, as a third memory, the agent's social goals. Note that the relationships are not stored in long-term memory and are provided for each message generation to enforce the consistency of the agent's behavior.

Following Generative Agents [79], the size of long-term memory is limited to 100 memories, which are sufficient for our purposes. When the memory is full, pruning is incorporated when storing the new memory. To make such a process the least computationally demanding, one of the memories with the oldest last access timestamp is removed, making space for the new information to be stored in the long-term memory. Similarly to short-term memory, long-term memory pre-calculates the Faiss index for the topics and keywords to optimize the dynamic retrieval process.

When moving the memory from short-term to long-term, the system performs the memory consolidation process. In this process, when new information is passed into short-term memory, the previous information is saved in long-term memory. Optionally, the framework can use the process of *memory filtering* in which the LLM is tasked to decide whether short-term memory should be saved in long-term memory. It is based on memory similarity (new memory to be stored and memories already stored), emotional state, and the importance of memories.

One of the core concepts of short-term and long-term memory is the dynamical memory retrieval process. This approach is inspired by [79]. The main task of memory retrieval is to select the most important memories based on multiple criteria to have an adequate number of memories that can be filled into the prompt without exceeding the model's input context size. The complete process is as follows. First, the framework generates so-called *focal points*. These are the three most salient high-level questions we can answer about the subjects in the statements. These questions serve as a retrieval query, returning relevant memories. From the work mentioned, I chose 30 memories to be retrieved for each focal point.

There are many possible implementations of a retrieval function. I decided to follow [79] and implement the three used components: *recency*, *importance*, and *relevance*. With that, I extend this retrieval function with two new concepts: *emotional state* and *short-term memory relevance*. Recency assigns a higher score to recently accessed memories, and so they have a newer last access timestamp. The recency is treated as an exponential decay function with decay factor 0.995. Importance assigns a higher score to memories that are important to an agent, which stands for the stored memory structure importance parameter. Relevance assigns a higher score to memories related to the current situation. This is done by comparing the query topic and keywords with the topics of memories and keywords using the Faiss index search.

To these components in this work, I optionally add two other elements important to the system's agent's design: emotional state and short-term memory relevance. The emotional state compares the current agent's emotional state with the emotional state when storing a single memory. It assigns a higher score to memories of an emotional state that resembles the current one. It is done by calculating the cosine similarity between the created emotional state vectors. The short-term memory relevance is evaluated similarly to a base relevance, with the difference that the focal point as a query is replaced with the information currently stored in short-term memory. Based on that, it assigns a higher score to memories that are more similar to what the agent perceives.

The final retrieval score calculation is provided in Equation 5.1, where all αs are set to 1.

$$\begin{aligned} score = {} & \alpha_{recency} \cdot recency \\ & + \alpha_{importance} \cdot importance \\ & + \alpha_{relevance} \cdot relevance \\ & + \alpha_{emotional\ state} \cdot emotional\ state \\ & + \alpha_{stm\ relevance} \cdot stm\ relevance \end{aligned} \tag{5.1}$$

The top-ranked memories that fit within the language model's context window are included in the prompt. Such a process is shown in Figure 5.1.

General Cognition Modules

The rest of human brain cognition functions are implemented in the same-named modules: *action*, *natural language*, *perception*, *planning*, and *reflection*.

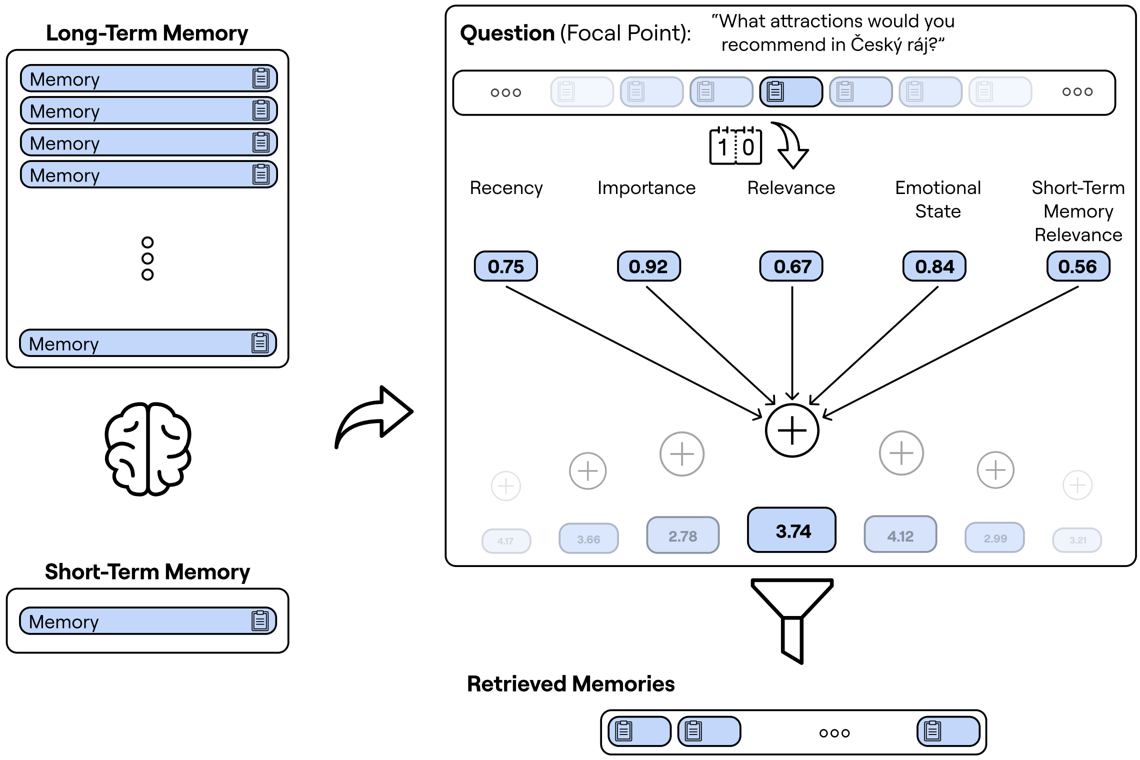

Figure 5.1: The visualization of a single focal point's dynamic memory retrieval process.

Action Module Based on the action module architecture described in Section 4.1.1, it implements the need to talk generation, consensus openness score generation, and the logic for selecting one of the actions performed for a single agent: planning or reflection. For such processes, it uses relevant memories and the current emotional level of the agent.

Natural Language Module The natural language module implements the logic for speech generation. Alongside the information needed for the message generation and the parts of the prompt that are responsible for the stylization of the message, I also incorporate the knowledge levels in the speech generation prompt. In simulation scenarios, I consider five levels of knowledge, and so *no knowledge*, *basic awareness*, *elementary knowledge*, *intermediate knowledge*, *advanced knowledge*, and *expert knowledge*. For each knowledge level, the typical age range, education, knowledge characteristics, and limitations, with example responses, are provided. The example of the part of the prompt for the intermediate knowledge level is shown in Listing 5.1.

```
Example Response:
Q: "What is gravity?"
A: "Gravity is what makes things fall to the ground. It's why planets orbit the
   sun!"
Level 3: Intermediate Knowledge

• Typical Age Range: 16-22 years old or an adult with a high school diploma or
   practical work experience.
```

```
• Education: High school graduate, technical training, or early university
  studies.
• Knowledge Characteristics:
  • Can discuss high school-level subjects in some depth.
  • Can analyze basic topics critically.
  • Understands fundamental economics, politics, and science.
• Limitations:
  • Cannot provide expert-level knowledge.
  • May have misconceptions or an incomplete understanding of complex theories.
  • Lacks specialized knowledge in professional fields.
```

Listing 5.1: The part of the prompt with an example of the intermediate knowledge level characteristics.

Perception Module The perception module implements the basic logic for the perception of a single persona of the message in the conversation.

Planning Module The planning module implements the logic of long-term planning of the agent. For system purposes, the planning module implements the *persona identity revision*, whether it *should react to the last message*, and *enforces the social goals*, if defined.

Reflection Module This module implements the inner reflection of the agent. First, it generates inner *thoughts*. Based on the relevant memories, I prompt the LLM similarly to [79] to create five high-level insights about the agent's memories. Then, the keywords are extracted for each insight, and each pair of insight with keywords is stored in the agent's memory.

5.2.2 Environment

One of the essential parts of the PerSimChat framework is the simulation environment. For the multi-agent conversation, the environment is primarily needed to store the selected personas and simulation settings. At the beginning of the simulation, the environment is responsible for creating the selected personas with predefined parameters and a judge agent if the group debate pattern is performed.

The environment is also responsible for providing social goals and predefined relationships to the agents, following [118]. Social goals are specified for each agent separately and stored in their long-term memory before the beginning of the conversation. To enforce the defined relationships between personas, the relationships are provided for the agent during each message generation. In this work, I consider five types of relationships: *neutral* (unknown, known by name, and acquaintance), *positive relationships* (friends, close friends, allies, admiration, and romantic), *negative relationships* (rivals, enemies, annoyance, and betrayed), *hierarchical relationships* (leader-follower, authority-subordinate, and mentor-mentee), and *family relationships* (parent-child, siblings, grandparent-grandchild, uncle/aunt-nephew/niece, cousins, spouse, stepparent-stepchild, and adoptive parent-child).

5.2.3 Judge

The judge agent is in the PerSimChat framework used for the group debate simulation with consensus achievement. Based on architecture design, my system offers two types of judge agents: *sliding window* or *persona architecture*.

The base sliding window judge agent stores in the basic agent memory for each message its complete text word by word, the author, and the emotional state of the author. At the same time, the external parameter defines the maximum memory size. The memory works as a sliding window concept in a standard manner.

For the persona architecture, the judge possesses the most similar architecture to the personas incorporated in the communication. It has its own short-term and long-term memory system with perception, dynamic retrieval, planning, and reflection. This simulates more realistically the persona-like facilitator.

5.2.4 Profiles Generation and Conversion

For the framework simulations, the personas' descriptions have to be provided in an expected format, which is composed of the persona's *name and surname*, *description* with one paragraph length, list of *characteristics*, and list of *traits*. For the variety of personas and the property rights, the system allows the creation of a persona's description from scratch or also convert the Lakmoos AI, s.r.o. personas' descriptions into the required format.

For both, a single large language model (GPT-4-turbo, version *gpt-4-turbo-2024-04-09*) is prompted, using a zero-shot prompting technique, to generate from scratch or convert the persona.

5.2.5 Main Simulation Loop

The core of the PerSimChat framework is the implementation of simulation runs. The system offers two versions of the simulation, synchronous and asynchronous. Both algorithms are pretty similar, and the logic stands for the algorithm described in Section 4.2.3. The message texts and metadata are continuously yielded from the simulation loop. The exact loop implementation is used for the CLI run as well as for the user interface requests.

5.3 From Command-Line Interface to User Interface

I created the user interface for demonstration to increase usability and awareness of the PerSimChat framework capabilities. This was implemented using the *Typescript*[6] language with *React*[7] library and *Node.js*[8] environment. The complete PerSimChat framework as a web application backend is provided using *FastAPI*[9] API with a *Uvicorn*[10] web server.

The user can choose from the predefined list of available models and personas. Based on the models capabilities and pricing methods the four models by OpenAI[11] were chosen, namely GPT-4-turbo, GPT-4o-mini, GPT-4o, and o1-mini in versions *gpt-4-turbo-2024-*

[6] https://www.typescriptlang.org/, accessed: 2025-04-29.
[7] https://react.dev/, accessed: 2025-04-29.
[8] https://nodejs.org/en, accessed: 2025-04-29.
[9] https://fastapi.tiangolo.com/, accessed: 2025-04-30.
[10] https://www.uvicorn.org/, accessed: 2025-04-30.
[11] https://openai.com/, accessed: 2025-04-30.

04-09, *gpt-4o-mini-2024-07-18*, *gpt-4o-2024-08-06*, and *o1-mini-2024-09-12*. The system currently offers 10 predefined personas in the current version.

The user can also change the simulation settings and experiment with the simulation parameters. It can be chosen between asynchronous and synchronous versions, while the asynchronous version is strictly recommended if there are no problems with exceeding the number of requests in time. Especially for the user interface, complex analysis is available, which lets score the single large language model (GPT-4-turbo) the *coherence*, *sentiment*, and *engagement* of the single message in the range of 0–10 floating point.

The user can use one of the predefined example *conversation topics* or use the text area to type the custom topic.

Also, a single agent setting is available for the user, while the user can set the functionality on or off, determining whether it is performed. The possible parameters for the settings are *planning and reflection*, *memory filtering*, *emotional state retrieval*, *short-term memory retrieval*, *including defined relationships*, and *including social goals*. The user should define the social goals for personas, and they can also be left empty.

Available simulation scenarios are *free discussion* and *group debate*, while for both, the communication pattern can be set to one of the following: *round robin* or *need to talk*. For the need to talk, the two types are available, *maximum likelihood* and *softmax*, while for softmax, the user can change the *temperature* value. For this conversation pattern, the user can optionally allow *speaker repetition*, which means that a single persona can speak multiple times in a row. For free discussion, a *number of messages* has to be set, and for group debate, the user can set the judge parameters. That means, especially, model similarly to the personas, if we use *persona architecture* for the judge or *sliding window* by default. For the default variant, the user has to set the judge *window size*.

I provide visualisations of the real-time parameters in the user interface as well. For each persona, the *last message response time* is shown with *coherence*, *sentiment*, and *engagement* if complex analysis is set. With that, the trend for the *average response time* is shown, and for the complex simulation analysis parameters as well, if set. The analysis made during the simulation run is the *total number of messages*, *average response time*, *last message length* (provided in words and tokens), *most active agent name*, *highest coherence*, *average engagements*, and *average sentiment*. The *number of messages* sent during the simulation run is shown for each persona.

The web application's user interface is provided in Appendix A.

5.4 Deployment of the Tool at Lakmoos AI

The PerSimChat framework with the user interface extension is a standalone web application. For now, the system was deployed for the external partner as a running container composed of two separate images — one for the backend and one for the frontend of the PerSimChat demonstration application. I used well-known *Docker*[12] platform for containerization, and the application is provided by the Lakmoos AI, s.r.o. publicly for demonstration purposes[13].

[12] https://www.docker.com/, accessed: 2025-04-29.
[13] https://persimchat.lakmoos.com/, accessed: 2025-04-30.

Chapter 6

Evaluation of Experiments and Simulations

This chapter analyses the performance of the proposed system in multiple experimental cases. First, I evaluate the performance of the *PerSimChat* framework with two baseline solutions while using a single large language model (LLM) as an evaluator. I compare these systems on multiple evaluation criteria of the free discussion. Secondly, my solution is compared to the baselines in a paired comparison using the FairEval [103] experimental architecture. As the third experiment, I compare the performance of the group debate simulation scenario to three other multi-agent debate (MAD) frameworks using the benchmarks focused on commonsense and math. In connection with free discussion, a human evaluation study was conducted through a public survey. Lastly, the system is studied in a few use cases, and the free discussion simulation runs are discussed in detail.

In all experiments, I use *Python* version 3.10.16. The models involved during the experiments are provided using the *OpenAI API*[1] version 1.71.0 and *Azure services*[2] for model deployments. The *AutoGen* library is tested with all *autogen-core*, *autogen-agentchat*, and *autogen-ext* version 0.5.1. The experiments run locally on a computer with Ubuntu 24.04.2 LTS operating system (Linux kernel GNU/Linux 6.11.0-21-generic x86_64), processor Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz with 16 GiB RAM available and Intel Corporation Ethernet Connection I219-V (rev 21) network interface card with 1 Gbps internet connection. The model responses are obtained through the API requests to Azure deployments. The temperature is set by default to 1 for all LLMs used in the experiments.

6.1 Evaluation of Free Discussion using a Large Language Model

Evaluation of free discussion is a complex and challenging task. In such an experiment, I need to score the system based on how naturally it performs and how well it simulates a real-world persona conversation. Following these requirements, the system is rated using several criteria.

[1] https://github.com/openai/openai-python, accessed: 2025-04-29.
[2] https://azure.microsoft.com, accessed: 2025-04-29.

6.1.1 Experimental Setup and Evaluation Process

Based on recent work and available resources, conversation transcripts are scored by a single LLM, which replaces human crowd workers.

To compare the proposed PerSimChat system with existing solutions, the published state-of-the-art tools do not support multi-agent communication according to our requirements, or the available implementations support only two agents [60, 118]. Based on that, my system is compared to two created baseline solutions that were implemented. First, I prompted a single LLM with persona descriptions to generate the complete personas' communication (noted as *Single-Agent Zero-Shot*). As a second baseline, I used the AutoGen[109] library for implementing the baseline multiple agents' communication. All systems during experiments are set with the same persona descriptions, and the output is always transformed into the same format for the evaluator model.

In this experiment, the systems are gradually evaluated using multiple criteria dimensions, while a single LLM rates their performance. The experiment uses 10 dimensions, of which the first four are taken from the evaluation framework SOTOPIA-EVAL [118]. The dimensions are provided with the score range, where a higher score corresponds to better performance, in the format **[lower bound–upper bound]** form with the explanation:

Goal Completion (GOAL) [0–10] The goal completion focuses on the extent to which the agent achieved their *social goals* [118].

Believability (BEL) [0–10] The believability focuses on the extent to which the agent's behavior is perceived as natural, realistic, and aligned with the agent's character profile, thus simulating believable proxies of human behavior [118].

Knowledge (KNO) [0–10] Knowledge captures the agent's ability to acquire new information actively. Specifically, it considers the following criteria: What information the agent has gained through the interaction, whether the information the agent has gained is new to them, and whether it is important to them [118].

Relationship (REL) [-5–5] Relationship captures the fundamental human need for social connection and belonging. In this dimension, the participant's *relationship* with the other agent(s) before the interaction is asked about. The LLM then evaluates whether agents' interactions with others help preserve or enhance their relationships [118].

Credibility (CRE) [0–10] Credibility focuses on the evaluation of the agent's statements in terms of a tone that feels informed, grounded, and responsibly reasoned. The evaluator pays attention to whether the agent avoids exaggeration or casual overstatements, providing measured, consistent, and plausibly justified information.

Turn-Taking and Flow (TTF) [0–10] This metric evaluates the quality of turn-taking and conversational flow between the agents in the discussion. It prioritizes interactions that reflect thoughtful pacing and balanced participation, where agents allow each other space to contribute, and transitions feel purposeful and connected.

Content Depth and Relevance (CDR) [0–10] This dimension rates how the statements reflect the agent's distinct internal world, such as past experiences, personal values, goals, and emotional states. It assesses how well each contribution builds meaningfully on the ongoing conversation while preserving each agent's unique perspective, even if that leads to occasional disagreements or narrative tangents.

Social and Emotional Responsiveness (SER) [0–10] Social and emotional responsiveness metric places the highest value on emotionally expressive, socially attuned responses that emerge naturally through back-and-forth interaction between distinct agents. It especially looks for moments where agents demonstrate emotional awareness, not just through empathy or support, but also through more human behaviors like teasing, defensiveness, validation, or disagreement.

Goal Progression or Task Solving (GPTS) [0–10] This dimension focuses on whether a shared sense of purpose or individual goals emerge naturally through interpersonal interaction. The evaluator rewards situations where distinct agents contribute unique perspectives or partial solutions that build on each other over time, reflecting distributed cognition and adaptive teamwork. The LLM gives extra weight to signs of flexible planning, shifting strategies, and agents constructively challenging or supporting each other's approaches.

Conversation Closure (CLO) [0–10] Conversation closure evaluates how effectively the agents collaboratively bring the conversation to a meaningful conclusion. The LLM evaluator focuses on whether the agents acknowledge the end of the discussion in a natural, coordinated manner, looking for signs of multiple agents signaling closure, such as expressing final thoughts, summarizing key points, or giving each other space to close the conversation.

Following publications [13, 103, 118] and requests pricing, I selected *gpt-4-turbo-2024-04-09* model by OpenAI[3] as an evaluator model. The *gpt-4o-2024-08-06* model is primarily used for the agent based on reasoning capacities and pricing. During the experiments, I also report the two models by Mistral AI[4], namely Mistral Small (*mistral-small-2402*) and Mistral NeMo (*open-mistral-nemo-2407*). For the experimental setup, other models by Meta[5], Cohere[6], Microsoft[7], and DeepSeek[8] were also tested. However, these models lack reasoning capacities and lack output format requirements (Llama-3.3-70B, Llama-3.1-8B, Cohere-command-r-08-2024, Phi-4-mini-instruct, and Phi-3.5-mini-instruct), or their computations via Azure are very time-consuming (DeepSeek R1, DeepSeek V3, and Mistral Large). Also, the DeepSeek models are very clearly characterized by frequent vulgar expressions.

For the comparisons with Lakmoos AI[9] products, I compare the PerSimChat framework, including the proprietary LLM (noted as *LakMod*), and the complete Lakmoos system (noted as *LakSys*) for message generation. During these experiments, the GPT-4o model was used for decision-making tasks.

[3] `https://openai.com/`, accessed: 2025-04-29.
[4] `https://mistral.ai/`, accessed: 2025-04-29.
[5] `https://www.meta.com`, accessed: 2025-04-29.
[6] `https://cohere.com/`, accessed: 2025-04-29.
[7] `https://www.microsoft.com`, accessed: 2025-04-29.
[8] `https://www.deepseek.com/`, accessed: 2025-04-29.
[9] `https://lakmoos.com/`, accessed: 2025-04-29.

I used the two types of persona description, and so generated which were obtained by prompting the GPT-4o model. The second type is obtained by converting the Lakmoos persona descriptions into the required format by prompting the same single LLM.

The dataset of tasks created for this experiment is partially created from the FairEval questions [103] and extended with custom tasks. In total, the data set contains 100 experimental tasks and includes multiple topics: *generic*, *knowledge*, *roleplay*, *commonsense*, *fermi*, *counterfactual*, *writing*, and *controversial*.

During the first three comparisons, each system was run within a single run with the same settings on the same random subset of 30 tasks, while each simulation ran for 10 messages exchanged. For the final comparison of the PerSimChat framework with the two baselines, each system was run within three rounds with a subset of 15 tasks for a 20-message limit. Due to the cost associated with API-based models, each simulation is configured with three personas. For comparison with the Lakmoos system, all systems use the Lakmoos persona data for simulations. In the other tests, the automatically generated personas are set. During each comparison, the same personas for all systems are used.

For PerSimChat simulations, I use the asynchronous algorithm version and set the *emotional state retrieval* and *short-term memory retrieval* on, and I do not allow *speaker repetition*. The *need to talk* method was chosen with the *maximum likelihood* version. The other architecture design parameters are variable during the experiments. The architecture design settings with notation for the system versions are listed in Table 6.1. Table 6.2 provides the models used during the experiments.

Architecture Design	Planning and Reflection	Social Goals and Relationships	Memory Filtering
PerSimChat MF	✔	✘	✔
PerSimChat Base	✔	✘	✘
PerSimChat NoPR	✘	✘	✘
PerSimChat SGR	✔	✔	✘
PerSimChat LakData	✔	✘	✘
PerSimChat MiS	✔	✘	✘
PerSimChat MiNM	✔	✘	✘
PerSimChat LakMod	✔	✘	✘
PerSimChat LakSys	✔	✘	✘

Table 6.1: Description of architecture design settings for the PerSimChat framework. The *MF* stands for memory filtering, *Base* for the baseline version used in most of the tests, *NoPR* without planning and reflection, *SGR* with social goals and relationships included, *LakData* with incorporating converted personas from Lakmoos AI data instead of from scratch generation, *MiS* and *MiNM* stands for Mistral Small and Mistral NeMo models, *LakMod* for the proprietary Lakmoos model, and lastly, *LakSys* for the proprietary complete Lakmoos system for the message text generation.

During the dimensions evaluation, I first tested the different PerSimChat architecture designs. The main goal of these tests is to choose the best performance based on whether the memory filtering is performed, planning and reflection, or testing different models for the agents. Secondly, the evaluation focuses on the performance of the PerSimChat compared with the two baseline solutions while incorporating the predefined social goals for each persona and relationships between personas. In the third scenario, while using the

Architecture Design	Model
PerSimChat MF	GPT-4o
PerSimChat Base	GPT-4o
PerSimChat NoPR	GPT-4o
PerSimChat SGR	GPT-4o
PerSimChat LakData	GPT-4o
PerSimChat MiS	Mistral Small
PerSimChat MiNM	Mistral NeMo
PerSimChat LakMod	Lakmoos Model
PerSimChat LakSys	Lakmoos System

Table 6.2: List of models for each architecture design for the PerSimChat framework. The *MF* stands for memory filtering, *Base* for the baseline version used in most of the tests, *NoPR* without planning and reflection, *SGR* with social goals and relationships included, *LakData* with incorporating converted personas from Lakmoos data instead of from scratch generation, *MiS* and *MiNM* stands for Mistral Small and Mistral NeMo models, *LakMod* for the proprietary Lakmoos model, and *LakSys* for the proprietary complete Lakmoos system for the message text generation.

converted personas using the Lakmoos data, the baselines are compared with the PerSimChat framework using both the GPT-4o model and the Lakmoos system. Lastly, I created a final comparison with a longer conversation between the three systems compared.

6.1.2 Free Discussion Evaluation Results

This section provides the results of the experiments using the evaluation dimensions. The full tables of the experiments are provided in Appendix B.

The comparison between different architecture design versions for the proposed framework is provided in Table 6.3 and Table 6.4.

Architecture Design	BEL	KNO	CRE	TTF
PerSimChat MF	$\mathbf{8.79 \pm 0.74}$	4.70 ± 1.39	$\mathbf{8.80 \pm 0.61}$	$\mathbf{8.57 \pm 0.63}$
PerSimChat Base	8.53 ± 0.94	4.27 ± 1.55	8.63 ± 0.72	8.47 ± 0.63
PerSimChat NoPR	8.52 ± 1.06	5.53 ± 1.63	8.47 ± 1.72	8.40 ± 0.67
PerSimChat MiS	8.68 ± 0.72	5.13 ± 1.72	8.53 ± 0.78	8.23 ± 0.63
PerSimChat MiNM	8.23 ± 1.19	$\mathbf{5.60 \pm 1.43}$	7.97 ± 1.67	8.10 ± 0.66
PerSimChat LakMod	8.50 ± 0.75	5.43 ± 1.63	8.20 ± 1.77	7.97 ± 0.61

Table 6.3: Free discussion dimensions evaluation results for the different PerSimChat architecture designs. *BEL* stands for believability dimension, *KNO* for knowledge, *CRE* for credibility, and *TTF* for turn-taking and flow. For PerSimChat version, *MF* stands for memory filtering, *Base* for the baseline version used in most of the tests, *NoPR* without planning and reflection, *MiS* and *MiNM* stand for Mistral Small and Mistral NeMo models, and *LakMod* for the proprietary Lakmoos model.

Based on the results, incorporating *memory filtering* while some of the memories are not stored in long-term memory leads to a higher score in 6 of 7 evaluated dimensions. Even though some of the memories could be wrongly labeled by a model as unimportant

Architecture Design	CDR	SER	GPTS
PerSimChat MF	$\mathbf{8.00 \pm 0.74}$	5.43 ± 1.04	$\mathbf{8.93 \pm 0.52}$
PerSimChat Base	7.93 ± 1.11	5.57 ± 1.07	8.90 ± 0.80
PerSimChat NoPR	7.60 ± 2.22	5.60 ± 2.03	8.73 ± 0.83
PerSimChat MiS	7.43 ± 0.97	6.00 ± 1.08	$\mathbf{8.93 \pm 0.58}$
PerSimChat MiNM	7.73 ± 1.68	6.27 ± 0.94	8.77 ± 0.81
PerSimChat LakMod	7.17 ± 2.55	$\mathbf{6.50 \pm 1.48}$	8.83 ± 0.83

Table 6.4: Evaluation of the free discussion for the remaining dimensions, and so *CDR* for content depth and relevance, *SER* for social and emotional responsiveness, and *GPTS* for goal progression or task solving. For PerSimChat version, *MF* stands for memory filtering, *Base* for the baseline version used in most of the tests, *NoPR* without planning and reflection, *MiS* and *MiNM* stand for Mistral Small and Mistral NeMo models, and *LakMod* for the proprietary Lakmoos model.

and removed, the overall performance could remove the distractions and lead to a more natural conversation.

Except the knowledge dimensions incorporating *planning and reflection* for the agent improve the system's performance in most evaluated dimensions, namely believability, credibility, turn-taking and flow, content depth and relevance, and goal progression or task solving.

Comparing the different models in the PerSimChat framework, the GPT-4o model excels in credibility, turn-taking and flow, and content depth and relevance dimensions. The Mistral Small model outperforms in believability and goal progression or task-solving dimensions. The Mistral NeMo model performs well in knowledge, and the Lakmoos model performs best in the social and emotional responsiveness dimension while obtaining satisfactory results in other dimensions.

The conversation closure is not important in this comparison, because the lack of conversation closure comes out of the system's design, as the agents are not well informed about when the conversation ends.

The second comparison is focused on *social goals* and *relationship* incorporation. These results are shown in Table 6.5.

Method	GOAL	BEL	REL	CDR
Single-Agent Zero-Shot	8.90 ± 0.80	8.90 ± 0.66	3.70 ± 0.84	7.93 ± 0.74
AutoGen	$\mathbf{9.27 \pm 0.52}$	8.86 ± 0.74	$\mathbf{4.20 \pm 0.85}$	7.63 ± 1.97
PerSimChat SGR	8.97 ± 0.96	$\mathbf{8.93 \pm 0.80}$	3.83 ± 0.79	$\mathbf{8.07 \pm 0.94}$

Table 6.5: Evaluation of the PerSimChat framework with two baseline solutions — Single-Agent and AutoGen when including the social goals and predefined relationships (and therefore PerSimChat SGR design version). *GOAL* dimension stands for goal completition, *BEL* for believability, *REL* for relationship, and *CDR* for content depth and relevance.

The AutoGen baseline solution provides the best results in the goal completion dimension. With that, the proposed solution outperforms the single-model solution. Also, similar results can be seen for the relationship dimensions as well. The main reason for such behavior could be that the complex focus of the PerSimChat framework on the naturalness of the conversation can sometimes overload the model. So, it is harder for the model in such a complex system to always hold the predefined social goals and relationships. The PerSimChat framework in this experimental scenario outperforms in the believability and content depth and relevance dimensions, while the worst results are provided by the AutoGen solution.

Using the Lakmoos personas data, the comparison results between the baselines with the PerSimChat framework using the GPT-4o model and PerSimChat with the Lakmoos system are provided in Table 6.6.

Method	BEL	CRE	TTF	CDR
Single-Agent Zero-Shot	8.36 ± 1.01	8.30 ± 0.95	$\mathbf{8.70 \pm 0.53}$	$\mathbf{7.83 \pm 0.83}$
AutoGen	8.62 ± 0.96	8.73 ± 1.68	8.60 ± 0.83	7.33 ± 1.79
PerSimChat Base	8.67 ± 0.88	8.87 ± 0.43	8.67 ± 0.49	7.67 ± 1.30
PerSimChat LakSys	$\mathbf{8.79 \pm 0.86}$	$\mathbf{9.00 \pm 0.64}$	8.67 ± 0.55	$\mathbf{7.83 \pm 1.21}$

Table 6.6: Comparison of two versions of the PerSimChat framework (GPT-4o model and Lakmoos system) with the baselines (both with GPT-4o model) — Single-Agent and AutoGen. For the dimensions, *BEL* represents believability, *CRE* credibility, *TTF* turn-taking and flow, and *CDR* content depth and relevance.

For the provided dimensions out of 8 dimensions measured, incorporating the Lakmoos system improves the performance of the proposed framework in all four dimensions. At the same time, it performs best in the believability, credibility, and content depth and relevance dimensions. For the turn-taking and flow dimension, the single model solution performs slightly better than PerSimChat, and the AutoGen baseline performs the worst.

The last and the widest comparison in terms of experimental rounds and conversation length is the final comparison of the PerSimChat framework with the two baseline solutions. Table 6.7 provides the most important results.

Method	BEL	CRE	TTF	CDR
Single-Agent Zero-Shot	8.65 ± 0.82	7.96 ± 0.88	$\mathbf{8.75 \pm 0.53}$	8.40 ± 0.81
AutoGen	8.67 ± 0.79	8.69 ± 1.51	8.40 ± 1.49	8.15 ± 2.13
PerSimChat Base	$\mathbf{8.76 \pm 0.59}$	$\mathbf{8.78 \pm 0.52}$	8.58 ± 0.59	$\mathbf{8.51 \pm 0.84}$

Table 6.7: The final dimensions evaluation of the PerSimChat framework compared to Single-Agent and AutoGen baselines. *BEL* stands for the believability dimension, *CRE* for credibility, *TTF* for turn-taking and flow, and lastly *CDR* the content depth and relevance. For the proposed system, the baseline solution is used, while all methods use the GPT-4o model.

The proposed PerSimChat framework outperforms in three of four dimensions: believability, credibility, and content depth and relevance, and so in the most essential dimensions for the framework's purposes. In these dimensions, the worst results have the single-agent solution in believability and credibility, and the AutoGen baseline in the content depth and relevance. The single-agent zero-shot system provides the best results for turn-taking and flow dimensions, while PerSimChat stands in second place. These results reflect the nature of the systems because the positive aspect of the single-agent system is the linking of the conversation.

6.2 Scoring Free Discussion with FairEval

Although the free discussion was already compared in the previous experimental scenarios, only a single method was evaluated at a time. To obtain richer insights about the performance of individual systems, I also compared the results in a pairwise comparison between all three frameworks. Since direct comparison with only one LLM suffers from several shortcomings, I use the approach adopted from the FairEval [103] publication. In this experiment, I compare the *realism* of the methods' results side-by-side. The notation and experimental setups are the same as for the dimensions evaluation, see Section 6.1.

6.2.1 Experimental Setup and Evaluation Process

For the evaluation run, the systems' outputs must be post-processed to obtain the same output format from each compared solution and achieve the most reliable experimental results. The format is the same as mentioned in Section 6.1.

Following the FairEval [103] architecture, for each run, the evaluator scores both systems within the range 1–10 based on the quality of the output. Based on that, the scores determine whether the first system *wins*, *loses*, or if there is a *tie* between the two systems' outputs. The percentages of wins, ties, and losses to the total number of compared tasks are reported for each comparison.

Similarly to Section 6.1, this experiment uses the same dataset of 100 tasks. For comparisons between the PerSimChat framework architecture versions and the comparison with the Lakmoos system, the results provided are obtained from a single run over the same subset of 30 tasks with 10 messages exchanged during conversation. This setup was chosen based on the computational costs. For the final comparison of the base PerSimChat version with the two baselines, each system runs within three rounds with the same subset of 15 samples, which is changed for each run. The number of messages in this experiment is set to 20. Each run has three agents. I use the *gpt-4o-2024-08-06* model for all agents, and for the FairEval evaluator, the *gpt-4-turbo-2024-04-09* model. The outputs of the systems are post-processed to obtain the same output format. For each method, the percentage of wins, ties, and losses is reported using mean and standard deviation. The notation of the PerSimChat framework versions stands for the versions described in Section 6.1.

6.2.2 FairEval Scoring Results

First, I compared the different architecture versions of the proposed framework. The results obtained are shown in Table 6.8. In all experiments, the same generated personas from scratch are used.

From the results, it can be deduced that for the FairEval, the framework's performance is better when the memory filtering is off. This can be caused by the fact that some essential memories are chosen from the large language model as unimportant and are not saved in long-term memory. However, from Section 6.1 we can see that the memory filtering positively influences some of the evaluation dimensions.

Incorporating the planning and reflection of the agent improves the group's performance during the evaluations, which confirms the results from the dimensions experiments. When comparing the different models, the GPT-4o model outperforms the Mistral NeMo model, while Mistral Small outperforms GPT-4o. By incorporating the Lakmoos model into the PerSimChat system, the Lakmoos model beats all three mentioned models.

Method 1	Method 2	Wins (%)	Ties (%)	Losses (%)
PerSimChat MF	PerSimChat Base	24.14	31.03	**44.83**
PerSimChat Base	PerSimChat NoPR	**43.33**	20.00	36.67
PerSimChat Base	PerSimChat MiS	33.33	26.67	**40.00**
PerSimChat Base	PerSimChat MiNM	**46.67**	16.67	36.67
PerSimChat MiS	PerSimChat MiNM	**42.86**	25.00	32.14
PerSimChat Base	PerSimChat LakMod	40.00	16.67	**43.33**
PerSimChat MiS	PerSimChat LakMod	34.48	6.90	**58.62**
PerSimChat MiNM	PerSimChat LakMod	42.86	10.71	**46.43**

Table 6.8: The pairwise comparison of the PerSimChat architecture designs using the FairEval evaluator. The percentages of wins, ties, and losses of the first method over the second are provided for each pair. The *MF* stands for memory filtering, *Base* for the baseline version used in most of the tests, *NoPR* without planning and reflection, *MiS* and *MiNM* stand for Mistral Small and Mistral NeMo models, and *LakMod* for the proprietary Lakmoos model for the message text generation.

Secondly, the system is compared to the baseline solutions, and with the incorporation of the Lakmoos system for message generation. For this experiment, I used the same three personas for all tools, using converted Lakmoos data, except the *PerSimChat Base*, which uses the generated ones. The results are listed in Table 6.9.

Method 1	Method 2	Wins (%)	Ties (%)	Losses (%)
PerSimChat Base	PerSimChat LakData	26.67	33.33	**40.00**
PerSimChat LakData	Single-Agent Zero-Shot	33.33	26.67	**40.0**
PerSimChat LakData	AutoGen	**36.67**	30.0	33.33
PerSimChat LakData	PerSimChat LakSys	31.03	17.24	**51.72**
Single-Agent Zero-Shot	PerSimChat LakSys	43.33	10.0	**46.67**
AutoGen	PerSimChat LakSys	23.33	**46.67**	30.0

Table 6.9: Pairwise evaluation of the PerSimChat framework (*Base*, *LakData*, and *LakSys* versions) with two baseline solutions: Single-Agent and AutoGen. I provide the percentage of wins, ties, and losses for each pair. The *Base* version stands for PerSimChat's baseline version used in most of the tests, *LakData*, which incorporates converted personas from Lakmoos data instead of from scratch generation, and *LakSys* for the proprietary complete Lakmoos system for the message text generation.

For comparison between the data generated from scratch and that converted from the Lakmoos data, the second version outperforms the generated data. This indicates the quality of the data obtained and the greater credibility of the personas.

With comparing the winning system using the GPT-4o model with the baseline solutions, the PerSimChat framework wins over the AutoGen version, however it lacks the performance of the single-agent solution. This can be caused by the fact that the single model in such a small experimental example better connects the thoughts of individual agents during a conversation. The Lakmoos system used in the proposed system beats the GPT-4o PerSimChat version and both the baseline solutions.

Lastly, the final and more inclusive experimental evaluation was performed. Similarly to the first experiment, the same generated personas from scratch are used. The Table 6.10 shows the final results.

Method 1	Method 2	Wins (%)	Ties (%)	Losses (%)
Single-Agent Zero-Shot	AutoGen	40.00	8.89	**51.11**
PerSimChat Base	Single-Agent Zero-Shot	31.82	27.27	**40.91**
PerSimChat Base	AutoGen	**38.64**	25.00	36.36

Table 6.10: Final pairwise evaluation of the PerSimChat baseline version with two other solutions: Single-Agent and implementation using the AutoGen open-source framework. For each test, I provide the win, tie, and loss rate of the first method over the second method.

In this experiment, the AutoGen baseline solution outperforms the Single-Agent Zero-Shot solution. Comparing the proposed system with the AutoGen solution, while slightly better performing, I present these solutions in the FairEval experiments as comparable. On the other hand, the single-agent solution performs better than a PerSimChat framework. The results can be confusing because there is no clear performance ranking among the three tested systems. Although the AutoGen library beats the single agent solution, and the AutoGen and PerSimChat framework performance is similar, the single agent beats the PerSimChat framework solution in this comparison. Given these results, I attach importance to the fact that even the FairEval evaluation prevents the favoring of one system over another based on the order in which their results are presented, the evaluation model during these experiments assigns weight to different content of the text during evaluation. This means that while in some experiments the evaluator focuses on the flow of conversation, while comparing the other pair with the different system, it can, for example, focus on the naturality of the text. Even though the results are converted into the same format, the text provided in the messages looks quite different based on which system generated such a conversation, and it can cause different evaluator weighting for the importance of characteristics.

6.3 Comparison of Group Debate with Multi-Agent Debate Systems

From the other point of view, with a focus on group debate, the proposed PerSimChat system can also solve real-world problems. Knowing the correct answer, the framework can be evaluated as a whole while measuring its accuracy on predefined tasks. Based on that, I can compare my system with the other multi-agent debate (MAD) methods.

Despite that, the main use case of the PerSimChat framework is to simulate a believable human conversation. If the system passes the benchmarks, it may also indicate the reasoning capacities of the group debate and the whole system.

6.3.1 Experimental Setup and Evaluation Process

Following the work [17], for the evaluation process, I use four chosen benchmark datasets, namely two commonsense (StrateqyQA [37] and ECQA [2]) and two math (GSM8K [20] and AQuA [64]).

For comparison, I selected five systems — three MAD (MAD [26], MAD with Judge [63], and ReConcile [17]) and the two single large language model (LLM) solutions, Single-Agent Zero-Shot and Single-Agent Zero-Shot CoT, which represents the chain of thought prompting technique.

As a reference model, based on reasoning capabilities and pricing, I selected the *gpt-4o-2024-08-06* model for all agents and judges as well.

I run every compared system in three runs, primarily accounting for the variance caused by the decoding strategy. Due to computational costs in each run, each system was tested on a subset of 30 tasks. The subset differs between runs, but is the same for each system in a single run. For experiments, the mean accuracy and the standard deviation are reported.

For the PerSimChat framework, for each benchmark, I prompted the ChatGPT[10] GPT-4o model (version *gpt-4o-2024-08-06*) to propose the best three personas for each benchmark to hit the best knowledge levels in the message generation prompt. Based on the profiles, I let it generate the personas' descriptions in the required format. For the simulation judge agent, the base sliding window version was chosen.

The simulation runs are configured with an asynchronous algorithm version using *planning and reflection*, *emotional state retrieval*, and *short-term memory retrieval*. I selected the *need to talk* pattern with the *maximum likelihood* version for this experiment. All other parameters are set to false.

6.3.2 Group Debate Evaluation Results

The summary results of all three runs are shown in Table 6.11.

Method	StrateqyQA	ECQA	GSM8k	Aqua
Single-Agent Zero-Shot	82.2 ± 3.9	81.1 ± 8.4	46.7 ± 3.3	46.7 ± 6.7
Single-Agent Zero-Shot CoT	$\mathbf{86.7 \pm 3.3}$	77.8 ± 1.9	90.0 ± 6.7	78.9 ± 1.9
MAD	81.1 ± 7.7	$\mathbf{83.3 \pm 3.3}$	$\mathbf{91.1 \pm 1.9}$	$\mathbf{85.6 \pm 1.9}$
MAD with Judge	70.0 ± 12.0	76.7 ± 6.7	87.8 ± 6.9	83.3 ± 3.3
ReConcile	84.4 ± 5.1	68.9 ± 1.9	90.0 ± 3.3	82.2 ± 5.1
PerSimChat	71.7 ± 9.6	60.0 ± 6.7	88.9 ± 5.1	77.8 ± 13.5

Table 6.11: Group debate evaluation of the PerSimChat framework with five other solutions, namely two single-agent (Single-Agent Zero-Shot and chain-of-thought version, Single-Agent Zero-Shot CoT) and three multi-agent debate (MAD) solutions: MAD, MAD with Judge, and ReConcile. All these systems were evaluated on four benchmarks, two commonsense (*StrategyQA* and *ECQA*) and two math (*GSM8k* and *Aqua*).

Despite the primary use cases of the PerSimChat framework for the StrategyQA benchmark, it outperforms the MAD with Judge system. From the MAD style system, the ReConcile framework performs the best while lagging behind the single-agent zero-shot chain-of-thought version. Overall, only the ReConcile MAD system outperforms the base single-agent zero-shot version.

The ECQA dataset tasks are quite demanding for my system. Compared to the StrategyQA commonsense benchmark, which requires a binary response of true or false, the ECQA dataset lets the system choose from five options available (multiple-choice). This can cause PerSimChat to underperform in this task. However, it cannot be defined whether the

[10] https://chatgpt.com/, accessed: 2025-04-30.

performance is caused by the system implementation itself or by the nature of the group conversation, and by the fact that the group debate is not always effective for the best results. In this benchmark, the single model without the chain-of-thought version outperforms the version with this technique. From the MAD frameworks, only the MAD framework beats the single-model implementations, while MAD with Judge beats the ReConcile system.

When looking at a math problem, on the GSM8k dataset, PerSimChat outperforms MAD with Judge architecture. When comparing the MAD systems with a single-agent zero-shot chain of thought, the MAD and ReConcile frameworks beat a single model, while MAD performs better than ReConcile. The single-agent zero-shot version without the chain-of-thought technique underperforms in this task.

With the Aqua benchmark, the PerSimChat performance is better than a single-agent zero-shot version and slightly lags behind the chain-of-thought version. They perform better on this task than MAD systems, while the best performance is measured for MAD. The MAD with Judge outperforms the ReConcile framework.

Various factors can cause a lack of performance compared to the single-agent zero-shot chain of thought. I give the most significant weight to the fact that the model, besides the mere inference about the correct solution during the simulation and message generation, must emphasize the constant construction of the persona's personality. This places greater demands on the model and reduces its overall performance.

6.4 Human Evaluation of Free Discussion

Despite the possibilities of replacing human crowdworkers with LLM evaluators [13, 55, 103, 118], nowadays, human insights are still valuable and irreplaceable. Simultaneously, the models suffer from numerous problems, as described in Section 2.3.3. Based on the LLM characteristics, it is not guaranteed that data bias will not be reflected in evaluations. To provide a broader context, I also conducted a survey with people using a questionnaire.

6.4.1 Experimental Setup and Evaluation Process

The experiment took place in the form of a survey created using Google Forms[11] platform.

In the evaluation process, participants were asked to complete a questionnaire consisting of AI-generated communications from three different systems. In total, for these systems, the four different initial conversation topics were chosen:

- *What are the biggest pros and cons of working remotely?*
- *What are the most effective ways to deal with stress?*
- *Imagine you're part of a Mars colony — what challenges would you discuss as a group?*
- *How do language and cultural barriers affect the way people communicate and form relationships in multicultural societies?*

This corresponds to a total of 12 conversation ratings. Such tasks represent topics of *generic* and *knowledge* type and were taken from the created dataset as described in Section 6.1, which partially consists of the FairEval [103] questions. With these tasks,

[11] https://workspace.google.com/products/forms/, accessed: 2025-05-10.

I represent a broader range of possibilities in terms of the topic and complexity of the question.

For simulations, I selected the four runs of PerSimChat *Base*, AutoGen, and Single-Agent Zero-Shot systems. The notation and systems used correspond to Section 6.1 (namely, Table 6.1 and Table 6.2). From each run, I took the first five messages to avoid overwhelming the evaluator. The order of these frameworks' outputs was random for each conversation topic, and the system labels do not correspond to the frameworks themselves (e.g., *system A* was a different system's output between these sections). Participants were tasked to rate each system on a 10-point Likert scale[12] (so in a range of 1-10) with the description of the evaluation task, focused on naturality, realism, and believability, as shown in Listing 6.1.

```
In this task, you will assess how natural, realistic, and believable the agent
   appears during the conversation. Your judgment should consider whether the
   agent behaves in a way that is consistent with how a real human with that
   character/personality might speak or act.

Score Range, Meaning
1-3 Limited Realism: Agent feels robotic, generic, or inconsistent. Hard to
   believe as a real character.
4-6 Moderate Believability: Some moments feel real, others don't. Mixed signals.
7-8 Highly Credible: Mostly believable, with consistent and realistic behavior.
9-10 Human-like Believability: Feels like a real person. Seamless, authentic
   interaction throughout.
```

Listing 6.1: The description of the evaluation goal for the participants. This part was provided for each section (and corresponding conversation topic) before the three rated systems.

At the end of the questionnaire, the crowdworkers are tasked with two open-ended questions:

- *What did you find realistic?*
- *What did you find less believable and could be improved?*

These tasks were chosen to ask for the positives and negatives of all three systems tested at once.

The evaluators were citizens of the Czech Republic and Slovakia, mostly with an adequate level of English, and older than 18 years old. In total, I recruited 25 evaluators. The biggest subgroups of this part of the population consist of students (bachelor's degree program or higher) and college graduates. Specifically, the number of men prevailed over women or non-binary individuals. The oldest participants tasked were in the range of 50–60 years old. More than half of the participants are people working or studying in the field of information technology and AI, but the rest are people working in other fields (textile industry, logistics, etc.). Completing the questionnaire was voluntary and without any financial benefit. The participation lasted around 20 minutes.

[12] https://www.surveymonkey.com/mp/likert-scale/, accessed: 2025-05-10.

6.4.2 Human Evaluation Results

The results of the human evaluation are shown in Figure 6.1. For all the participants' ratings, I calculated the *Krippendorff's alpha*[13], $\alpha = 0.3674$, which reflects the low level of agreement among multiple evaluators. This means that the participants only slightly agree beyond chance and that there is a lot of variability in the ratings.

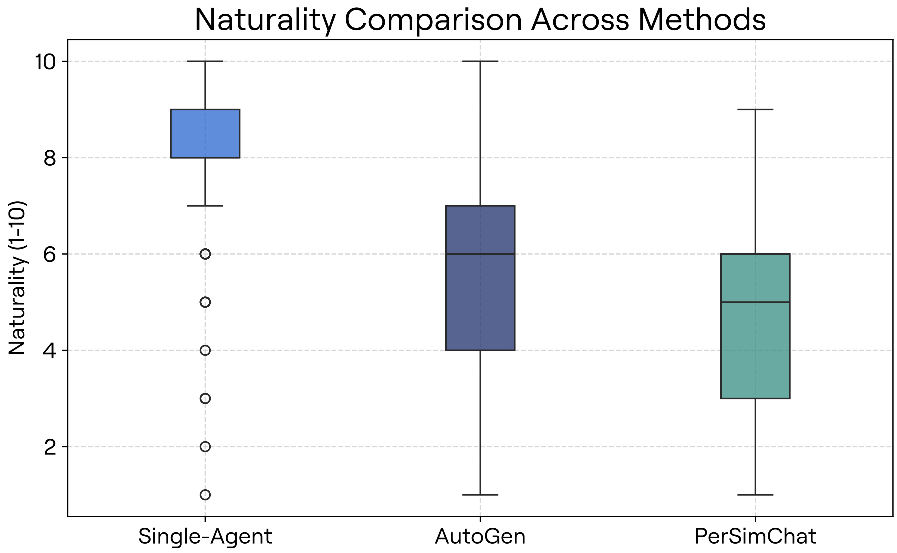

Figure 6.1: The comparison between the system-specific methods, including the proposed PerSimChat framework (with *Base* architecture design) and the two baseline solutions, Single-Agent (Zero-Shot) and AutoGen. The ratings by the human evaluators were focused on the naturality, realism, and believability of the provided conversations. The data shows ratings for four conversation topics for all three systems. The ratings use the 10-point Likert scale (range 1–10).

The results show that the crowdworkers significantly prefer the outputs of the single LLM (*Single-Agent*), which generates the entire conversation in one step. Simultaneously, the lowest scores for the AutoGen and proposed PerSimChat frameworks are the same. The highest score for the AutoGen baseline is assigned only by a single evaluation participant. Based on median values and quartiles, the data show a smaller advantage for the AutoGen solution over PerSimChat. However, using the *Wilcoxon signed-rank test*[14], the hypothesis that AutoGen performs significantly better than PerSimChat is not supported[15], with a *p-value* $= 0.54$. Based on that, there is not enough evidence to say that AutoGen is better than PerSimChat in this test scenario.

Based on the open-question answers, it is clearly observable that the participants evaluate the output by a single LLM better based on its shorter answers over the amount of information provided by multi-agent frameworks. This can be seen from the positive answers (*„The shorter conversations felt more real.“*, *„I prefer short answers and casual words.“*, and *„Shorter answers are, in my opinion, more realistic, since they keep the conversation*

[13] https://www.surgehq.ai/blog/inter-rater-reliability-metrics-an-introduction-to-krippendorffs-alpha, accessed: 2025-05-10.
[14] https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/how-to-conduct-the-wilcox-sign-test/, accessed: 2025-05-10.
[15] The significance level was set to $\alpha = 0.05$.

more engaging.“) as well as from the negative answers (*„The longer passages seemed to use overly sophisticated language.*“, *„Very long general answers with phrases you would not normally use.*“, and *„I think the long comprehensive answers don't feel very natural.*“).

Also, some of the crowdworkers prefer the slang used by a single model (*„The shorter answers with slang were more believable.*“ and *„I think the short sentences and less formal language are what I found more realistic.*“) over the more formal sentences used by multi-agent systems (*„People are too formal.*“, *„Some systems heavily use corporate speak.*“, and *„Two systems used far too much formal language, and the sentences were unnecessarily long.*“).

In addition, positive feedback is on common and short words (*„In my opinion, colloquial expressions are also a sign of naturalness in human conversation.*“), and external perceptions (*„For example, when the AI was laughing.*“). It is also remarkable that some of the participants think that the multi-agent systems outputs were generated by ChatGPT[16] platform instead of a single GPT-4o model solution, which is nowadays used in this tool (*„The chatgpt feel of some conversations, like I would like to see more like you are writing with friends in a random chat group.*“).

For the content of each individual message, the participants' answers differ about the agents' background with a positive point of view (*„Nice ideas and everybody feels like they have background and are just making stuff up.*“, *„That everyone has their own personality.*“, and *„How the people talk about their personal experiences.*“) and also negative (*„When the clones start to force their profile info into the conversations.*“). With that, the more complex and technical words lower the rating of multi-agent systems (*„A lot of technical terms.*“).

Some of the crowdworkers also focus on the difference between each of the personas represented (*„Every character talks the same way just about different hobbies.*“ and *„Some conversations have the same pattern of contributions.*“) and the amount of agreement and disagreement (*„Also, conflict or disagreement can feel more natural for certain topics.*“) and repeating of the words in the different agents' messages (*„Repeating the common phrases in more messages.*“).

Finally, the flow of the conversation affects the ratings (*„The conversation flow. I mostly liked that the agents asked each other questions and the quick response of the interviewed agent.*“ and *„Conversation having an introduction, agents reacting with each other. Asking questions and reflecting on the answers.*“).

The main differences between the ratings are primarily caused by the length of the individual agents' messages and the type of language used (colloquial or formal). Of 25 respondents, 88% (22 participants) preferred short messages (about 25 words per message on average) with informal vocabulary. The longer messages were, on average, 47 words long by AutoGen and 42 words long by PerSimChat.

However, there is a difference in the requirements for these systems. The proposed PerSimChat framework is also designed to be used in business use cases where formal language is mostly preferable over the use of slang expressions in messages. Based on this, the excessive use of colloquial language is contrary to the use cases of the system by the external partner.

[16] https://chatgpt.com/, accessed: 2025-05-10.

6.5 Evaluation of Specific Use Cases

Lastly, to add concrete examples of the discussion evaluation to the previous experiments, I provide analyses and a discussion of the specific runs of the PerSimChat framework in this section. I highlight the most essential parts of the strengths and limitations of the proposed system.

6.5.1 Experimental Setup

The tasks shown during the experiments are taken from the custom dataset, partially created from the FairEval [103] questions. This dataset contains 32 topics from multiple categories: *knowledge*, *coding*, and *math*. The system was tested on five different tasks:

- *Let's compare Newton's laws of physics and Einstein's theory of relativity.*
- *Write a short poem about the fall season.*
- *Radek Vávra has already bought a new smartphone. Let's write a conversation where Radek is trying to convince others how great the phone is and that they should buy it too.*
- *Does Technology Make Us More Connected or More Isolated?*
- *Why might someone prefer to shop at a small, locally-owned business instead of a large chain store, even if the prices are higher?*

These specific requirements and questions were chosen for multiple reasons. The first task includes a complex topic like the theory of relativity, and should show that the models are not overloaded with information and follow the required knowledge levels. The second task contains a complex wish of writing a poem, which sometimes the system does not meet the target later in the conversation, especially on very creative topics. The remaining two questions represent a *knowledge* and *commonsense* topics, respectively, and were randomly chosen from these two subgroups of the dataset. These tasks should force the agents to be creative in the message generation, sometimes also opposing each other, and to show the frequent system behavior, like content depth or coherence in the conversation.

For each task, I performed two runs. Every simulation includes the same three personas and allows them to converse for 20 messages. The model chosen for each agent is the *gpt-4o-2024-08-06*. For architecture design settings, I used the asynchronous version of PerSimChat with *planning and reflection* while using *emotional state retrieval* and *short-term memory retrieval*. For free discussion, the need to talk pattern with maximum likelihood version is used while not allowing the speaker repetition in a row. For all the experiments, the same three personas are used for the dimensions evaluation and FairEval evaluation, except for the knowledge level task, where a single persona is replaced by a persona representing a 5-year-old girl.

6.5.2 Use Cases Evaluation Results

In this section, I highlight some situations during the experimental runs that show the pros and cons of the PerSimChat framework.

✔ Personas Follow the Knowledge Levels

The first task incorporates a complex topic, such as the difference between laws of physics, which should be beyond young children's knowledge. In this experiment, we can see in Figure 6.2 that the 5-year-old participant (*Lucie Křížková*) follows the knowledge levels defined in the message generation prompt and responds adequately to her knowledge and experience. In addition, older participants add context and expertise to their age and education.

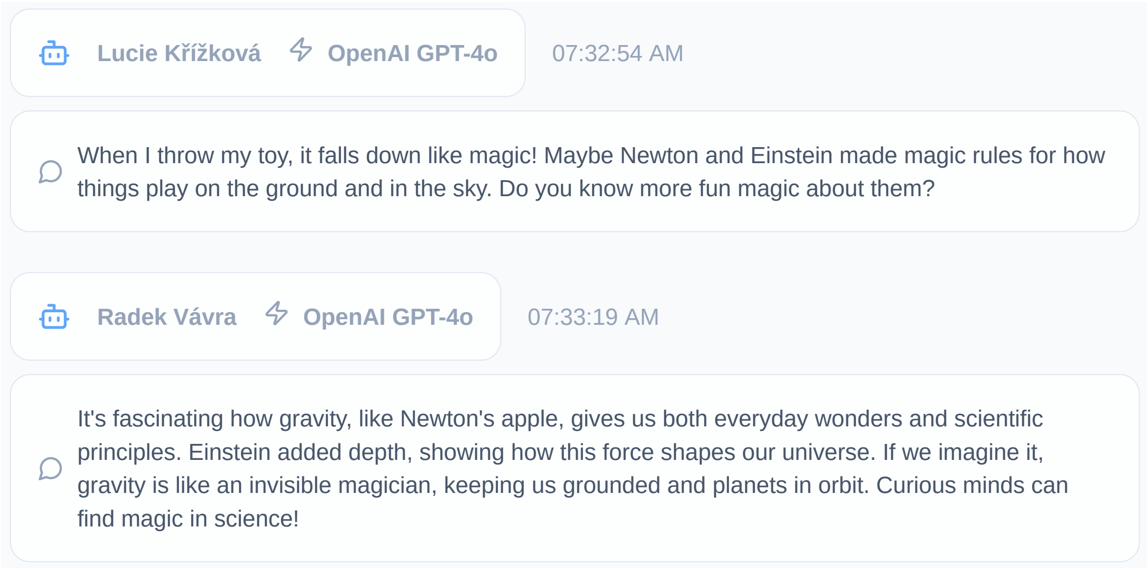

Figure 6.2: Use case study with a focus on compliance with knowledge levels where *Lucie Křížková* is 5 years old and *Radek Vávra* is a train conductor with a technical high school education.

✔ Agents Know About the Other Discussion Participants and Ask Them Questions

The critical part of the agent in the discussion is considering the other participants and their opinions. Especially in Figure 6.3, Figure 6.4, and Figure 6.5, the agents consider other participants and call for a response when asking for their opinions.

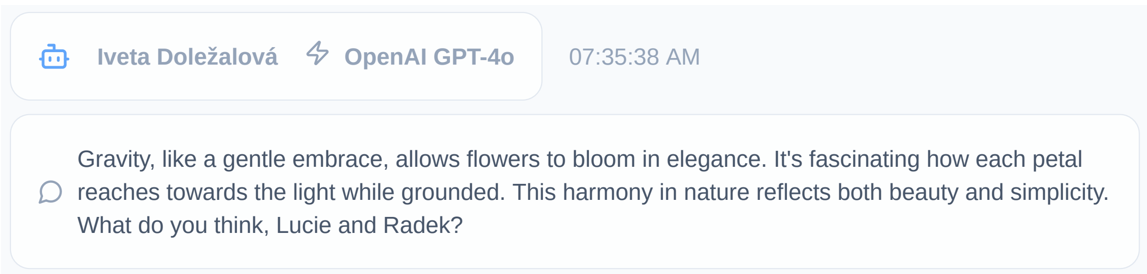

Figure 6.3: The simulation run where *Iveta Doležalová* directly asks a question to the other two participants in the conversation (*Lucie* and *Radek*).

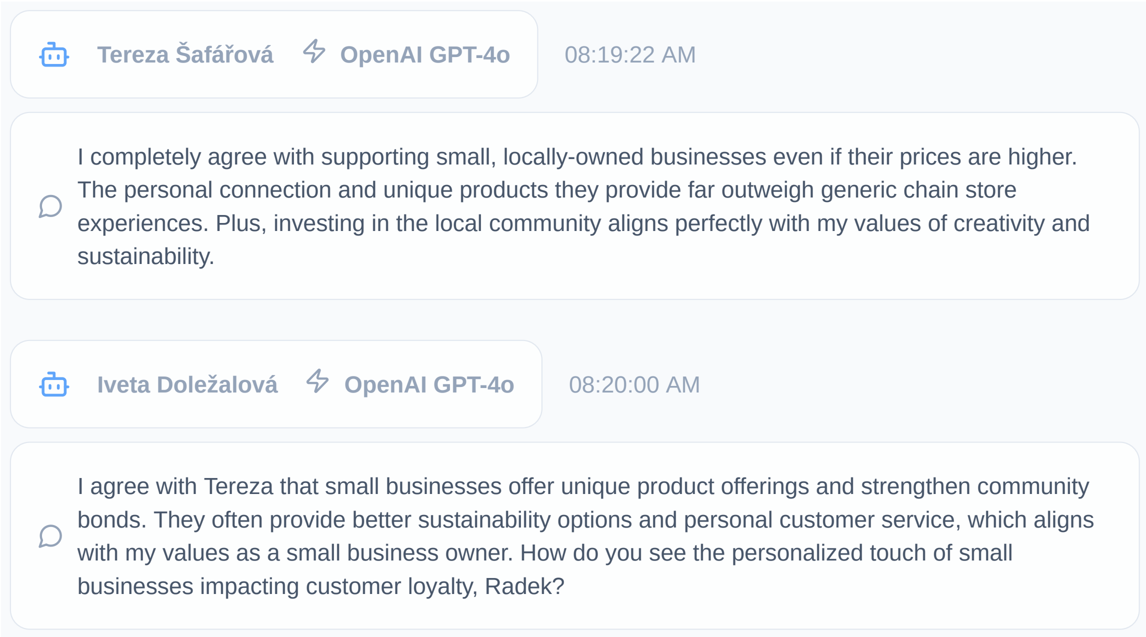

Figure 6.4: The part of communication to be continued in Figure 6.5 where the discussion participants refer directly to specific individuals.

✘ Personas Sometimes Repeat the Same Thoughts

Similarly to the SOTOPIA environment [118], I also observed the repetitions of thoughts during subsequent conversations. This is captured in Figure 6.6 and Figure 6.7 when a single agent repeats the same information during the discussion.

✘ Agents Sometimes Do Not Maintain the Context of the Assignment

During some more complex tasks where we need them to be creative and propose a final solution, the agents do not maintain the objective of the conversation and instead start discussing the topic of the problem. This can be seen, for example, in Figure 6.8.

✔ Agents Oppose Each Other or Are Skeptical

To achieve diversity in conversations, it is also necessary, in addition to the agreement of all agents on the opinion, to have their opposition and a critical view of the problem. With that, healthy skepticism is also appropriate. This behavior is shown in Figure 6.9 and Figure 6.10.

✔ Personas Mention Their Previous Experiences

During the conversation, it is also natural to point to characteristics typical of a given person, such as their status or history of experience. Based on that, the proposed framework provides these properties during conversations, as shown in Figure 6.11 and in Figures 6.9 and 6.10.

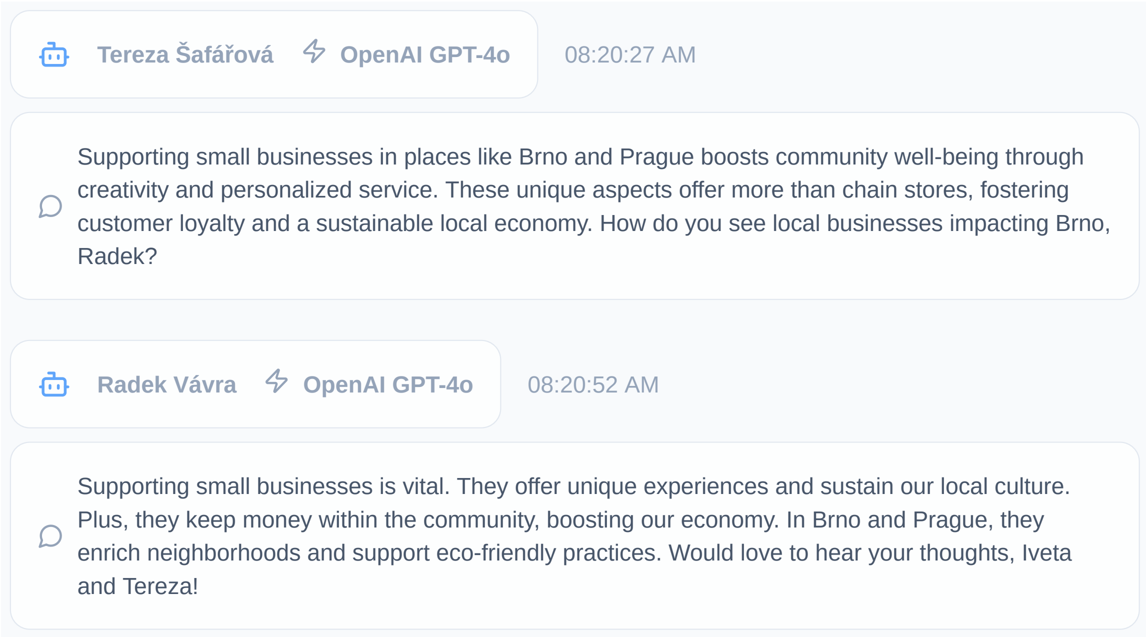

Figure 6.5: The part of communication which continues conversation from Figure 6.4 where the discussion participants refer directly to specific individuals.

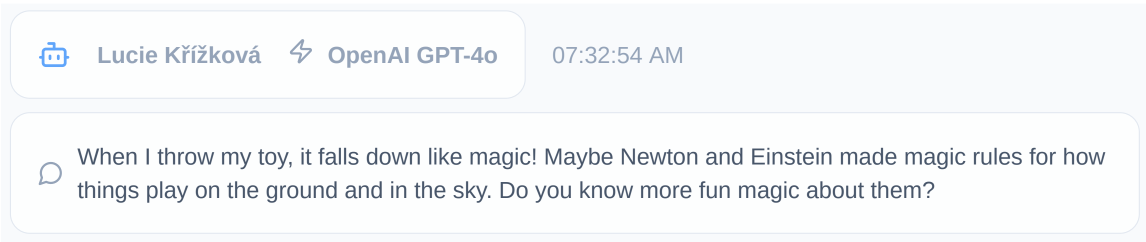

Figure 6.6: The use case study of the PerSimChat framework in which the problem with repeating the same thoughts is shown. Figure 6.7 provides the repetition in the same simulation run.

✔ Agents Are Creative in Message Generation

Creativity is one of the main factors of human behavior and existence, and should also be included in the discussion. PerSimChat also offers, during the conversation, some of the creative ways of how the agents generate the message, for example, in Figure 6.12.

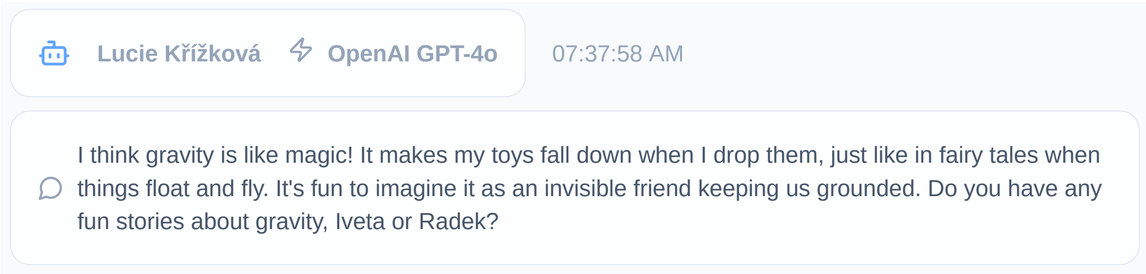

Figure 6.7: The use case study of the PerSimChat framework in which the problem with repeating the same thoughts is shown. The first mention of the thought is provided in Figure 6.6.

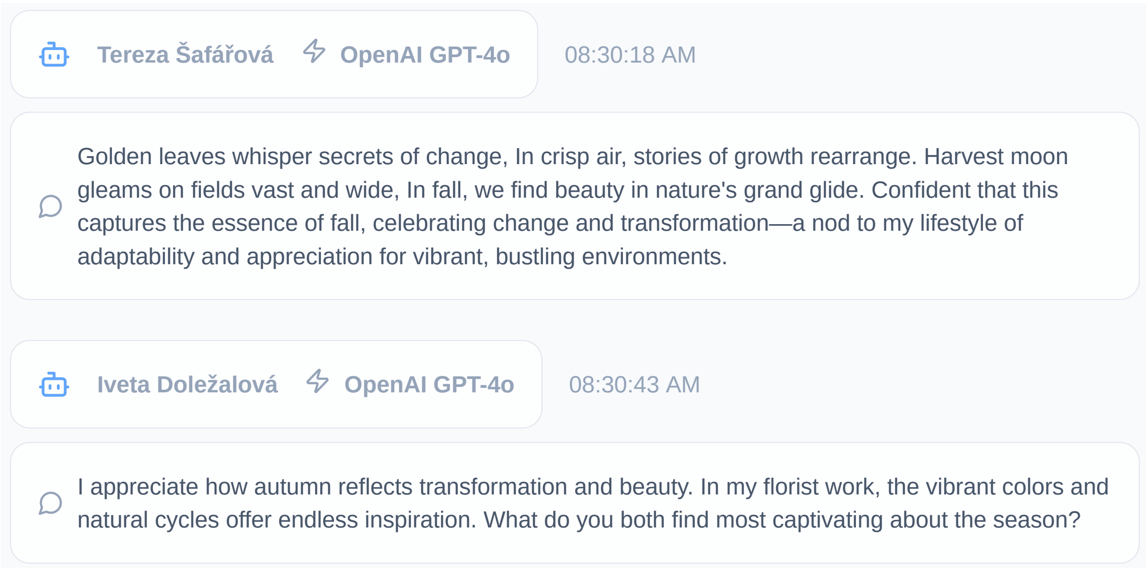

Figure 6.8: The example of a simulation run where the agents, instead of building the final solution of the poem about the fall season, discuss the topic instead.

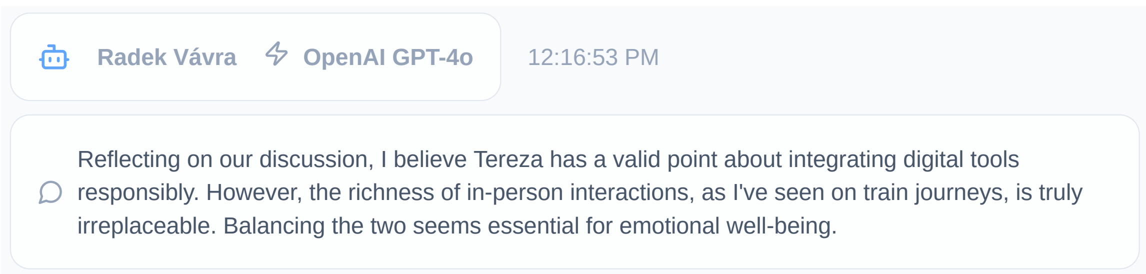

Figure 6.9: Situation during the conversation where the agent representing the persona opposes. In this message, the persona also mentions its previous experiences in the context of employment.

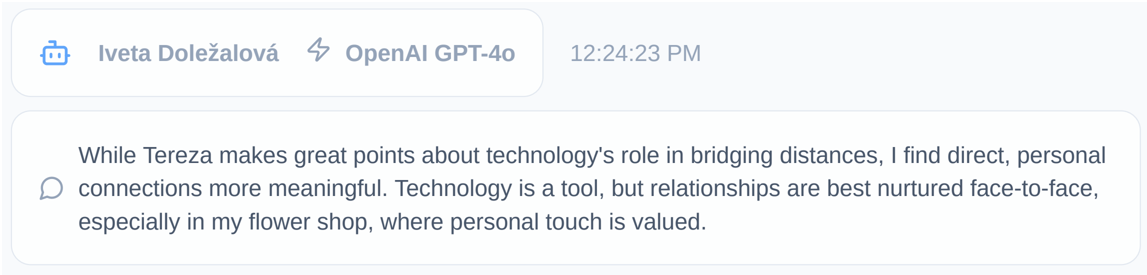

Figure 6.10: Simulation run with the scepticism in the persona's message. This text also includes employment and mentions previous personal experience.

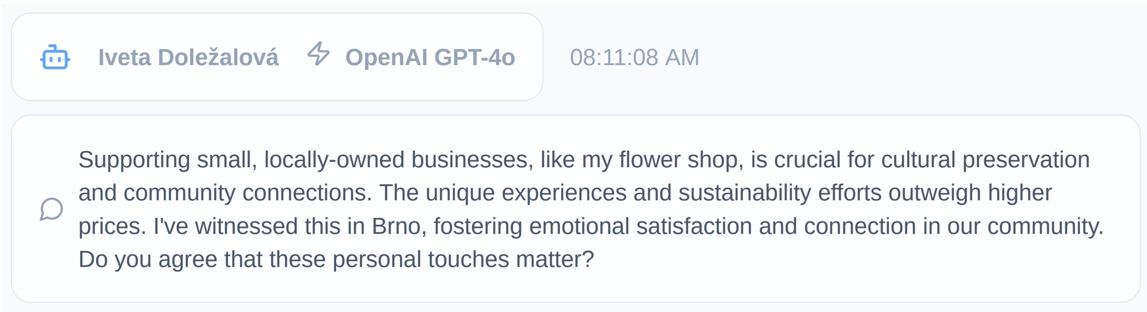

Figure 6.11: During the message generation, the agent also uses the persona's previous experiences as their place of living or employment.

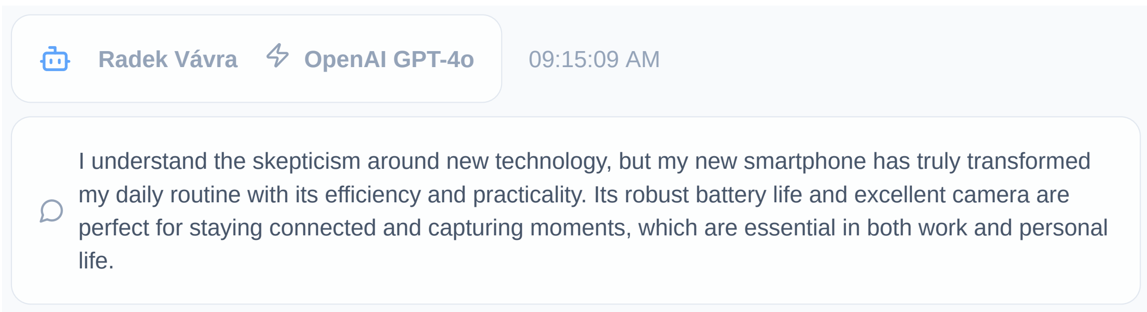

Figure 6.12: During the simulation, the agent is creative in how the message text is created and what information it incorporates.

6.6 Summary of Experimental Results

Based on the experiments with free discussion using evaluation dimensions, in summary, the PerSimChat framework outperforms in the following dimensions: *believability*, *credibility*, and *content depth and relevance*.

For the *turn-taking and flow* dimension, the best results are given by the single-agent zero-shot solution. Additionally, for this dimension, the PerSimChat framework beats the AutoGen baseline solution in all evaluation scenarios. These results show the naturalness of each system design. The pros of single-agent simulations are the connections between the messages because a single model can better hold the context of the conversation in a single output generation. This only confirms the results from [117]. With that, the PerSimChat framework performs better in turn-taking and flow than the AutoGen baseline because the system is designed to allow agents to speak in order based on their inner states and needs. With that, the AutoGen solution, while the external entity decides about the order of speech, could perform well when answering specific questions such as: „What do you think, John?“. However, this concept lacks the needs and consistency of a single agent behavior.

For both AutoGen and PerSimChat methods, these systems lack compared to a single model in terms of *conversation closure*. This is caused by the system design in that the agents are not informed about the message limit or when the conversation will end.

Incorporating the Lakmoos model for the message generation primarily, with other benefits, improves the framework's performance in the *social and emotional responsiveness*. Replacing the generated personas with those converted from Lakmoos data and replacing the model with the complete Lakmoos system for the message generation also improves the performance in the evaluation dimensions. The best results are especially in the dimensions where the framework performs best by itself, such as believability, credibility, and content depth and relevance. Those results are also shown in Figure 6.13.

The pairwise comparison between the systems even confirms the performance of the Lakmoos AI, s.r.o. proprietary solutions. The results using the Lakmoos model are shown in Figure 6.14 and incorporate the Lakmoos system in Figure 6.15.

Using open-source models, the Mistral Small outperforms the Mistral NeMo model. When comparing the PerSimChat framework with the baselines, it performs slightly better but is still comparable to the AutoGen solution. The AutoGen outperforms a single-agent system, while this system performs better than the proposed framework. Such inconsistency is primarily attributed to the fact that when the evaluator model sees a different pair of systems' solutions, it weights the evaluation metrics differently. So, when one metric is more important for one pair, this metric is not crucial to comparing the different pairs. Experiments in the broader subset of tasks should also be performed to clarify the results.

In the context of group debate, the proposed system performance is comparable to the other multi-agent debate (MAD) solutions, especially in the StrategyQA, GSM8k, and Aqua benchmarks. Even though the proposed system is not among the best-performing systems in those benchmarks, it beats the MAD with the Judge system and on two datasets — StrategyQA and GSM8k. I would describe these results as satisfactory, since the primary purpose and goal of the system is not to perform complex inferences, but to focus on reliable communication based on descriptions of people and their opinions. Also, the compared MAD systems are primarily tuned to the best performances on such datasets and reasoning tasks. With that, it is in the nature of human conversation that the group may not always get to the best solution in general. Additionally, these results show the reasoning capabilities of the whole PerSimChat framework.

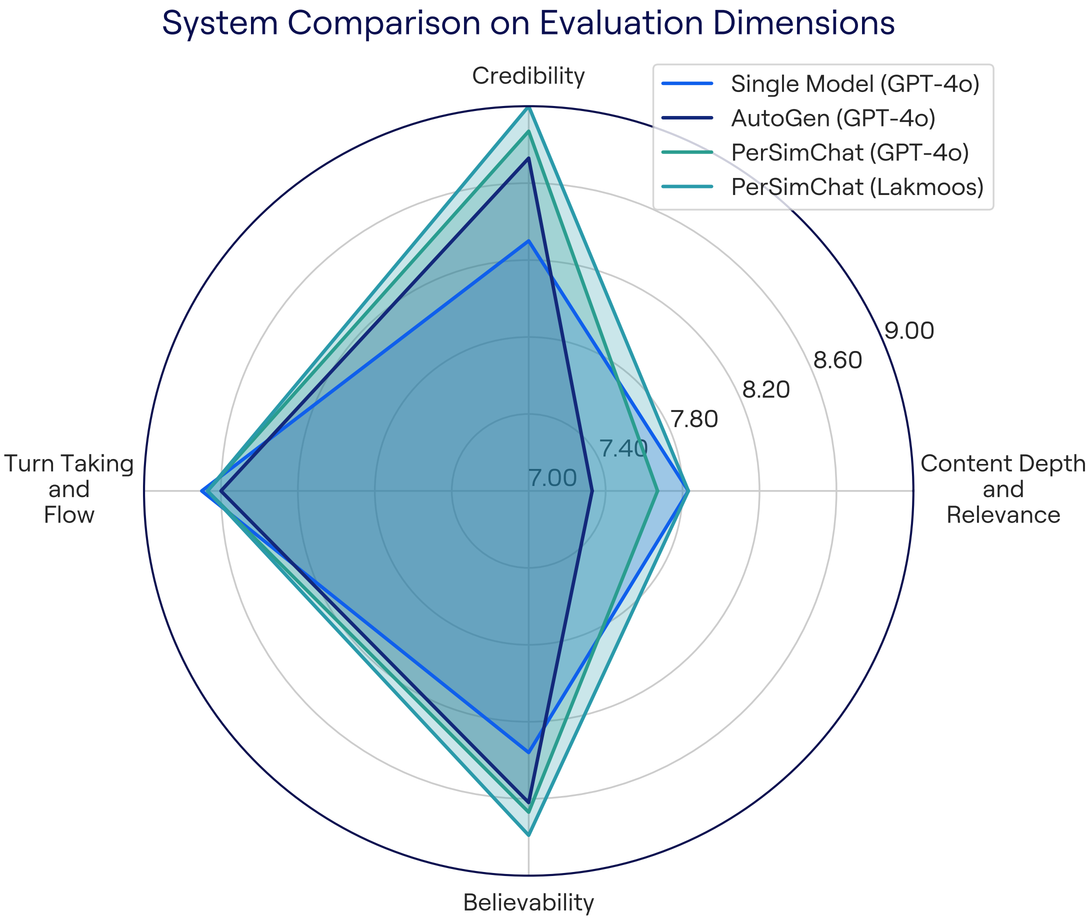

Figure 6.13: The evaluation of the PerSimChat framework, two versions (including GPT-4o or Lakmoos system for the message text generation, which stands for PerSimChat Base and PerSimChat LakSys) with baseline solutions — Single-Agent and AutoGen. It highlights the performance of the system on the evaluation dimensions, namely *believability*, *turn-taking and flow*, *credibility*, and *content depth and relevance*.

With the aim of gaining more knowledge about the properties of the designed system, the study, including human crowdworkers, was conducted in the form of a public survey. The results show that the participants prefer short messages with slang expressions, common words, and short words to those provided by a single LLM conversation generation. This also prevails over formal language used by a multi-agent system (AutoGen and PerSimChat) and primarily by the proposed solution. However, there is also a contradiction here with the cases where the tool will be used by an external partner. In addition, the expectations of the LLM evaluator and the human evaluators are also different.

Extending the measurements and evaluation of the PerSimChat framework with a large language model, benchmarks, or human crowdworkers, the subjective study of specific use cases brings further findings. Among the main advantages of the system design is the diversity of the agent's answers. With that, if relevant, the agents mention their previous experiences, such as a city of living, employment, or hobbies.

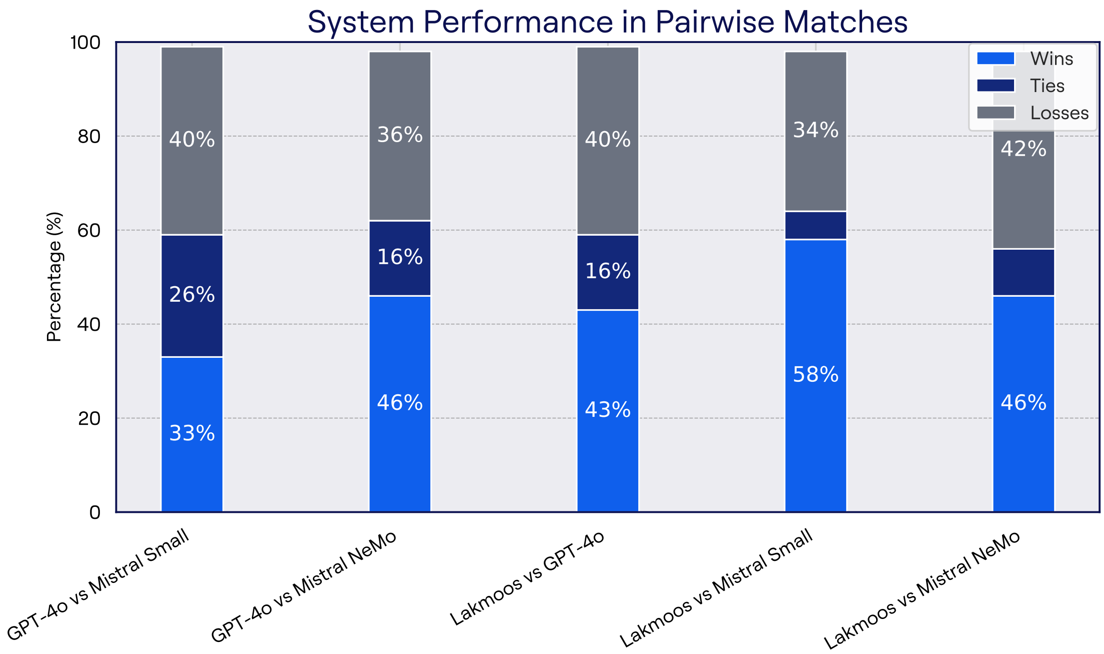

Figure 6.14: Visualization of the pairwise comparison of the multiple versions of the PerSimChat system. The data shown is for multiple models: GPT-4o, Mistral Small, Mistral NeMo, and Lakmoos model. These stand in order of PerSimChat Base, MiS, MiNM, and LakMod. The FairEval tool was used for the evaluation. For each pair, the percentages of *wins*, *ties*, and *losses* are provided.

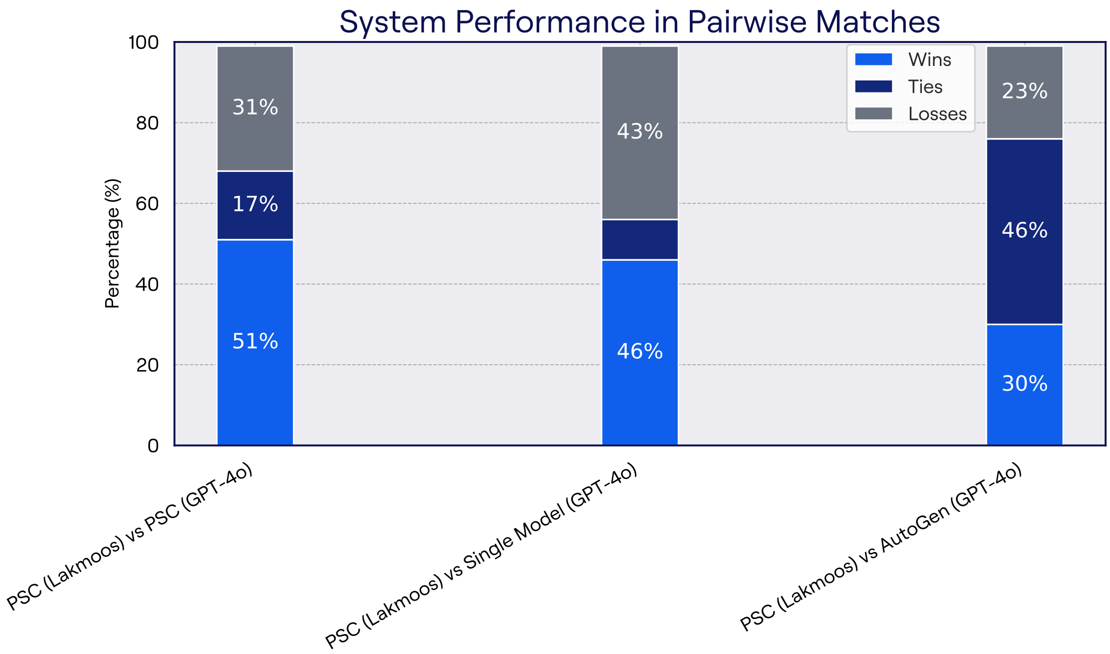

Figure 6.15: Pairwise comparison of PerSimChat framework (noted as PSC) as two versions, with GPT-4o model (PerSimChat Base) and Lakmoos system (PerSimChat LakSys). The two baselines are used in the comparison: Single-Agent (noted as Single Model) and the AutoGen solution. The FairEval tool was used for the evaluation. For each pair, the percentages of *wins*, *ties*, and *losses* are provided.

Chapter 7

Discussion and Future Directions

In this chapter, I discuss the overall summary of the system's design and the experimental results for the proposed framework evaluation cases. Simultaneously, the pros and cons of the system are highlighted with possible future directions and limitations. Additionally, I state the ethical considerations for this work and its implications.

7.1 Discussion

To the best of my knowledge, this work is the first to integrate and improve the dynamic retrieval system, similarly to [79], and study this concept embedded primarily in conversation generation. Unlike the authors of the generative agents, I incorporate this concept in the agent's behavior of message generation instead of performing actions in the game environment. Based on that, I discovered the benefits and limitations of this architecture. I consider the primary benefits to be offloading the language model and selecting only important memories for the agent that reflect the conversation context. However, the benefits are also the limitations based on how well this system selects the memories at a given moment.

Compared with the AutoGen library agent's speech order selection, I would rate the PerSimChat framework more similar to real-world conversation. Although the AutoGen solution is more comparable to a formal discussion with a human facilitator, my solution is more naturally based on unrestrained ordinary conversational communication. Compared with the SOTOPIA framework [118], the PerSimChat system offers richer context for the agents using the persona description structure, generated from scratch and those defined by incorporating the Lakmoos AI, s.r.o. proprietary data. This leads to many more possibilities from which the language model can take information and reflect the agent's personality. However, sometimes the language model is too focused on this information, repeating it multiple times, or it is too unnatural to base opinions on a person's place of residence or roots. Lastly, the memory filtering concept is as good as the model's performance in deciding about the relevance or irrelevance of memories.

One of the contributions of this work is the incorporation of the agent's knowledge levels in the message generation prompt. Based on that, the agent's response is reasonable to the persona's age, education, and social status. Although defining a knowledge level extends the length of the input text for the model, I observed a significant improvement in the plausibility of the text. The agent's messages became less machine-like and more like the everyday human dialect in formal scenarios.

Incorporating human ratings through a public survey conveys several important discoveries to us. Most people prefer short sentences using colloquial words. However, this result is different from what the „expert“ LLM evaluator prefers, and for what use cases the proposed system was designed for the external partner. Based on this, it depends on whether it is a real-life situation where people are facing each other and have to respond immediately, or whether it is, for example, an online discussion where a person has plenty of time to think up a response and arguments. In the second situation, more complex sentence structures are expected. This is also confirmed by some of the statements provided by the questionnaire participants (*„However, these conversations could be found anywhere. This does not say as much about the AI as it does about real people.“* and *„However, short response is not always the best response, mainly depending on the topic.“*).

The proposed PerSimChat framework can be modified in several ways to suit the preferences of human evaluators. First, the requirements for shorter and more informal sentences can be made through the message text generation prompt. In addition, I can force the model and the agent itself to stylize the text into a more colloquial speech, including slang expressions. Additionally, some examples of such language can be provided to the model because the single model generation possesses the informal speech characteristics for the complete conversation generation, but not for the question answering (like single message text creation) with chat completion. In general, we can offer two types of agents' behavior based on the use cases, or combine these two linguistic worlds (colloquial and formal) to achieve a consensual form of a text for all system users.

I also performed the human study with the 25 crowdworkers. Krippendorff's alpha, denoting the level of agreement among multiple evaluators, was calculated as $\alpha = 0.3674$. This denotes a lot of variability in the ratings. To increase the participants' agreement, the future study should decrease the level of subjectivity with a more precise description of the assessed dimension and a more thorough explanation of the evaluated task. Of course, the number of rated conversations and the number of participants included can also be increased. In addition, I also used the Wilcoxon signed-rank test, in which the hypothesis that the AutoGen solution performs better than PerSimChat was not proven (*p-value* $= 0.54$ with $\alpha = 0.05$). Based on the results of the human study, future improvements could focus on reducing the length of each message to half of its current length within the proposed framework. In addition, the formality of the text could be reduced by additional prompting requirements when the message is generated. This approach should primarily reduce the formality while also penalizing the system for using vulgar or socially inappropriate language.

From the other point of view, with a focus on use cases, the two main problems were shown. The first is that personas sometimes repeat the same thoughts during the conversation. This imperfection primarily stems from the situation that the current information was not provided during the dynamic memory retrieval process, or that the thought was too much generalized through the topic and keywords extraction. To solve these issues, multiple approaches can be used. One of the solutions can be for this extraction not to be so strict and provide a broader context of the conversation for the agent. With that, some possible improvements can be made to the retrieval function to highlight the information provided during the conversation. In addition, the penalization for repetition is possible.

The second problem is with the situation when agents sometimes do not maintain the context of the assignments. This also offers some possible fixes. For example, explicit tracking of the goal could be included, which means that in the agent's memory, the explicit goal that it should not forget can be added. Partially, the social goals offer these, but they have to be manually specified. With the concept of a judge, the external agent can monitor the direction to the goal, and alternatively manage individual agents. Lastly, in the agents' planning module, a mechanism that will continuously check whether progress is being made is realizable.

The cons of the framework's design are the conversation closure, and the primary limitation of the system is a fixed number of messages exchanged during the free discussion. To solve this complex problem, the agents can be better informed about the other person's needs to talk values. Alternatively, one of the agents can generate the closing message and end the conversation. Also, for the improvement of the performance, additional features such as dynamic memory retrieval, planning, and reflection increase the computational costs of the system and the time complexity. Optimization, such as using smaller and faster models for some processes, could reduce this issue.

In summary, the proposed PerSimChat framework performs best in the context of the naturality of the conversation based on the results of the believability, credibility, and content depth and relevance dimensions, using the language model evaluator. Replacing the GPT-4o model with the Lakmoos system for message generation even improves the system performance. Finally, the complete system proved its reasoning capacities during the group debate evaluation.

7.2 Future Directions and Plans

Possible improvements in the systems were discovered during the development and testing of the tool. Between these, the persona description can be extended with the other information to provide a richer context for the model, which can take the information and stylize the message according to the persona's naturalness. Also, concepts like dynamic memory retrieval and memory consolidation (memory filtering) have space for improvement. Furthermore, new findings in sociology and biology could extend the basic emotional state model or the memory model. The profile module could be extended with more advanced solutions to provide more natural persona behavior.

Among the main improvements, other concepts for free discussion generation, such as replacing the predefined number of messages and including the concept of conversation closure, would also be appropriate. Instead of using generated or data-like personas, it could be beneficial to incorporate concrete persona clones into the system. This could be done by providing the texts written by the persona to stylize the message output in this manner. Also, similar concepts like creating an exact living or non-living persona clone could be used, focusing on their style of writing. Including real-life situations can also increase the usability of the PerSimChat tool, like a simulation, when personas are under stress or the influence of narcotics or alcohol. Note that for now, this can be partially set by social goals. To make the simulation more complex and similar to real-world scenarios, concepts such as personas in a group, sometimes talking in pairs, and sometimes together, can be provided. Lastly, testing the system while each participating persona uses a different large language model would be appropriate. The PerSimChat framework design, of course, offers these options.

Using the results of the human feedback from the study conducted, the level of agreement, the amount of complicated word combinations (even in terms of pronunciation), and the formality of the text can be slightly decreased. Additionally, models representing the personas could be trained to provide shorter and polished messages (in terms of text flow), reducing the number of requirements in the message generation prompt.

I would rank this work among the first pioneers of simulating plausible human conversation in an otherwise field with enormous potential and space for improvement.

7.3 Limitations

The current state of the proposed system and its evaluation study contains certain limitations. Among the system's other benefits, such as adequate questioning and mentioning of the other conversation participants in the message, agents' opposing or skepticism, or creativity in the speech provided, the system also suffers from several shortcomings that are appropriate given the system's design. Similarly to the SOTOPIA framework [118], in the more extended conversation, the agents repeat the same thoughts. This can be caused by the fact that during the dynamic retrieval of the agent's memories, the previously said facts are not included in this limited buffer of memories. Also, with more complex tasks, rather than solving a problem, the agents start to converse about the problem topic. However, this can be fixed mainly by redefining the user prompt. Some of the other possible issues with the system are based on the nature of the behavior of language models. For example, even if the persona has already spoken, it starts the conversation as if it were at the beginning. Personas also repeat exact phrases or provide unexpected text outputs.

In terms of experiments, the evaluation was limited only to the concrete models investigated. With that, the focus was on concrete types of tasks and topics. Based on this fact, it is not clear how the agents will behave, for example, during crisis scenarios, negotiation, stress situations, etc. Despite an evaluation of the system by real humans, the number of participants was limited. With that, most of the participants had previous experience communicating with AI, and were made up of people who were clearly knowledgeable about the generative AI field. Additionally, pairwise comparison considers only comparing two of the three possible systems at once, which can cause the inconsistency of the weighting of these systems by the large language model evaluator. The evaluation is due to the costs associated with the chat completions, limited to only three personas, and the corresponding number of messages and rated conversations. With that, the system, in general, offers only a limited number of predefined personas, and the new ones have to be created by the generation or conversion process. Additionally, personas do not directly represent specific living or nonliving persons.

The nondeterminism of the text generation by the large language model causes the limitations of the reproducibility of the results. Despite the fact that the system is run with the same settings, parameters, and prompts, the responses of the agents differ. The evaluations by the language model can include the biases represented in its training data, and they automatically inherit the imperfections of the model design or the training data.

7.4 Ethical Considerations

The introduced PerSimChat framework for the simulation of human conversation also raises important ethical concerns that must be addressed.

First, based on language generation errors, the output text can contain harmful content with foul language expressions or abuse. In addition, processes of mutual influence and manipulation of agents may occur during communication. With that, the imperfections of the proposed system behavior do not have to reflect the real-world situations with humans. To address this issue, the systems and tools that will use the proposed solution or framework design should contain a disclaimer with clearly specified information that the user is watching digital entities, interacting with a simulation, and not real humans.

Second, the users of the framework may tend to form emotional relationships with the agents on many levels. At the same time, people may misinterpret it as communication between real humans. Similarly to the previous case, this problem can be partially solved by strongly informing the audience about the fact that this is a simulation and that all the people and information used are completely imaginary.

Third, with the expansion of the possibility of simulating a conversation between multiple persons, the use of this tool also increases the risk of deceiving people and the rise of deepfakes. Potential misuse includes generating persuasive but fake dialogue transcripts, which may resemble real individuals and influence public opinion – a typical *textual deepfake* threat. To detect these issues, on the server side, for public deployment at the external partner, the logs are saved, and the inappropriate communication can eventually be detected and closed.

Fourth, this simulation should not replace real humans, and its main purpose is to serve as a prototyping tool during the design phase of experiments. The restriction on using the predefined personas in the publicly available version prevents replacing personas in the conversation with data about real people. At the same time, by limiting the publication of the tool's implementation, I prevent its unethical and socially inappropriate use.

Lastly, the proposed approach for generating the persona descriptions from scratch and converting from the Lakmoos AI data provided does not use information about real persons, whose identities could be misused by the simulation. The proprietary data are created from the real statistics and surveys, but the entities are virtual.

Chapter 8

Conclusion

This work presented an experimental environment for simulating believable human multi-persona communication. The primary motivation for such a system is to design an experimental tool with a real-life conversation replication. The first purpose is to simulate possible scenarios of free discussion. At the same time, the second purpose of the system is information retrieval with the simultaneous consensus of multiple autonomous agents.

This thesis analyzed the related literature that has been published so far. First, it explained essential concepts in the field of artificial intelligence, focusing on neural networks and their use for natural language generation using large language models. It also described the latest approaches and concepts for using these models and maximizing their potential. I then discussed the human brain functions related to this topic and focused on the influential factors in real-persona conversations. With this knowledge, I summarized the most important concepts of autonomous systems, human-AI interaction, and approaches to the different parts of the system's design, with related technical literature and publications.

At the same time, this work defined itself in relation to already established systems, uses the latest introduced concepts, and proposes extensions and modifications to achieve more believable human interaction in the proposed *PerSimChat* framework. I described the architecture of the simulation framework, starting with the single agent design using cognitive modules, dynamic memory retrieval, persona description generation, incorporating the Lakmoos AI proprietary data, and knowledge levels. Subsequently, I explained the simulation algorithm for free discussion and group debate. This includes the *need to talk* concept and consensus achievement with openness to consensus scores. Lastly, the simulation scenarios considered were enumerated.

Furthermore, I describe the core of the implementation of the proposed simulation environment with the extension of the framework with a user interface. I have increased the tool's usability via the web application, which is publicly available and deployed by an external partner. In connection with the Lakmoos AI, s.r.o. company, the designed system is currently integrated into their code base. The proposed framework is evaluated in multiple experiments using multiple large language models and compared with the baseline solutions, achieving the best scores in the naturalness evaluation dimensions. Following the evaluation conducted by the human volunteers, the Wilcoxon signed-rank test was performed to assess the statistical significance of the observed differences across systems. In addition, the concrete use cases highlight the benefits and limitations of the system. The results are discussed, and possible improvements and future work have been defined.

Bibliography

[1] Abbasiantaeb, Z.; Yuan, Y.; Kanoulas, E. and Aliannejadi, M. *Let the LLMs Talk: Simulating Human-to-Human Conversational QA via Zero-Shot LLM-to-LLM Interactions*. 2023. Available at: https://arxiv.org/abs/2312.02913.

[2] Aggarwal, S.; Mandowara, D.; Agrawal, V.; Khandelwal, D.; Singla, P. et al. Explanations for CommonsenseQA: New Dataset and Models. In: Zong, C.; Xia, F.; Li, W. and Navigli, R., ed. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, August 2021, p. 3050–3065. Available at: https://aclanthology.org/2021.acl-long.238.

[3] Aher, G.; Arriaga, R. I. and Kalai, A. T. *Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies*. 2023. Available at: https://arxiv.org/abs/2208.10264.

[4] Averill, J. R. The Social Construction of Emotion. In: Harré, R., ed. *The Social Construction of Reality*. Academic Press, 1980.

[5] Bahdanau, D.; Cho, K. and Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In: *Proceedings of the 2014 International Conference on Learning Representations (ICLR)*. 2014. Available at: https://arxiv.org/abs/1409.0473.

[6] Bail, C. Can Generative AI improve social science? *Proceedings of the National Academy of Sciences*, may 2024, vol. 121.

[7] Bandhu, D.; Mohan, M. M.; Nittala, N. A. P.; Jadhav, P.; Bhadauria, A. et al. Theories of motivation: A comprehensive analysis of human behavior drivers. *Acta Psychologica*, 2024, vol. 244, p. 104177. ISSN 0001-6918. Available at: https://www.sciencedirect.com/science/article/pii/S0001691824000544.

[8] Bandura, A. *Social Learning Theory*. Prentice-Hall, 1977.

[9] Barsade, S. G. and Gibson, D. E. Group Emotion: A View From Top and Bottom. *Research in Organizational Behavior*, 1998, vol. 20, p. 81–102.

[10] Beck, A. T. *Cognitive Therapy and the Emotional Disorders*. International Universities Press, 1967.

[11] Bowlby, J. *Attachment and Loss: Volume 1. Attachment*. Basic Books, 1969.

[12] Campos Moinier, K.; Murday, V. and Brunel, L. Individual differences in social interaction contexts: Examining the role of personality traits in the degree of self-other integration. *Personality and Individual Differences*, 2023, vol. 203, p. 112002. ISSN 0191-8869. Available at: https://www.sciencedirect.com/science/article/pii/S0191886922005074.

[13] Chan, C.-M.; Chen, W.; Su, Y.; Yu, J.; Xue, W. et al. *ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate*. 2023. Available at: https://arxiv.org/abs/2308.07201.

[14] Chen, C.; Yao, B.; Ye, Y.; Wang, D. and Li, T. J.-J. Evaluating the LLM Agents for Simulating Humanoid Behavior. In: HEAL Workshop. *Proceedings of the HEAL Workshop 2024*. 2024. Available at: https://heal-workshop.github.io/papers/35_evaluating_the_llm_agents_for_.pdf.

[15] Chen, G.; Dong, S.; Shu, Y.; Zhang, G.; Sesay, J. et al. AutoAgents: A Framework for Automatic Agent Generation. In: *International Joint Conference on Artificial Intelligence*. 2023. Available at: https://api.semanticscholar.org/CorpusID:263310605.

[16] Chen, J. C.-Y.; Saha, S. and Bansal, M. ReConcile: Round-Table Conference Improves Reasoning via Consensus Among Diverse LLMs. *ArXiv preprint arXiv:2309.13007*, 2023.

[17] Chen, J. C.-Y.; Saha, S. and Bansal, M. *ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs*. 2024. Available at: https://arxiv.org/abs/2309.13007.

[18] Cho, K.; Merriënboer, B. van; Bahdanau, D.; Bengio, Y. and Schwenk, H. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, p. 1724–1734. Available at: https://arxiv.org/abs/1406.1078.

[19] Cipi, E. and Cico, B. Simulation of an Agent Based System Behavior in a Dynamic and Unpredicted Environment. *World of Computer Science and Information Technology Journal (WCSIT)*, january 2011, vol. 1, p. 2221–741.

[20] Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H. et al. Training Verifiers to Solve Math Word Problems. *CoRR*, 2021, abs/2110.14168. Available at: https://arxiv.org/abs/2110.14168.

[21] Costa, P. T. and McCrae, R. R. *Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) Professional Manual*. Odessa, FL: Psychological Assessment Resources, 1992.

[22] Damasio, A. *Descartes' Error: Emotion, Reason, and the Human Brain*. G.P. Putnam's Sons, 1994. Explores the role of emotions in decision-making and how effective decision-making requires integration of emotional responses, memory, and reasoning.

[23] Del Giudice, M.; Gangestad, S. W. and Kaplan, H. S. Life history theory and human behavior: Testing predictions from evolutionary ecology. *Annual Review of Psychology*. Annual Reviews, 2015, vol. 66, p. 405–428.

[24] Dibia, V.; Chen, J.; Bansal, G.; Syed, S.; Fourney, A. et al. *AutoGen Studio: A No-Code Developer Tool for Building and Debugging Multi-Agent Systems*. August 2024. Available at: https://www.microsoft.com/en-us/research/publication/autogen-studio-a-no-code-developer-tool-for-building-and-debugging-multi-agent-systems/. Preprint.

[25] Douglas, M. *The Disciplined Trader: Developing Winning Attitudes*. Paramus, NJ: New York Institute of Finance, 1990. ISBN 9780132157575. Available at: https://www.amazon.com/Disciplined-Trader-Developing-Winning-Attitudes/dp/0132157578.

[26] Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B. and Mordatch, I. *Improving Factuality and Reasoning in Language Models through Multiagent Debate*. 2023. Available at: https://arxiv.org/abs/2305.14325.

[27] Ekman, P. An argument for basic emotions. *Cognition and Emotion*. Taylor & Francis, 1992, vol. 6, 3-4, p. 169–200.

[28] Elliott, E. W. and Kiel, L. D. Exploring cooperation and competition using agent-based modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 2002, vol. 99, p. 7193 – 7194. Available at: https://api.semanticscholar.org/CorpusID:9380125.

[29] Engineering, C. U. S. of and Science, A. *Why do we remember emotional events better?* 2023. Available at: https://www.sciencedaily.com/releases/2023/01/230118195926.htm. Accessed: November 22, 2024.

[30] (FAIR)†, M. F. A. R. D. T.; Bakhtin, A.; Brown, N.; Dinan, E.; Farina, G. et al. Human-level play in the game of <i>Diplomacy</i> by combining language models with strategic reasoning. *Science*, 2022, vol. 378, no. 6624, p. 1067–1074. Available at: https://www.science.org/doi/abs/10.1126/science.ade9097.

[31] Faul, L. and LaBar, K. S. Emotional Memory in the Human Brain. In: *The Oxford Handbook of the Neurobiology of Learning and Memory*. Oxford University Press, 2020. ISBN 9780190069162. Available at: https://doi.org/10.1093/oxfordhb/9780190069162.013.2.

[32] Fisher, R. and Ury, W. *Getting to Yes: Negotiating Agreement Without Giving In*. 3rdth ed. New York: Penguin Books, 2011. ISBN 0143118757.

[33] Fourney, A.; Bansal, G.; Mozannar, H.; Tan, C.; Salinas, E. et al. *Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks*. 2024. Available at: https://arxiv.org/abs/2411.04468.

[34] Gao, C.; Lan, X.; Li, N.; Yuan, Y.; Ding, J. et al. *Large Language Models Empowered Agent-based Modeling and Simulation: A Survey and Perspectives*. 2023. Available at: https://arxiv.org/abs/2312.11970.

[35] Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J. et al. *S3: Social-network Simulation System with Large Language Model-Empowered Agents*. 2023. Available at: https://arxiv.org/abs/2307.14984.

[36] Gao, J.; Gebreegziabher, S. A.; Choo, K. T. W.; Li, T. J.-J.; Perrault, S. T. et al. A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM, May 2024, p. 1–11. CHI '24. Available at: http://dx.doi.org/10.1145/3613905.3650786.

[37] Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D. et al. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*. Cambridge, MA: MIT Press, 2021, vol. 9, p. 346–361. Available at: https://aclanthology.org/2021.tacl-1.21.

[38] Goldstein, E. B. *Cognitive Psychology: Connecting Mind, Research and Everyday Experience*. 4thth ed. Cengage Learning, 2014. Describes the role of selective attention in perception and learning.

[39] Goleman, D. *Emotional Intelligence: Why It Can Matter More Than IQ*. Bantam Books, 1995.

[40] Goodfellow, I.; Bengio, Y. and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[41] Grossmann, I.; Feinberg, M.; Parker, D.; Christakis, N.; Tetlock, P. et al. AI and the transformation of social science research. *Science (New York, N.Y.)*, june 2023, vol. 380, p. 1108–1109.

[42] Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S. et al. *Large Language Model based Multi-Agents: A Survey of Progress and Challenges*. 2024. Available at: https://arxiv.org/abs/2402.01680.

[43] Gürcan Önder. *LLM-Augmented Agent-Based Modelling for Social Simulations: Challenges and Opportunities*. 2024. Available at: https://arxiv.org/abs/2405.06700.

[44] Habermas, J. *The Theory of Communicative Action: Volume 1: Reason and the Rationalization of Society*. Boston: Beacon Press, 1984. ISBN 0807011546.

[45] Hagos, D. H.; Battle, R. and Rawat, D. B. *Recent Advances in Generative AI and Large Language Models: Current Status, Challenges, and Perspectives*. 2024. Available at: https://arxiv.org/abs/2407.14962.

[46] Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M. et al. *FOLIO: Natural Language Reasoning with First-Order Logic*. 2024. Available at: https://arxiv.org/abs/2209.00840.

[47] Hana, J. *Introduction to Linguistics*. 2020. Available at: https://ufal.mff.cuni.cz/~hana/teaching/ling1/ling1.pdf. Accessed: 2024-12-06.

[48] Harvey, J. B. The Abilene Paradox: The Management of Agreement. *Organizational Dynamics*, 1974, vol. 3, no. 1, p. 63–80.

[49] Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S. et al. Measuring Mathematical Problem Solving With the MATH Dataset. *CoRR*, 2021, abs/2103.03874. Available at: https://arxiv.org/abs/2103.03874.

[50] Hochreiter, S. and Schmidhuber, J. Long Short-term Memory. *Neural computation*, december 1997, vol. 9, p. 1735–80.

[51] Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y. et al. *MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework*. 2024. Available at: https://arxiv.org/abs/2308.00352.

[52] Janis, I. L. *Groupthink: Psychological Studies of Policy Decisions and Fiascoes*. Boston: Houghton Mifflin, 1972. ISBN 0395177343.

[53] Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D. et al. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* New York, NY, USA: Association for Computing Machinery, march 2023, vol. 55, no. 12. ISSN 0360-0300. Available at: https://doi.org/10.1145/3571730.

[54] Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B. et al. *Scaling Laws for Neural Language Models*. 2020. Available at: https://arxiv.org/abs/2001.08361.

[55] Khan, A.; Hughes, J.; Valentine, D.; Ruis, L.; Sachan, K. et al. *Debating with More Persuasive LLMs Leads to More Truthful Answers*. 2024. Available at: https://arxiv.org/abs/2402.06782.

[56] Kim, C. Y.; Lee, C. P. and Mutlu, B. Understanding Large-Language Model (LLM)-powered Human-Robot Interaction. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, March 2024, p. 371–380. HRI '24. Available at: http://dx.doi.org/10.1145/3610977.3634966.

[57] Lazarus, R. S. *Emotion and Adaptation*. Oxford University Press, 1991.

[58] LeCun, Y.; Bottou, L.; Bengio, Y. and Haffner, P. Backpropagation Applied to Handwritten Zip Code Recognition. In: *Neural Information Processing Systems*. 1989, p. 396–404.

[59] Lewis, P. S. H.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *CoRR*, 2020, abs/2005.11401. Available at: https://arxiv.org/abs/2005.11401.

[60] Li, G.; Hammoud, H. A. A. K.; Itani, H.; Khizbullin, D. and Ghanem, B. *CAMEL: Communicative Agents for „Mind“ Exploration of Large Language Model Society*. 2023. Available at: https://arxiv.org/abs/2303.17760.

[61] Li, Y.; Zhang, Y. and Sun, L. *MetaAgents: Simulating Interactions of Human Behaviors for LLM-based Task-oriented Coordination via Collaborative Generative Agents*. 2023. Available at: https://arxiv.org/abs/2310.06500.

[62] Li, Y.; Du, Y.; Zhang, J.; Hou, L.; Grabowski, P. et al. *Improving Multi-Agent Debate with Sparse Communication Topology*. 2024. Available at: https://arxiv.org/abs/2406.11776.

[63] Liang, T.; He, Z.; Jiao, W.; Wang, X.; Wang, Y. et al. *Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate*. 2024. Available at: https://arxiv.org/abs/2305.19118.

[64] Ling, W.; Yogatama, D.; Dyer, C. and Blunsom, P. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In: Barzilay, R. and Kan, M.-Y., ed. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, p. 158–167. Available at: https://aclanthology.org/P17-1015.

[65] Liu, R.; Yang, R.; Jia, C.; Zhang, G.; Zhou, D. et al. *Training Socially Aligned Language Models on Simulated Social Interactions*. 2023. Available at: https://arxiv.org/abs/2305.16960.

[66] Long, J. *Large Language Model Guided Tree-of-Thought*. 2023. Available at: https://arxiv.org/abs/2305.08291.

[67] Macal, C. and North, M. Tutorial on Agent-based Modeling and Simulation. In:. January 2005, vol. 2005, p. 2–15.

[68] Mccabe, D. P.; Roediger, H. L.; McDaniel, M. A.; Balota, D. A. and Hambrick, D. Z. The relationship between working memory capacity and executive functioning: evidence for a common executive attention construct. *Neuropsychology*, 2010, 24 2, p. 222–243. Available at: https://api.semanticscholar.org/CorpusID:7487230.

[69] McCulloch, W. S. and Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, 1943, vol. 5, no. 4, p. 115–133. Available at: https://www.cs.cmu.edu/~./epxing/Class/10715/reading/McCulloch.and.Pitts.pdf.

[70] Meek, C.; Chickering, D. M. and Heckerman, D. Autoregressive Tree Models for Time-Series Analysis. In: *Proceedings of the 2002 SIAM International Conference on Data Mining*. SIAM, 2002, p. 229–244.

[71] Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1956, vol. 63, no. 2, p. 81–97. Available at: https://psycnet.apa.org/record/1959-09865-001.

[72] Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R. et al. *Large Language Models: A Survey*. 2024. Available at: https://arxiv.org/abs/2402.06196.

[73] Minsky, M. *Society of Mind*. New York: Simon and Schuster, 1988. ISBN 978-0671657130.

[74] Mozannar, H.; Chen, V.; Wei, D.; Sattigeri, P.; Nagireddy, M. et al. Simulating Iterative Human-AI Interaction in Programming with LLMs.

In: *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*. December 2023. Available at https://openreview.net/pdf?id=0nRcZeeE5f.

[75] Nickerson, R. S. *Making Decisions*. MIT Press, 2001. Discusses how decision-making involves evaluating information based on preferences, beliefs, and emotions. Also covers the influence of cognitive biases and social contexts.

[76] Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J. et al. Adversarial NLI: A New Benchmark for Natural Language Understanding. In: Jurafsky, D.; Chai, J.; Schluter, N. and Tetreault, J., ed. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, p. 4885–4901. Available at: https://aclanthology.org/2020.acl-main.441.

[77] Northouse, P. G. *Leadership: Theory and Practice*. 8thth ed. Thousand Oaks, CA: SAGE Publications, 2018. ISBN 978-1544324261.

[78] Okada, A. et al., ed. *Behaviormetrics: Quantitative Approaches to Human Behavior*. Springer Nature Switzerland AG, 2024. Behaviormetrics. Available at: https://www.springer.com/series/16001.

[79] Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P. et al. *Generative Agents: Interactive Simulacra of Human Behavior*. 2023. Available at: https://arxiv.org/abs/2304.03442.

[80] Power, R. A. and Pluess, M. Heritability estimates of the Big Five personality traits based on common genetic variants. *Translational Psychiatry*, 2015, vol. 5, no. 7, p. e604–e604. ISSN 2158-3188. Available at: https://doi.org/10.1038/tp.2015.96.

[81] Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y. et al. *ChatDev: Communicative Agents for Software Development*. 2024. Available at: https://arxiv.org/abs/2307.07924.

[82] Radford, A. and Narasimhan, K. Improving Language Understanding by Generative Pre-Training. In:. 2018. Available at: https://api.semanticscholar.org/CorpusID:49313245.

[83] Rumelhart, D. E.; Hinton, G. E. and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 1986, vol. 323, no. 6088, p. 533–536. ISSN 1476-4687. Available at: https://doi.org/10.1038/323533a0.

[84] Ryan, R. M. and Deci, E. L. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 2000, vol. 55, no. 1, p. 68–78. Discusses concepts related to self-evolution and personal growth through learning, experience, and reflection.

[85] Rye, C. Memory Consolidation and Reconsolidation: Neural Mechanisms and Clinical Relevance. *Cambridge Journal of Human Behaviour*, october 2023, vol. 2, p. 24–30.

[86] Sarker, I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2021, vol. 2, no. 3, p. 160. ISSN 2661-8907. Available at: https://doi.org/10.1007/s42979-021-00592-x.

[87] Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K. et al. *Reflexion: Language Agents with Verbal Reinforcement Learning*. 2023. Available at: https://arxiv.org/abs/2303.11366.

[88] Spencer, J. and Helgesen, S. *The New Economics of Leadership: Distributed Leadership in Teams*. Boston, MA: Harvard Business Press, 2015. ISBN 978-1422197502.

[89] Sreedhar, K. and Chilton, L. *Simulating Human Strategic Behavior: Comparing Single and Multi-agent LLMs*. 2024. Available at: https://arxiv.org/abs/2402.08189.

[90] Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A. A. M.; Abid, A. et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Available at: https://openreview.net/forum?id=uyTL5Bvosj.

[91] Sternberg, R. J. and Sternberg, K. *Cognitive Psychology*. 7thth ed. Cengage Learning, 2016. Covers reasoning as a critical process for problem-solving and decision-making, emphasizing its role in analyzing situations and formulating solutions.

[92] Surowiecki, J. *The Wisdom of Crowds*. New York: Doubleday, 2004. ISBN 0385503865.

[93] Sutskever, I.; Vinyals, O. and Le, Q. V. Sequence to Sequence Learning with Neural Networks. In: Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. and Weinberger, K., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27. Available at: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[94] Tang, J.; Gao, H.; Pan, X.; Wang, L.; Tan, H. et al. *GenSim: A General Social Simulation Platform with Large Language Model based Agents*. 2024. Available at: https://arxiv.org/abs/2410.04360.

[95] Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A. et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. Available at: https://arxiv.org/abs/2302.13971.

[96] TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, october 1950, LIX, no. 236, p. 433–460. ISSN 0026-4423. Available at: https://doi.org/10.1093/mind/LIX.236.433.

[97] Tyng, C. M.; Amin, H. U.; Saad, M. N. M. and Malik, A. S. The Influences of Emotion on Learning and Memory. *Frontiers in Psychology*, 2017, vol. 8. ISSN 1664-1078. Available at: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2017.01454.

[98] VAN KLEEF, G. A. How Emotions Shape Social Life: The EASI Model. *Current Directions in Psychological Science*, 2009, vol. 18, no. 3, p. 184–188. Available at: https://doi.org/10.1111/j.1467-8721.2009.01633.x.

[99] VASWANI, A.; SHANKAR, N.; PARMAR, N.; USZKOREIT, J.; JONES, L. et al. Attention Is All You Need. In: *Advances in Neural Information Processing Systems*. 2017, p. 5998–6008. Available at: https://arxiv.org/abs/1706.03762.

[100] WANG, G.; XIE, Y.; JIANG, Y.; MANDLEKAR, A.; XIAO, C. et al. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. 2023. Available at: https://arxiv.org/abs/2305.16291.

[101] WANG, L.; MA, C.; FENG, X.; ZHANG, Z.; YANG, H. et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*. Springer Science and Business Media LLC, march 2024, vol. 18, no. 6. ISSN 2095-2236. Available at: http://dx.doi.org/10.1007/s11704-024-40231-1.

[102] WANG, P.; LI, L.; CHEN, L.; ZHU, D.; LIN, B. et al. Large Language Models are not Fair Evaluators. *ArXiv*, 2023, abs/2305.17926.

[103] WANG, P.; LI, L.; CHEN, L.; ZHU, D.; LIN, B. et al. Large Language Models are not Fair Evaluators. *ArXiv*, 2023, abs/2305.17926.

[104] WANG, Q.; WANG, Z.; SU, Y.; TONG, H. and SONG, Y. *Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key?* 2024. Available at: https://arxiv.org/abs/2402.18272.

[105] WANG, X.; WEI, J.; SCHUURMANS, D.; LE, Q.; CHI, E. et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2023. Available at: https://arxiv.org/abs/2203.11171.

[106] WANG, Z.; MAO, S.; WU, W.; GE, T.; WEI, F. et al. Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. In: DUH, K.; GOMEZ, H. and BETHARD, S., ed. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Mexico City, Mexico: Association for Computational Linguistics, June 2024, p. 257–279. Available at: https://aclanthology.org/2024.naacl-long.15.

[107] WANG, Z.; CHIU, Y. Y. and CHIU, Y. C. *Humanoid Agents: Platform for Simulating Human-like Generative Agents*. 2023. Available at: https://arxiv.org/abs/2310.05418.

[108] WEI, J.; WANG, X.; SCHUURMANS, D.; BOSMA, M.; CHI, E. H. et al. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *CoRR*, 2022, abs/2201.11903. Available at: https://arxiv.org/abs/2201.11903.

[109] WU, Q.; BANSAL, G.; ZHANG, J.; WU, Y.; LI, B. et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In: *COLM 2024*. August 2024. Available at: https://www.microsoft.com/en-us/research/publication/autogen-enabling-next-gen-llm-applications-via-multi-agent-conversation-framework/.

[110] Xiao, Y.; Cheng, Y.; Fu, J.; Wang, J.; Li, W. et al. *How Far Are LLMs from Believable AI? A Benchmark for Evaluating the Believability of Human Behavior Simulation*. 2024. Available at: https://arxiv.org/abs/2312.17115.

[111] Xiong, K.; Ding, X.; Cao, Y.; Liu, T. and Qin, B. *Examining Inter-Consistency of Large Language Models Collaboration: An In-depth Analysis via Debate*. 2023. Available at: https://arxiv.org/abs/2305.11595.

[112] Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L. et al. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 2023. Available at: https://arxiv.org/abs/2305.10601.

[113] Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I. et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. 2023. Available at: https://arxiv.org/abs/2210.03629.

[114] Zhang, J. *Cognitive Functions of the Brain: Perception, Attention and Memory*. 2019. Available at: https://arxiv.org/abs/1907.02863.

[115] Zhang, J. *Secrets of the Brain: An Introduction to the Brain Anatomical Structure and Biological Function*. 2019. Available at: https://arxiv.org/abs/1906.03314.

[116] Zhang, Y.; Yang, J.; Yuan, Y. and Yao, A. C.-C. *Cumulative Reasoning with Large Language Models*. 2024. Available at: https://arxiv.org/abs/2308.04371.

[117] Zhou, X.; Su, Z.; Eisape, T.; Kim, H. and Sap, M. *Is this the real life? Is this just fantasy? The Misleading Success of Simulating Social Interactions With LLMs*. 2024. Available at: https://arxiv.org/abs/2403.05020.

[118] Zhou, X.; Zhu, H.; Mathur, L.; Zhang, R.; Yu, H. et al. *SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents*. 2024. Available at: https://arxiv.org/abs/2310.11667.

[119] Ziems, C.; Held, W.; Shaikh, O.; Chen, J.; Zhang, Z. et al. *Can Large Language Models Transform Computational Social Science?* 2024. Available at: https://arxiv.org/abs/2305.03514.

Appendix A

User Interface for the PerSimChat Application

The user interface of the PerSimChat web application provided for the eponymous framework is shown in Figure A.1. The main emphasis was placed on user-friendliness with a representative user interface. The web application allows for a simple understanding of how messages are exchanged between personas. In addition, it provides many settings, while the application offers wide possibilities for experimenting with the system architecture for the users themselves. At the same time, basic information about conversations and optionally a comprehensive analysis of individual messages is provided.

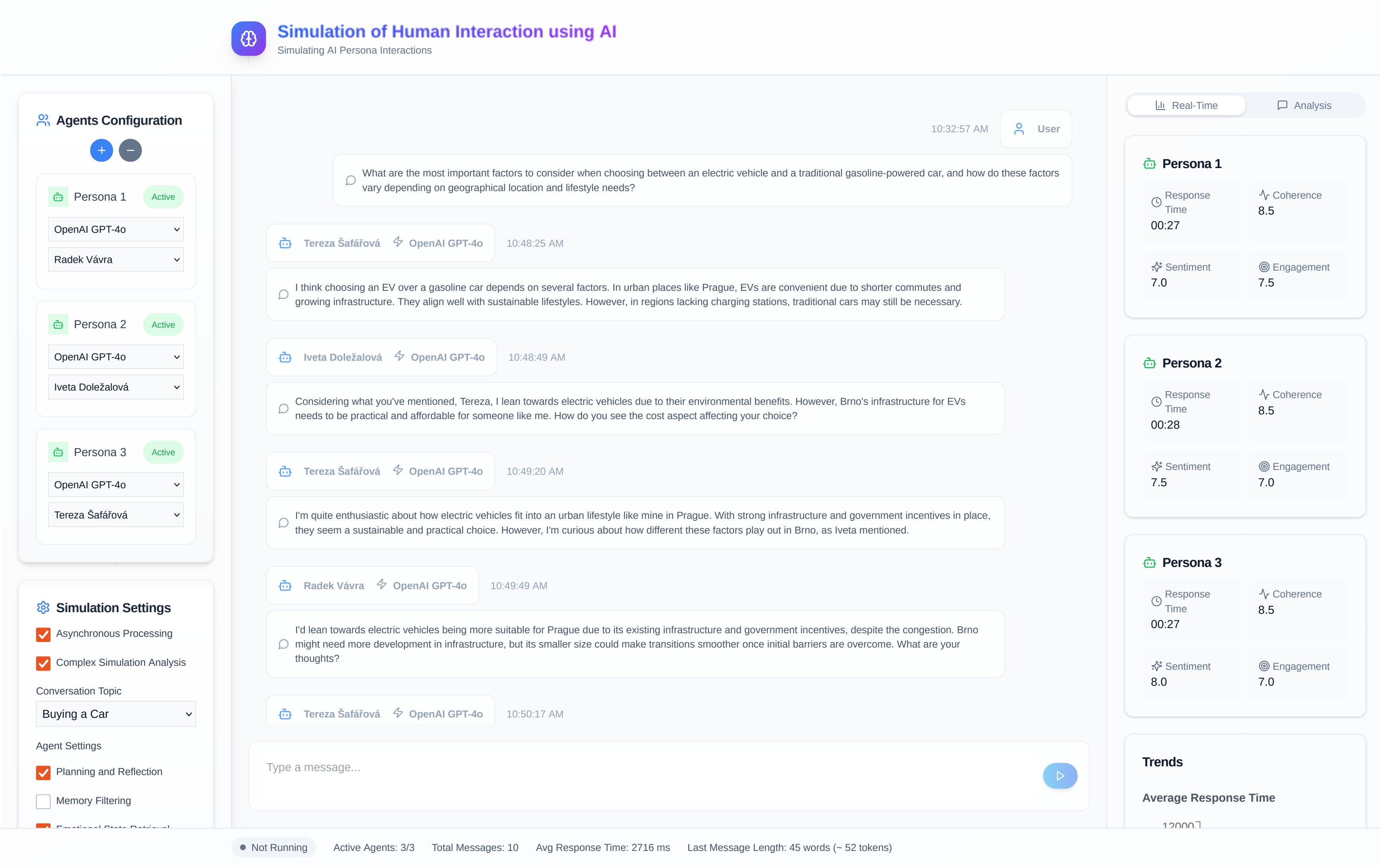

Figure A.1: User interface of the PerSimChat application.

Appendix B

Full Results of the Free Discussion Evaluation

The full results for the dimensions experimental evaluation using the *social goals* and predefined *relationships* are provided in Table B.1. The full dimensions results of the PerSimChat framework with two baseline solutions while using the Lakmoos AI converted persona data are shown in Table B.2. Lastly, the Table B.3 provides the results for the final dimensions evaluation of the compared systems for running for three rounds. The notation and system design are the same as in Section 6.1.

Method	GOAL	BEL	KNO	REL	CRE	TTF	CDR	SER	GPTS	CLO
Single-Agent Zero-Shot	8.9 ± 0.8	8.9 ± 0.66	5.4 ± 1.3	3.7 ± 0.84	8.27 ± 0.69	8.57 ± 0.5	7.93 ± 0.74	$\mathbf{7.7 \pm 1.02}$	8.97 ± 0.67	$\mathbf{8.77 \pm 0.77}$
AutoGen	$\mathbf{9.27 \pm 0.52}$	8.86 ± 0.74	$\mathbf{5.87 \pm 1.22}$	$\mathbf{4.2 \pm 0.85}$	$\mathbf{8.8 \pm 1.73}$	$\mathbf{8.97 \pm 0.67}$	7.63 ± 1.97	5.7 ± 1.44	$\mathbf{9.27 \pm 0.45}$	6.93 ± 1.74
PerSimChat SGR	8.97 ± 0.96	$\mathbf{8.93 \pm 0.8}$	5.47 ± 1.22	3.83 ± 0.79	8.5 ± 1.68	8.6 ± 0.56	$\mathbf{8.07 \pm 0.94}$	5.93 ± 1.39	9.03 ± 0.56	7.77 ± 1.22

Table B.1: Full evaluation results of the PerSimChat framework with two baseline solutions — Single-Agent and AutoGen when including the social goals and predefined relationships (and therefore PerSimChat SGR design version). *GOAL* dimension stands for goal completion, *BEL* for believability, *KNO* for knowledge, *REL* for relationship, *CRE* for credibility, *TTF* for turn-taking and flow, *CDR* for content depth and relevance, *SER* for social and emotional responsiveness, *GPTS* for goal progression or task solving, and *CLO* for conversation closure.

Method	BEL	KNO	CRE	TTF	CDR	SER	GPTS	CLO
Single-Agent Zero-Shot	8.36 ± 1.01	5.03 ± 1.38	8.3 ± 0.95	$\mathbf{8.7 \pm 0.53}$	$\mathbf{7.83 \pm 0.83}$	$\mathbf{7.27 \pm 1.39}$	8.93 ± 0.58	$\mathbf{8.57 \pm 1.22}$
AutoGen	8.62 ± 0.96	$\mathbf{5.33 \pm 1.73}$	8.73 ± 1.68	8.6 ± 0.83	7.33 ± 1.79	5.1 ± 1.32	9.03 ± 0.49	6.17 ± 2.2
PerSimChat Base	8.67 ± 0.88	4.6 ± 1.38	8.87 ± 0.43	8.67 ± 0.49	7.67 ± 1.3	4.8 ± 1.56	$\mathbf{9.13 \pm 0.82}$	7.17 ± 1.49
PerSimChat LakSys	$\mathbf{8.79 \pm 0.86}$	4.73 ± 1.86	$\mathbf{9.0 \pm 0.64}$	8.67 ± 0.55	$\mathbf{7.83 \pm 1.21}$	5.0 ± 1.44	8.9 ± 0.84	7.1 ± 1.45

Table B.2: Full comparison of two versions of the PerSimChat framework (GPT-4o model and Lakmoos system) with the baselines (both with GPT-4o model) — Single-Agent and AutoGen. For the dimensions, *BEL* represents believability, *KNO* knowledge, *CRE* credibility, *TTF* turn-taking and flow, *CDR* content depth and relevance, *SER* social and emotional responsiveness, *GPTS* goal progression or task solving, and *CLO* conversation closure.

Method	BEL	KNO	CRE	TTF	CDR	SER	GPTS	CLO
Single-Agent Zero-Shot	8.65 ± 0.82	4.96 ± 1.41	7.96 ± 0.88	$\mathbf{8.75 \pm 0.53}$	8.4 ± 0.81	$\mathbf{7.91 \pm 0.97}$	8.82 ± 0.69	$\mathbf{9.11 \pm 1.01}$
AutoGen	8.67 ± 0.79	$\mathbf{5.33 \pm 1.49}$	8.69 ± 1.51	8.4 ± 1.49	8.15 ± 2.13	6.6 ± 1.73	$\mathbf{9.22 \pm 0.56}$	6.84 ± 2.74
PerSimChat Base	$\mathbf{8.76 \pm 0.59}$	5.05 ± 1.51	$\mathbf{8.78 \pm 0.52}$	8.58 ± 0.59	$\mathbf{8.51 \pm 0.84}$	5.15 ± 1.07	8.91 ± 0.47	8.36 ± 1.19

Table B.3: The full final dimensions evaluation of the PerSimChat framework compared to Single-Agent and AutoGen baselines. *BEL* stands for the believability dimension, *KNO* for knowledge, *CRE* for credibility, *TTF* for turn-taking and flow, *CDR* for content depth and relevance, *SER* for social and emotional responsiveness, *GPTS* for goal progression or task solving, and *CLO* for conversation closure.

Appendix C

Contents of the Attached Data Storage

The attached medium contains the following directories and files:

- `thesis_text/` — directory containing the source files of the thesis text and its versions in `pdf`:
  - `thesis_pdf/` — directory containing both versions of the thesis text in `pdf`.
  - `thesis_src/` — directory containing source files of the thesis text in LaTeX and another files needed for generation.
- `src/` — source files for the *PerSimChat* framework and the eponymous web application.
- `experiments/` — experimental results and the repositories of the external frameworks used during the evaluations.
- `excel_fit/` — directory containing all materials used for the Excel@FIT2025 student conference.
- `README.md` — file containing a description of the data layout in directories on the storage medium and other supplementary information.