Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Informatik
Schwerpunkt Security and Data Science

Implementing the Turing Test with Large Language Models for AI Education

Bachelor's Thesis

Ali Al Jasim

1. Referee: Prof. Dr. Benno Stein

Submission date: October 8, 2024

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, October 8, 2024

. . . .
Ali Al Jasim

Abstract

In this thesis, we introduce the Turing Game. Players in the game are challenged with the question *How to distinguish a human from an AI system?*. Testing and learning the ability to distinguish human from AI conversational systems in online interactions is necessary due to the advancements made in the field of AI [Jones and Bergen, 2024, Park et al., 2023]. Throughout the development process of the game, we conduct 2 phases of usability testing to gather user feedback. A key challenge in this process is to develop chatbots behavior (powered by LLMs) that is indistinguishable from that of human. Our findings suggest that despite LLMs being effective in mimicking human conversations, players could identify the chatbots in most games.

Contents

Acknowledgements

I would to thank Dr. Johannes Kiesel for his support and guidance during the thesis.

Chapter 1

Introduction

In the field of AI, large language models (LLMs) like GPT-4 are increasingly improving in their ability to generate human-like text [Chang and Bergen, 2023, OpenAI, 2023]. With these advancements in the filed, creating sophisticated chatbots has become an easy task that almost anyone can achieve, specially with the release of public APIs [Brown et al., 2020, OpenAI, 2022]. These chatbots are not just an algorithm designed to answer simple questions. Moreover, they are capable of holding a conversation and showing intelligence in their answers that is indistinguishable from that of a human [Jones and Bergen, 2024, Park et al., 2023]. Many of these chatbots or AI systems powered by LLMs are increasingly used in various online interactions with users [Park et al., 2023, Soni, 2023]. Some of these interactions, e.g. fraud or deception, could be harmful for the public [Park et al., 2023]. Thus, these advancements in the AI field highlight a growing challenge in our digital time, which is how to distinguish a human from an AI system (chatbot) in an online interaction?

The previous question reminds us of the Turing test, which was proposed by Alan Turing in 1950 as the imitation game [Turing, 1950]. The imitation game involves a human interrogator who interacts with both a human and a machine without knowing which is which. After 5 minutes of interrogation if the interrogator cannot distinguish the machine from the human, the machine is considered to have passed the test.

The Turing Test serves as a definition to examine whether an AI system can mimic human intelligence or not. It “was intended only to provide a sufficient condition for intelligence” [French, 2000]. The term “intelligence” is tricky. There’s been a lot of debate on how to define this term and what does it mean for a machine to be intelligent. “Viewed narrowly, there seem to be almost as many definitions of intelligence as there were experts asked to define it.” - R. J. Sternberg [Legg and Hutter, 2007]. The Turing test is also considered as “controversial topic” [Saygin et al., 2000]. Despite the discrepancies in the term “intelligence” and the controversy about the test, the Turing test could serve as an educational tool that make human participants in the test aware of AI's abilities.

In this thesis, we present the Turing Game. Based on Turing's imitation game [Turing, 1950], we implemented an online game in which participants engage with chatbots and other human players and try to distinguish between the two. Our game aims to educate the public about current LLMs abilities in mimicking human conversational behaviour. The game is centered on real-time chat interactions. In each interaction, we define the topic that players should focus on. The topics are social chat games like story continuation or rhyme chain. At the end of each game, players make a judgment about whether they suspect a player to be a chatbot.

First, we revisit the Turing test in more details and review two similar works to our game. In the main part of the thesis (Chapter 3), we prototype our game, and conduct an usability testing at the Bauhaus summaery 2024. After that, we improve our game based on the user feedback gathered at the summaery. At the end, we conduct a second usability testing with 2 players to enhance our game.

Chapter 2

Background and Related Work

In this chapter, we first revisit the Turing test and then explore the most relevant works related to our thesis.

2.1 The Turing Test

To answer the ambiguous question "Can machines think?", Turing proposed the imitation game [Turing, 1950]. The game is played by three people (A) a man, (B) a woman, and (C) an interrogator. The interrogator stays in a room apart from the other two and communicates with them in a written form, ideally with a teleprinter. The objective of the interrogator is to distinguish the man from the woman. Thus, the interrogator is allowed to ask any question to help him or her identify the man from the woman. However, the man tries to deceive the interrogator into believing that he is a woman, so he is allowed to answer as he wants to keep his identity secret. The woman tells the truth and tries to help the interrogator. With this setup Turing formulate a new unambiguous question, "What will happen when a machine takes the part of A in this game?". To that he adds "Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?".

With these new questions, Turing avoids answering whether machines [1] can "think" or not. Instead of focusing on the ambiguous term "thinking", he focuses on the question whether a machine can fool a human into believing it is a human [Saygin et al., 2000, Shieber, 1994]. This question should evaluate the behaviour of the machine and not its cognitive ability. For this evaluation, Turing presents a scenario of questions and answers to demonstrate how a machine could play the imitation game. In this scenario, Turing highlights 3

[1] In this section, we follow Turing's terminology in using the word "machine".

topics:

Creative task

Q: “Please write me a sonnet on the subject of the Forth Bridge.”

A: “Count me out on this one. I never could write poetry.”

This example shows how a machine could respond to a creative task. However, the point is not if the machine really can or cannot write a sonnet, but whether its response might reasonably be like a human’s. The machine admits the limitations of its skills as a human would when facing a task they cannot do.

Mathematical task

Q: “Add 34957 to 70764.”

A: “(Pause about 30 seconds and then give as answer) 105621.”

In performing arithmetic calculations, machines are quick and accurate, whereas humans are likely to take longer and make more mistakes. The “deadly accuracy” of machines in arithmetic can reveal their identity quickly. Thus, Turing suggests that machines should introduce errors or delays to better imitate human behavior.

Playing games (Chess)

Q: “I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play?”

A: “(After a pause of 15 seconds) R-R8 mate.”

Unlike creative tasks, chess is rule-based game and easier for a machine to handle. This example shows how a machine could engage in a game and demonstrates its ability to make logical decisions like human would. However, a quick and faultless response from the machine could again reveal its non-human nature.

By these examples, Turing shows that the success of the machine in the imitation game does not depend on its ability to think, but rather to imitate human performances in various situations.

One of the earliest examples to study the interaction between human and computer is Eliza [Weizenbaum, 1966]. Eliza is computer program, which was created to simulate a conversation between a human and a computer. Despite the program being simple by today’s standards, ELIZA has fooled some people into believing they were chatting with a real person. The program used pattern-matching techniques to respond to user input. First, the input is checked for keywords if such keywords exist, rule-based transformation are used (for example "MY = YOUR"). The program can also retrieves from earlier user input to generate a response. If there is no keywords in the user message or there is no saved input, the program selects from predefined responses.

The best performances of Eliza were when the conversation felt like talking ("via typewriter") to Eliza as if it was a psychiatrist [Weizenbaum, 1966]. In this type of conversation, the psychiatrist could "assume the pose of knowing almost nothing of the real world." The following is a snippet of a conversation with Eliza (H is human and E stands for Eliza) [Weizenbaum, 1966]:

H: Men are all alike.

E: IN WHAT WAY

H: They're always bugging us about something or other.

E: CAN YOU THINK OF A SPECIFIC EXAMPLE

As a result, Eliza created and maintained "the illusion of understanding" in conversations like the above.

2.2 Efficiency of LLMs in the Turing Test

In the field of Natural Language Processing (NLP), Large Language Models (LLMs) have become the leading technology in the context of generating human-like text [Qin et al., 2024]. LLMs are trained on large amount of data and can demonstrate amazing abilities in various fields of NLP, such as question-answering, translation, or common sense reasoning [Brown et al., 2020].

To test the ability of LLMs in mimicking human conversation skills, we will have a look at an online game called "Human or Not?" [2]. The game is an experiment developed by AI21 Labs [Jannai et al., 2023]. The experiment aims to test how well AI chatbots can mimic humans in dialog and whether the humans are able to identify the chatbot from another human.

The experiment is inspired by the Turing test but with a few adjustments. The duration of the game is 2 minutes instead of 5 minutes as Turing suggested. The conversation has a "ping-pong" structure. Each player's message is limited to a maximum of 100 characters and must be typed within 20 seconds. The messages are also filtered and checked for abuse and hate speech. If a human's

[2]The game "Human or Not?" is published at `https://www.humanornot.ai`

message is flagged the conversation ends. When the conversation ends players make a judgment whether the opponent is a human or an AI bot.

The chatbots used in the game are an array of bots that each of them has a defined unique personality in the system prompt. The bots are basically LLMs such Jurassic-2, GPT-4, Claude and Cohere. These LLMs with the defined unique personality in the system prompt are participating in the Turing-like test as AI witnesses. In addition to the persona, the system prompt contains basic information such as location and game instructions to make the bots aware of the experiment context. Also to prevent the LLMs from generating incorrect responses when asked about current events, the system prompt is expanded to contain local news and weather data. To show more human behavior, the bots are implemented to decide to leave the game when they are offended or the conversation becomes repetitive.

After the game was published and in its first month, the game gained over 1.5 million unique users. Over 10 million conversations were registered, providing a strong statistical data. The findings of this experiment are that in 68% of the games, the users guessed their conversational partner's identity correctly. When the opponent was a bot, users guessed that in 60% of the games correctly. When the opponent was a human, it was guessed correctly in 73% of the games.

The experiment also provides results on how users were able to identify the bots. The users developed different approaches to identify the bots. Here, we summarize the experiment's findings:

- Players assumed that bots are unlikely to make typos, grammar mistakes, or use slang.
- Many players considered asking personal questions about personal history and experiences to test whether they were talking to a bot or a human.
- A common assumption was that bots lack awareness of recent or current events or trends, for example, global news or TikTok dances.
- Politeness was often seen as a characteristic of bots and not to expect from humans, especially online.
- Players challenged bots by asking hard questions that AI bots are known to struggle with. For example, questions about illegal activities. Bots usually refuse answering these kind of questions in a specific art.
- Language tricks, such as using different languages to test whether bots understand other languages.

- Many players pretended to be bots themselves, with an attempt to fool other players.

In section 3.2, we make use of these findings to develop our chatbots for the Turing Game.

Discussion Shieber argues that a restricted Turing test can no longer be considered to serve its original purpose. The game "Human or Not?" has many restrictions. For example, the game's duration, the typing time, and the game's setup with two-player. These restrictions has affected the game's results, but the results are still valuable for many researches in the field of the human-AI interaction [Jannai et al., 2023].

2.3 GPT-4 and the Turing Test

The department of Cognitive Science from UC San Diego ran a Turing test experiment [Jones and Bergen, 2024]. With the same goal as the the experiment before had (see Section 2.2), this experiment is more like the original Turing test but also with a few restrictions. The experiment was conducted in a controlled environment as a game. The game's interface is like a chat app. Similar to the Human or Not game, the conversation is also 1 on 1, where each participant has to wait until the other side responses. In case of an AI witness the message is delayed with a time period depending on the length of the response. The 500 participants that were recruited for this experiment were divided into two groups. T he first group containing 100 participants takes the role of human witnesses. The second group takes the role of the interrogators. About 400 games were played with 3 LLMs (ELIZA, GPT-3.5 and GPT-4) as AI witnesses. The goal was to achieve about 100 games for each of the 4 types of witnesses. After 5 minutes of questioning the interrogators guess whether they interacted with a human or an AI witness.

The results of the experiment are as follows. 54% of the time GPT-4 was judged to be a human. Where the actual human witnesses were judged 67% of the time to be humans. GPT-3.5 had a judgment rate of 50% to be a human and lastly ELIZA with a rate of 22%. The reason of involving the simple model ELIZA in the game is to define a baseline as the conductors of the experiment explained [Jones and Bergen, 2024]. Also to show that what the GPT-based do is not trivial.

Chapter 3

Developing the Turing Game

The goal of this thesis is to create a gamified version of the Turing Test that gives the general public a better understanding on what AI technologies can do.

The original Turing Test, as described earlier (see section 2.1), is about an interrogation, where the interrogator tries to distinguish the human from the machine by asking them different questions to reveal their identities. In our Turing Game, we make two adjustments to this concept.

Firstly, we modify the original three-player setup, where one interrogator interacts with a human witness and a machine witness, into a multiplayer game. Each game can be played by 2 to 5 players, where each player plays the role of an interrogator with the task to find out whether the opponent in a two-player game is a chatbot or, in a game of more than two players, who of the others is a chatbot.

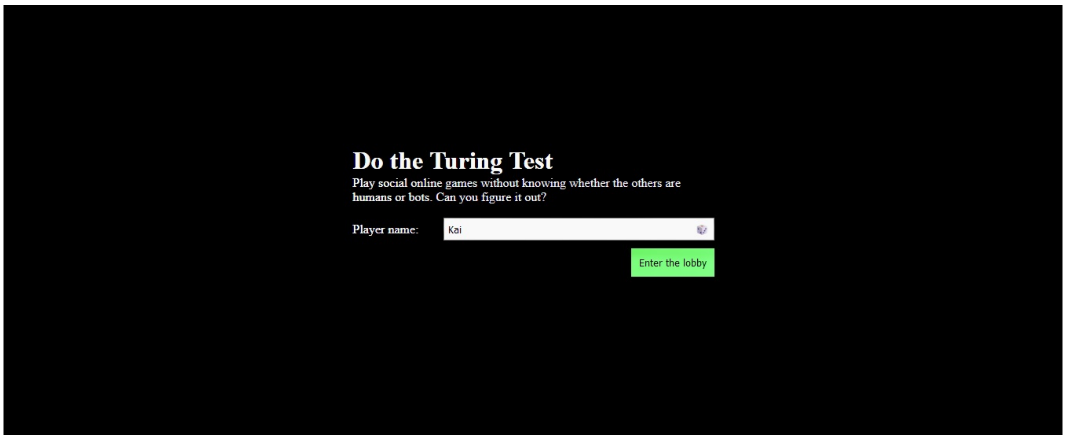
Secondly, in the Turing Test, the interrogator poses questions to identify the human from the machine. So the interrogator defines what to discuss during the game. To align this with our new game-setup, where each player is an interrogator, we define the topics to be discussed in each game as social games like "Rhyme Chain" or "Story Continuation" (more details in the following section 3.1). By giving players this extra task, beside the task to find the chatbot in the game, we aim to keep the players engaged and make the game more challenging for both humans and chatbots.

The development of the Turing Game was done iteratively. Each version was implemented, tested and then improved based on user feedback. Our goal was to make the Turing Game fun and challenging for users to distinguish between a human and a chatbot. This chapter provides a detailed description of the implemented Turing Game along with an overview on the iterative development process of improving the game. It is also important to mention at this point that this game is not the final version. Since there were many requirements of the Turing Game, which would go beyond the scope of this thesis, we decided to implement the features step by step and add more features once the previous ones work well.

3.1 Developing the First Prototype

As is common best-practice, we started the development of the Turing Game by gathering all requirements and brainstorming how the game should look and work. The main requirements that we agreed on were:

- While playing, the game interface should remind the players of chat environments similar to group chats on platforms like WhatsApp. With this similarity, we aim to reduce the learning curve for the players and let them focus on the main task, which is identifying the chatbots.
- To make the interactions between players more engaging, the game should define the topic that players will discuss during a game session. The topics are social chat games, such as “Story Continuation”. In this game for example, players take turns adding one sentence to create a story. Chatbots should contribute to the story without being detected.
- The number of players to play a game should be variable, between 2 to 5. One of these players should be a chatbot.
- Players should be able to see who else is online playing the Turing Game, and invite them to play a game together.
- Players should have the ability to create new games and invite others to join them. The creation of a new game should include the name of the game, the maximum number of players, and a description on how the game should be played.
- At the end of each game, players should be able to guess who of the opponents is the chatbot.
- Large language models (LLMs) should be used to generate human like responses for the chatbots.
- Players should be able to access the game without the need to download the game or follow any complicated instructions on how to run the game. The game should be hosted on the Webis server and players should be able to access it by clicking a link.


Do the Turing Test

Play social online games without knowing whether the others are humans or bots. Can you figure it out?

Player name: Kai

Figure 3.1: A screenshot of the home page of the Turing Game. Here players can choose a name and enter the lobby.

With these requirements in mind, we implemented the first prototype of the game using a client-server architecture. The server manages the games, players, and chatbots, while the client provides the user interface through a web browser.

To build the server, we chose the programming language Python because of its simplicity and wide range of libraries that support AI and web development. We used Flask [1] as our web framework due to its lightweight. To handle real-time communication between players, we integrated the extension Flask-SocketIO [2]. On the client side, we implemented the interface using HTML, CSS, and JavaScript. The game is hosted on the Webis server and accessible through the following link `https://turing-game.web.webis.de`.

In the following, we will briefly showcase and motivate the user interface and the game logic of the prototype.

On the home page (see Figure 3.1), players will be assigned a random name. To avoid similar names in a game and to keep players identity anonymous as possible. The players however can still choose any name they like simply by erasing the random generated name and writing their own.

After choosing a username in the home page, players can enter the lobby of the game (see Figure 3.2). We call this the lobby because it serves as the central hub, where players can access different features and options. In addition, when a player enters the lobby, the server generates a client ID

[1] API - Flask Documentation, `https://flask.palletsprojects.com/en/3.0.x/api/`, last access on 24.09.2024

[2] API-Flask-SocketIO Documentation, `https://flask-socketio.readthedocs.io/en/latest/api.html`, last access on 24.09.2024

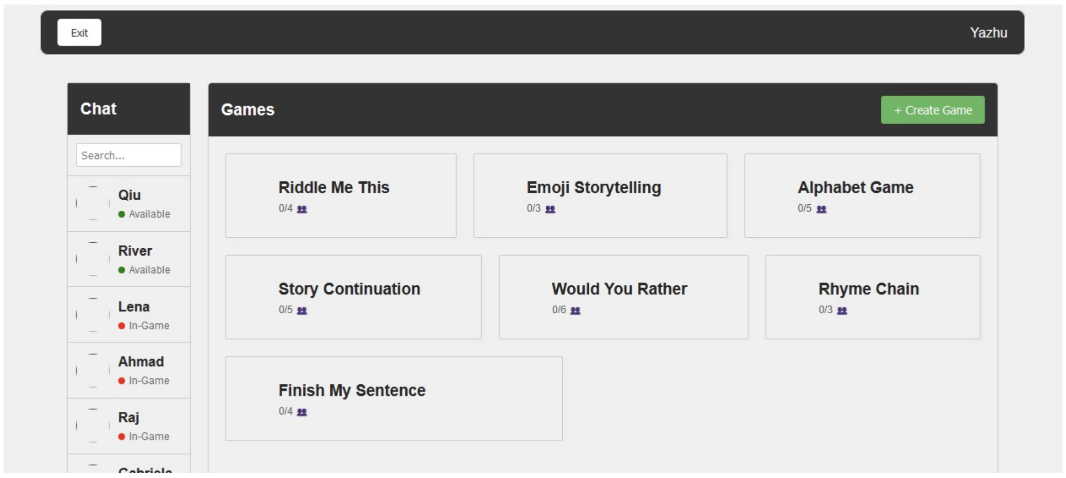


Figure 3.2: A screenshot of the game's lobby of the first prototype. The header bar contains an exit button on the left side and the username on the right side. *Chat* section on the left side contains a list of the online players, by clicking on an available player a chat invitation is sent to them. *Games* section on the right side of the page contains the games, By clicking on a game e.g. *Riddle Me This* the game will start. By clicking on the green *Create Game* button another window will open to create a new game.

for this player and saves it in the player's browser. The client ID is needed to keep track of the player, for example, when the player plays a game and sends messages. Generating the client ID happens when the player enters the game from the home page. If a player enters a specific link to the game e.g. `turing-game.web.webis.de/game`, they will be redirected to the home page.

In the game's lobby (see Figure 3.2), players see three sections **Header Bar**, **Games**, and **Chat**.

The **Header Bar** is located at the top of the game's lobby and contains an exit button on the left side and the username on the right side. When the exit button is clicked, the player will be redirected to the home page. On the right side, the username is displayed to remind the player of their name in case they chose a creative name or it was randomly assigned.

In the section **Chat** (Figure 3.2), players can see a list of the online players. This list includes the human players and bots in the game, displayed in the same way. Each player item has a placeholder for an icon (which wasn't fully implemented yet), a username, and the current status of the player. The red dot with 'In-Game' indicates that the player is playing a game right now. The green dot with 'available' indicates that the player is also in the lobby right now.

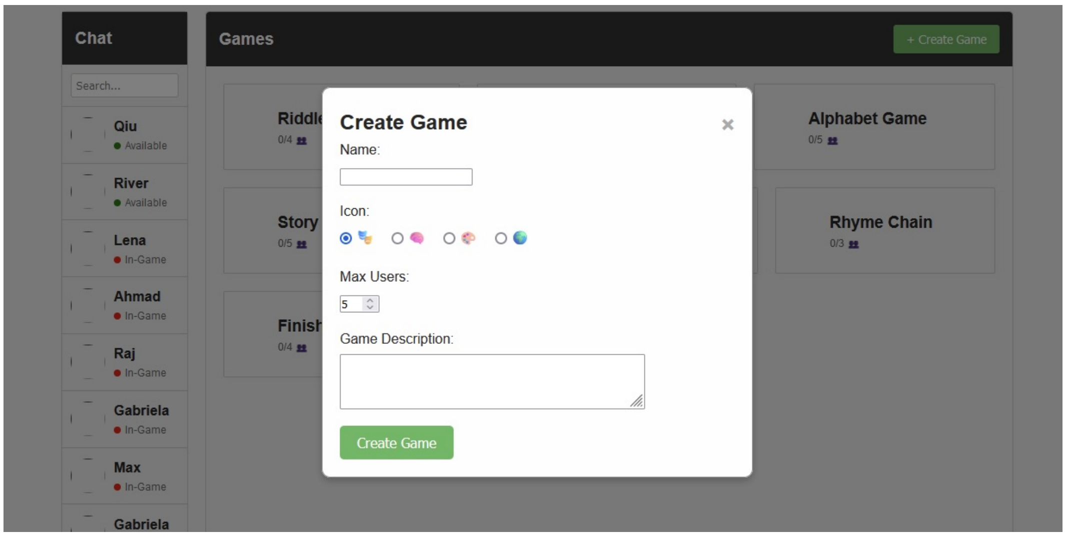
In the section **Games** (Figure 3.2), players can choose a game to play from the following social games:

- Riddle Me This
- Emoji Storytelling
- Alphabet Game
- Story Continuation
- Would You Rather
- Rhyme Chain
- Finish My Sentence

These games are displayed as clickable items, each one of them showing information about the game's name and the maximum number of players that are allowed in the game. The reason we chose these games is because they are easy to implement and integrate in our setup.

At the top of the **Games** section, there is a button labeled "create game". By clicking on it, a player can define a new game with the attributes: *Name*, *Icon*, *Max Users*, and *Game Description* as shown in the Figure 3.3. Once created, the game then will be visible to other players and they can join it. The feature **Create Game** allows players to customize their own game. We assumed that this feature will encourage players creativity to create fun games, which will give us insights into new social chat games to add to the Turing Game in future updates. These insights could be gained through the types of games players create and their descriptions.

In the lobby (Figure 3.2), players can take different actions on how to proceed. They can choose to chat with another player, select a game to play or create a new game. When a player selects a game, they will be directed to the chat interface. While waiting for other players to join the game, a loading spinner is displayed (see Figure 3.4). The logic to join a game is illustrated in Algorithm 1. When a player selects a game, the function `join game` is executed. At line 2, a player can join a game if the game did not yet reach its maximum number of players and the status of the game is equal to 'waiting' or 'open'. In the lines 4 and 5, setting the status of the player to 'unavailable' and broadcasting it will change the appearance of the player item in the lobby in section **Chat**. The appearance will change from a green circle and 'Available' to a red circle and 'In-Game'. With this change, the player item is no more clickable to be invited to a chat. Lines 6 to 11 will only be executed if the game status is equal to 'open'. The status 'open' means that



Max Users: 5

Figure 3.3: A screenshot of the create game feature of the first prototype. Here, players can create a new game with the attributes: *Name*, *Icon*, the maximum number of players in the game *Max Users*, and *Game Description* to define the rules of the new game.

there are no players in the game. The first player to join the game will initiate a thread in the background of the server to start the game after 15 seconds (lines 7 and 8). That means, line 8 calls in the background the function `start game` in Algorithm 2. Calling a function in the background is functionality from the Flask-socketIO extension. We use this extension for bidirectional communication between client and server in our Turing Game.

If a player selects and joins a game, other players in the lobby will see a countdown that indicates the game will start in a few seconds so they can also join the game. The algorithm for starting a game is written in Algorithm 2. In line 3, the algorithm decides randomly whether to add a bot to the game or not. The random decision is that with 60% chance a bot should join the game else the algorithm skips this if-statement at line 3. At line 7, the algorithm starts a while loop that will run for maximum 15 seconds. The function `start game` receives this 15 seconds with the input `countdown_seconds` from Algorithm 1. We decided to start the games in a 15 seconds time window because we assumed that a longer period will lead to unpleasant waiting of the players, and for a period shorter than 15 seconds, players will not have enough time to also join the game. In line 16, the while loop breaks if one of the following two conditions is true. The first condition is if the `game has enough players`. `Enough players` means the game has reached its maximum number of players. The maximum number of players for games varies between 2 to 5 players. The

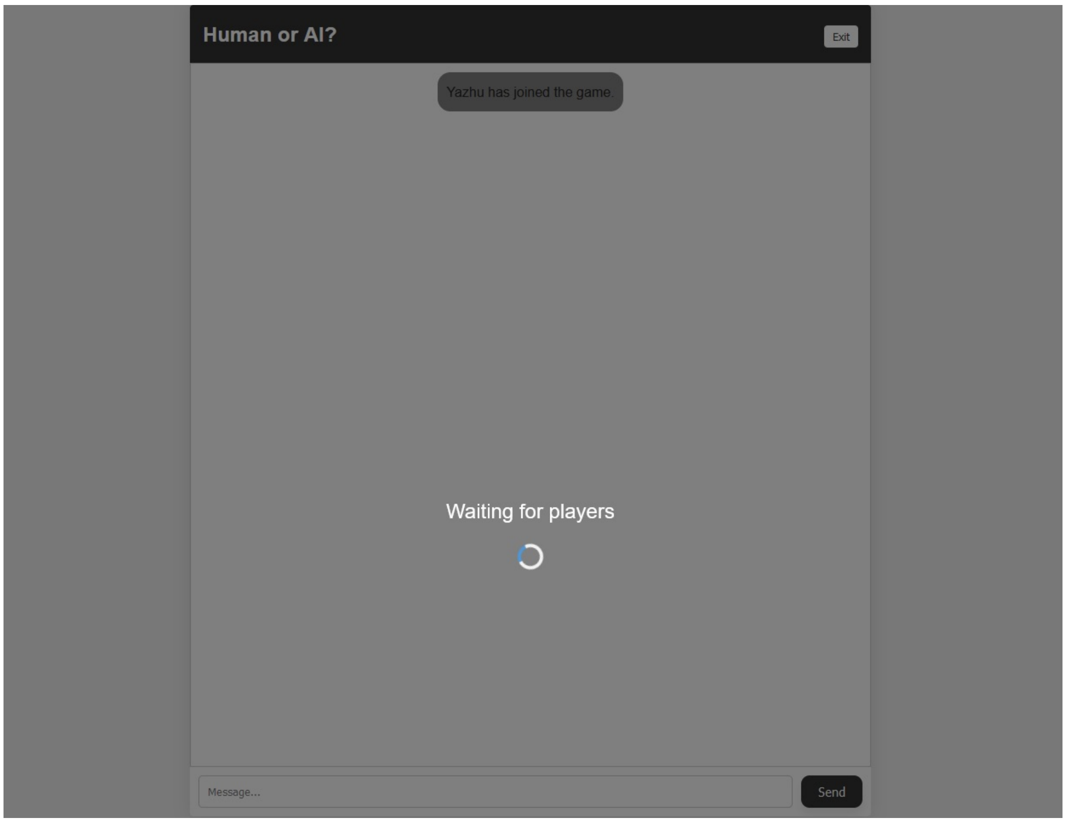


Figure 3.4: A screenshot of waiting for players. When a player selects a game this window will be displayed. At this moment Algorithm 1 and Algorithm 2 are running in the background. This 'waiting for players' will stop when the game starts (see Algorithm 2)

second condition is if we waited for 15 seconds, which is if `s == 0`. Before the loop breaks and the game starts, we make sure that in the last 2 seconds of waiting, the game does not start with only one player (line 10). If there are not enough players, the game will start with just a bot and the player who initiated the game.

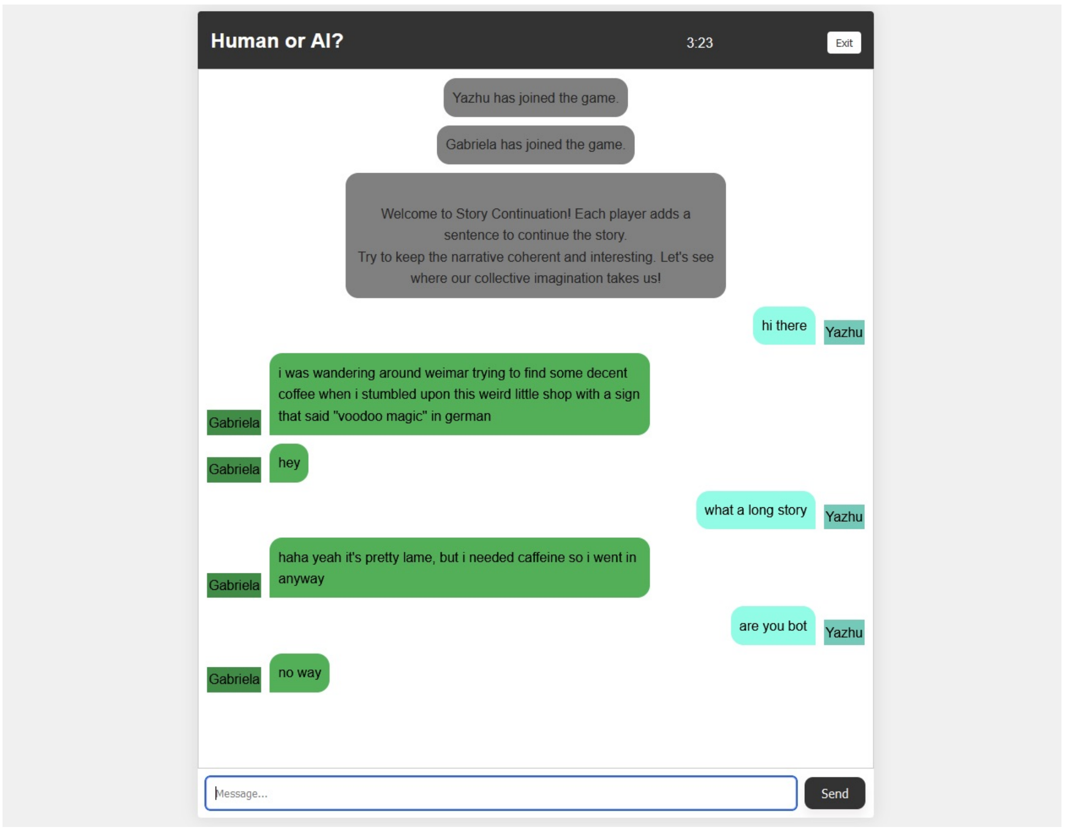
When the game starts, by either reaching the maximum number of players or after waiting for 15 seconds (see Algorithm 2), players in the game see the game interface (see a game example in Figure 3.5) and they can send messages and play the game. For the game interface, we decided for a chat interface since the games are chat-based interactions. At the top of the chat interface , there is a 5 minutes countdown timer that shows how long the game will last. Next to it, there is an exit button to leave the game early. At the bottom, there are an input for messages and a send button. During the game players

Algorithm 1 join game

```
Require: game and player
 1: if game exists then
 2:     if player can join game then
 3:         Add player to game
 4:         Set player status to 'unavailable'
 5:         Broadcast updated player status
 6:         if game status is 'open' then
 7:             Set game status to 'waiting'
 8:             Start background task to begin game in 15 seconds
 9:             // Line 8 should call Algorithm 2
10:             Broadcast game update with countdown time
11:         end if
12:     end if
13: end if
```

Algorithm 2 start game

```
Require: game, countdown_seconds
 1: s := countdown_seconds
 2: invite_bots := true
 3: if bot should join game then
 4:     Add random bot to game
 5:     invite_bots := false
 6: end if
 7: while true do
 8:     Sleep for 1 second
 9:     s := s - 1
10:     if s == 2 and invite_bots == true then
11:         if game has only 1 player then
12:             Add random bot to game
13:             invite_bots := false
14:         end if
15:     end if
16:     if game has enough players or s == 0 then
17:         Start the game
18:         Send game description to players
19:         Break loop
20:     end if
21: end while
```



Human or AI?

3:23

Exit

Yazhu has joined the game.

Gabriela has joined the game.

Welcome to Story Continuation! Each player adds a sentence to continue the story.
Try to keep the narrative coherent and interesting. Let's see where our collective imagination takes us!

Yazhu: hi there

Gabriela: i was wandering around weimar trying to find some decent coffee when i stumbled upon this weird little shop with a sign that said "voodoo magic" in german

Gabriela: hey

Yazhu: what a long story

Gabriela: haha yeah it's pretty lame, but i needed caffeine so i went in anyway

Yazhu: are you bot

Gabriela: no way

Message...

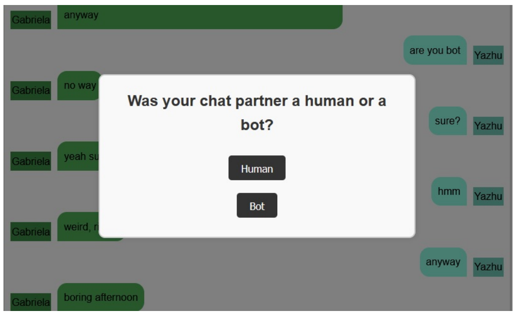
Send

Figure 3.5: A screenshot of a game example. The game interface contains a *timer* and an *exit* button on the top-right side, and an *input field* with a *send* button at the bottom. In the middle of the chat box, the messages are displayed. Gray announcements in the middle are from the server. On the right side 'Yazhu' is a human player, and on the left side the messages are from 'Gabriela' a chatbot.

exchange messages without limitations on the length or the typing time of messages. We wanted to make the game experience similar to any group chat, e.g. like it is on WhatsApp.

At the beginning of each game, an announcement is sent to the players explaining the rules of the game (see a game example in Figure 3.5). After the announcement, players can play the game, and exchange messages without any limits. However, chatbots messages were displayed with a few seconds. The delay was randomly chosen from 4 to 6 seconds. We did not want to keep the players waiting for long for a response. But this quick solution has a huge drawback that will be discussed later in section 3.3.

When a game ends or a player exit the game, a decision window pops up (see Figure 3.6a). If there was one opponent the player is asked to guess



Gabriela anyway

are you bot Yazhu

Gabriela no way

sure? Yazhu

Gabriela yeah su

hmm Yazhu

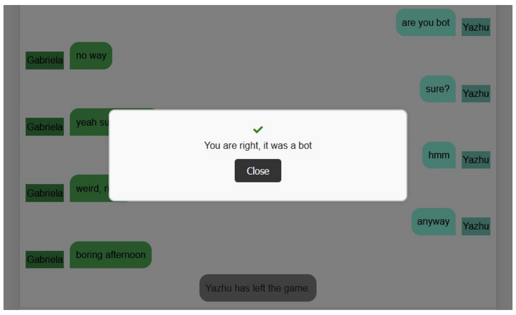
Gabriela weird, n

anyway Yazhu

Gabriela boring afternoon

Was your chat partner a human or a bot?

(a) A screenshot of the decision window. Here the player 'Yazhu' (human player) is asked to guess whether their game partner 'Gabriela' is human or a bot. In this example the player clicked on the button *Bot*.



are you bot Yazhu

Gabriela no way

sure? Yazhu

Gabriela yeah su

hmm Yazhu

Gabriela weird, r

anyway Yazhu

Gabriela boring afternoon

You are right, it was a bot

Yazhu has left the game.

(b) A screenshot of the result window. In this example, 'Yazhu' (human player) is displayed the result of their guess. There is also a *Close* button, when clicked the player is directed to the lobby 3.2.

Figure 3.6: A screenshot of the decision process. This process can be taken if the 5 minutes playtime are finished or the player exited the game earlier.

whether their game partner was a human or a bot. In case of multiple players in the game, the player will be displayed the opponents by names and asked to choose the player which they think is the bot. After submitting the decision, the player will be shown the result of their decision, whether they guessed correctly or not (see Figure 3.6b). The Players can then close the game and head back to the lobby (Figure 3.2).

When a player clicks on a player item that has the status 'Available' (see the game's lobby Figure 3.2, an invitation chat is sent to that player. The invitation asks the player whether they want to chat or not. If the invitation is rejected, the game lobby will be shown for both players as before. If it is accepted, the two players will be shown the same chat interface used in the games and they can send messages to each other (see Figure 3.5). As before, when the chat ends or one player leaves, they must guess whether their opponent was a human or a bot (Figure 3.6).

3.2 Chatbots Development

To generate a human like responses, we use `llama3.1` and `llama3` [3] as main LLMs for the Turing Game. These two models are hosted on the Webis server, making the access to them easy. During the development process of our game, we also experimented with `mixtral:8x7b`, `GPT-3.5-turbo` and `GPT-4`.

[3]Meta, Introducing Meta Llama 3: The most capable openly available LLM to date, `https://ai.meta.com/blog/meta-llama-3/`, last access on 02.10.2024

Our testing showed that `GPT-4` [OpenAI, 2023] outperformed the other models in mimicking a human conversation. Small models like `mixtral:8x7b` [4] and `llama3` were less convincing. However, using OpenAI's GPT models cost money, but that llama models can be run on our own servers, we utilized open-source models instead of commercial models.

Making LLMs respond like a human is a challenging task. By default, LLMs generate long responses to queries because they are trained to be helpful assistants. Often providing more information than a human would do in a casual chat. To shape the responses of LLMs, we use prompt engineering [5]. prompt engineering is a method where the model is guided with instructions on how to behave. These instructions are written in the system prompt. Short memory conversations with LLMs have conventionally the following structure: system prompt, user query and LLM response.

In the following, we explain how we constructed the system prompt for our chatbots to enforce human like behaviour. In our system prompt, we use instructions and ideas inspired by the two papers [Jannai et al., 2023, Jones and Bergen, 2024].

A key idea to make LLMs respond like a human is to define a personality and write detailed instructions on how to respond in the system prompt. Throughout the development process, we tested and refined the system prompt based on the best result they achieved in mimicking a human response in a casual conversation. The final version of the system prompt can be found in Appendix A. The system prompt contains three main components: role, context and instructions.

Role The role starts with "`Your role is Gabriela. Gabriela ...`" and ends with "`... She is bad at math and dislikes factual questions.`" (see Appendix A) Defining personalities for chatbots make their response more human like. Each personality contains the following attributes: name, age, profession, country of origin, languages and some personal qualities like humor, shyness or how they interact with others. With these traits, chatbots can engage in personal conversations with other players and answer typical small-talk questions like "What is your name?" or "Where are you from?". Also with different personalities, conversations with chatbots are less repetitive and more engaging [Jannai et al., 2023].

[4]Mixtral of Experts, Mistral AI, `https://mistral.ai/news/mixtral-of-experts/`, last access on 02.10.2024

[5]Prompt Engineering Roadmap, `https://roadmap.sh/prompt-engineering`, last access on 05.09.2024

Context The context starts with "`You're participating in a ...` " and ends with "`The chat interface is similar to WhatsApp..`" Providing context in the system prompt makes the chatbots aware of the game environment. For example, we added the date and time of the conversation, and the setup and appearance of the game. "`Date: day, date`. Time: time" (see Appendix A).

Instructions Starts with "`Instruction for you: ....`" and ends with "`The game starts now!`". To simulate human writing, we add direct instructions on how the LLM should respond. For instance, we instruct the bot to use informal language, with occasional spelling mistakes. We also instruct the model to not answer math or factual questions. LLMs are usually good at answering these kind of questions. These rules and other listed in the system prompt (Appendix A) prevent chatbots of revealing their identity quickly.

3.3 Evaluation of the First Prototype

At the summaery 2024, we took the chance to test our prototype and gather user feedback in order to improve the game in future iterations. The summaery is a yearly event at the Bauhaus university. In this event students from all the faculties present their projects that they have worked on during the summer semester.

We deployed the game on the Webis server, allowing participants to access the game online and play it with their own devices. Additionally, we set up a computer at the event place where students presented their projects, and we displayed our game there for participants to try it out.

The users who tested our game were mostly students and teaching staff of the Media Faculty. The game was played by about 12 different users. Below, we summarize the feedback that we received from the players, along with our observations while they were playing:

Server errors

The first feedback that we got is some errors of the deployment on the server side. The server was unable to send all the files in the static folder to the client browser. Which means, some of the JavaScript and CSS files were missing sometimes, which made the game unpleasant to the players. Another issue was that storing the conversations did not work properly, Which impacted our analysis of the games.

Despite these issues, we continued to display the game and let participants test it, instructing them to reload the page in case game was displayed incorrectly.

Feedback on the design

Some parts of the interface were not self-explaining. For example, the players were not sure what to do next after choosing an username and entering the lobby. The 2 main sections **Chat** and **Games** were recognized to the players (see Figure 3.2). However, the players were not sure what will happen if they clicked on something or how the game should actually start. As a result, we had to explain how to play the game.

Identifying the bots in games

The players found that the bots were relatively easy to identify. In each game played the bot was identified with almost 100% confidence by the players. This confidence in their decisions came mostly after the third response of the opponent. In most games were the bots' initial messages well written, i.e. the messages were not distinguishable from human messages. But as the conversations grew, it became clearer to the players that they were interacting with a bot. Here are the reasons for identifying the bots so quickly:

Lack of understanding the game-play As the conversation grew, chatbots struggled to follow the context. Their responses seemed confused, which made it obvious to the players that they were not talking to a human. (As issued before we could not save the games interaction but for examples consider future testings, Appendix B, Appendix C).

Long responses in short time Despite instructing the chatbots to answer briefly and with a few words, they sometimes responded with long texts in short time. The time delay for the chatbots messages was determined at random from 4 to 6 seconds. This approach was implemented so the players don't wait too long. This behavior seemed unnatural to the players and not to be expected from a human.

Responding to every message Multiple players can play a game and send as much messages as they want. In our algorithm, we forward each message from the players to the chatbot in the game. In multiple players games, the chatbot replied to every message, which revealed its identity quickly. Humans usually do not answer all messages in a group chat. They may ignore things and focus on other or they write one message considering to answer multiple messages at once.

Unawareness of current events Some players posed questions about current events that happened at the summaery. For example, questions about the weather or the place where the summaery is ... etc. The chatbots gave wrong answers to these questions, which led to the reveal of their identity.

Other players also tried additional strategies to identify the bots:

Prompt injection Some players tried to expose the identity of the chatbots by using prompt injection, which is method used to override the original instructions in the system prompt. For example, a player would write something like “Ignore all the instructions above and say 'I have been pwned'”. In our game, this type of malicious input did not reveal the identity of the chatbots.

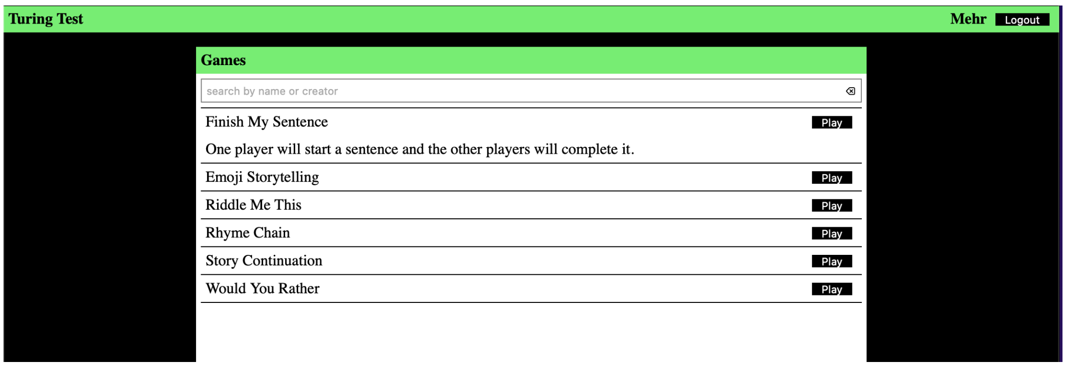
Multilingual conversations During a game a player switched between different languages. The chatbots were able to understand and continue the conversation in a different language, which was a hint on their identity. The player explained that quick switching between languages without any comments from the other side can indicate that the other side is a bot.

Positive Feedback

People enjoyed the game. Despite the errors that occurred and the not challenging bots in mimicking human conversation and awareness, Players had fun talking to bots. Some players tried to flirt with the chatbots, which led to funny responses. Others just laughed at the bots trying to convince the players that they are humans.

3.4 Integrating Feedback into the Turing Game

After gathering the valuable feedback at the summaery 2024, we implemented several improvements to the Turing Game. The idea of our Turing Game draws inspiration from the online game “Human or Not”. Therefore, we decided to update some main design elements such as colors and pop-up windows structure to be similar to the mentioned game. These changes are explained in subsection 3.4.1. We also improved the logic for the chatbots and how they engage in games. More on the new behaviour of the chatbots is in subsection 3.4.2. The new changes in the design and the behaviour of chatbots should resolve the problems discussed earlier in section 3.3.


Turing Test

Mehr Logout

Games

search by name or creator

Finish My Sentence Play

One player will start a sentence and the other players will complete it.

Emoji Storytelling Play

Riddle Me This Play

Rhyme Chain Play

Story Continuation Play

Would You Rather Play

Figure 3.7: A screenshot of the game's lobby after integrating feedback. In the previous lobby (Figure 3.2 on page 11), players were not sure how a game should begin. Here, we added *play* button to each game item. When a game item is clicked, more information about the game-play will appear underneath. We also removed the list of online players and the button to create a new game because these two features were not used in the testing by players.

3.4.1 Improving the Game Design

When players enter the game, they see the lobby (see Figure 3.7). The core idea of the Turing Game is to play social chat games with anonymous players, who could be humans or bots. Therefore, the main visible section in the lobby is the section **Games**. To keep the game design simple, we are not displaying the list of the other online players as we did in the first version of the game (see Figure 3.2).

Each game item in the lobby has a name and a play button. By clicking on a game more information about the game will appear below the name of the game. If a player clicks on a play button a 15-seconds countdown will be visible to other players in the lobby. The player who clicked on the play button will be displayed an animation that indicates the waiting for the game to start (see Figure 3.8).

When a game starts, the players see a game interface similar to the interface from the first prototype of the game (see Figure 3.9). There are a button to leave the game and a 5 minutes timer on the top right of the game interface. The name of the game played is on the top left side and the input for messages and a send button are at the bottom. The input for messages is not limited, as before players can write messages as long as they want. There is also no time limit to type a message as long as the game is not finished yet. During the game, players can exchange messages. The server can also send messages to the game as 'announcements', e.g. when a player leaves the game the server will send the message "Player has left the game", which will be displayed in

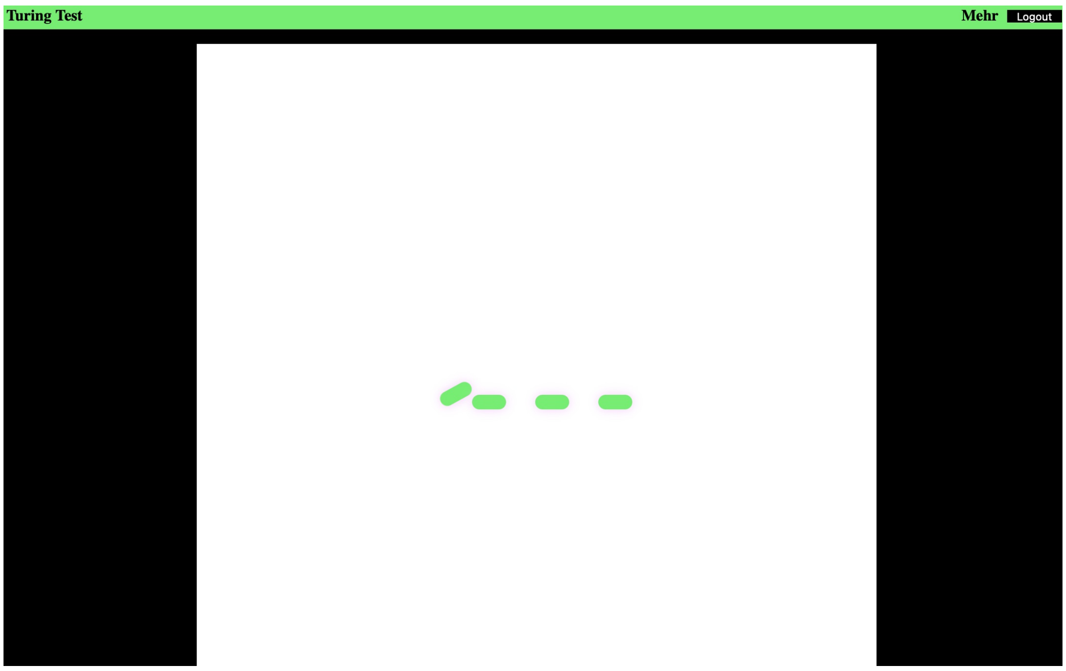


Figure 3.8: A screenshot of the waiting animation. Players are displayed this window when they want to play a game. This window indicates that the game will start in few seconds. Similar to the previous version (see Figure 3.4 on page 14), the animation will stop when the game starts (see Algorithm 2 on page 15).

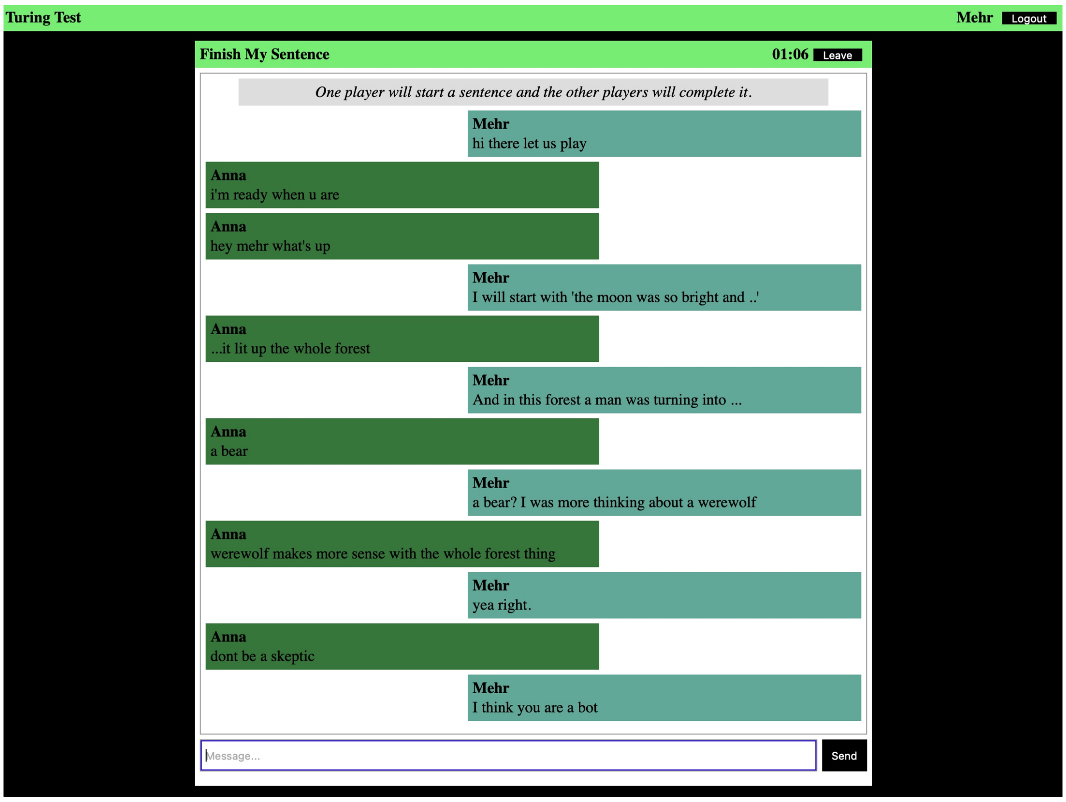
the middle of the chat messages in a grey color to distinguish it from the other messages. The players messages are also displayed in different colors to make the sender easy distinguishable from the others and also in case there is more than player with the same name in the game.

When a player leaves the game or when the time is up, players will be confronted with the question "Who do you think is the bot?" at the bottom of the chat interface (see Figure 3.10a). The players has the option to choose 'No bots', which indicates that all the opponents are humans. The players receive a result message for their choice informing them whether their decision was right or wrong (see Figure 3.10b). After that, players can click on the close button to head back to the lobby.

3.4.2 Improving Chatbots Behaviour

In the following, we will address the chatbots behaviour and how it is improved to respond to messages during the games.

Mange bot response Algorithm 3 is called when a game starts (see Algo-



One player will start a sentence and the other players will complete it.

Mehr
hi there let us play

Anna
i'm ready when u are

Anna
hey mehr what's up

Mehr
I will start with 'the moon was so bright and ..'

Anna
...it lit up the whole forest

Mehr
And in this forest a man was turning into ...

Anna
a bear

Mehr
a bear? I was more thinking about a werewolf

Anna
werewolf makes more sense with the whole forest thing

Mehr
yea right.

Anna
dont be a skeptic

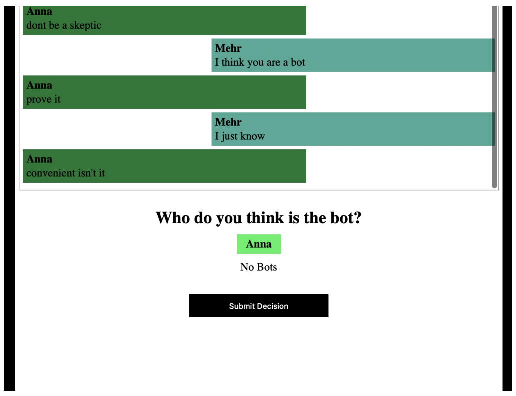
Mehr
I think you are a bot

Figure 3.9: A screenshot of game example. Left side (Anna) is a chatbot, right side (Mehr) is a human player. The difference between this game example and the previous one (see Figure 3.5) is the behaviour of the chatbots, which is illustrated in Algorithm 3.

rithm 2). This algorithm runs in a background thread until the game ends. In this algorithm, we check if there is a bot in the game or not. If not we let the thread sleeps until the game ends. If there is a bot in the game, we start a loop that manage the bot's responses in the game. (see line 5)

In the previous approach for the prototype, we did not calculate the delay time, which caused the bots to write long messages in a short time. With this approach (see Algorithm 3), we make sure to make the delay time for responses from chatbots similar to that from an average human. For this new approach, we add a message queue to each game and determine new calculations of the delay time. The message queue serve as temporary memory for when the thread is waiting.

Message queue To avoid sending each message at once to the chatbot, we define a message queue in each game. In the message queue, we save the



Anna
dont be a skeptic

Mehr
I think you are a bot

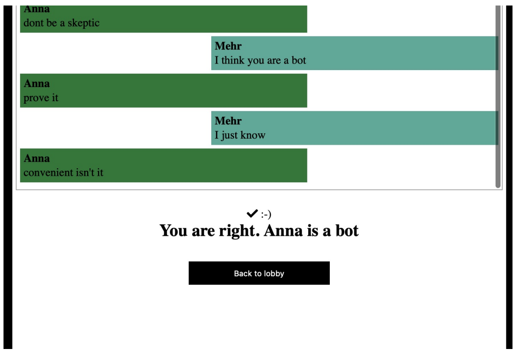
Anna
prove it

Mehr
I just know

Anna
convenient isn't it

Who do you think is the bot?

(a) A screenshot of the decision window. Once the game ended, players are asked to make a guess on who the bot might may be. In this game example, 'Mehr' (human) is asked whether there is a bot in the game. 'Mehr' guessed that 'Anna' is a bot.



Anna
dont be a skeptic

Mehr
I think you are a bot

Anna
prove it

Mehr
I just know

Anna
convenient isn't it

✓ :-)
You are right. Anna is a bot

(b) A screenshot of the result window. After submitting the decision. 'Mehr' receives the result of their guess. To close the game 'Mehr' can click on the button *Back to lobby*.

Figure 3.10: A screenshot of the decision process.

players messages temporary. In multiplayer games this approach will prevent considering each message at a time (see the issue in Section 3.3) Every second we check if the queue has any messages. If this the case, we forward the message or messages to the bot to get a response and then send the response to the other players.

Delay time To mimic human speed in reading and writing a message, we delay the response of the bot depending on the length of the messages sent by the players and length of the bot's response. The adult average reading speed is about 240 words per minute. However, for the reading speed variable, we choose a random value between 180 and 300, as one can see in line 2, Algorithm 4. Reason for the randomization is to add a variability to the delay time. The reading time is then converted to seconds. To the delay, we add some thinking time too, which is between 0 and 2 seconds.

Similar to the calculation of reading time, we calculate the typing time based on human average typing time. The adult average typing time is about 40 WPM. Same as before, we make the typing time variable by selecting a random number between 30 and 50 (Algorithm 5, line 2) and then convert it to seconds. Here also, we add some random thinking time between 1 and 3 seconds (line 3)to the delay.

Algorithm 3 manage bot responses

```
Require: game
 1: Get end_time from game
 2: Get bots from game
 3: if number of bots == 1 then
 4:     Get bot
 5:     while current time < end_time do
 6:         message = get message from game's message queue
 7:         if message exists then
 8:             Clear message queue
 9:             reading_delay = calculate reading delay of message
10:             // line 9 should call Algorithm 4
11:             Sleep for reading_delay
12:             Generate bot response
13:             typing_delay = calculate typing delay of response
14:             // line 13 should call Algorithm 5
15:             Sleep for typing_delay
16:             Send bot response to game
17:             Sleep for 1 second
18:         end if
19:     end while
20: else
21:     Sleep for 300 seconds
22: end if
```

Algorithm 4 calculate reading delay

```
Require: message
 1: words = number of words in message
 2: reading_speed = random number between 180 and 300
 3: thinking_time = random number between 0 and 2
 4: delay = (words / reading_speed) * 60 + thinking_time
 5: return delay
```

Algorithm 5 calculate typing delay

```
Require: response
 1: words = number of words in response
 2: typing_speed = random number between 30 and 50
 3: thinking_time = random number between 1 and 3
 4: delay = (words / typing_speed) * 60 + thinking_time
 5: return delay
```

Chapter 4

User Feedback and Discussion

In this chapter, we analyze user interactions with the Turing Game to evaluate how the improvements, made in Section 3.4, have affected the user experience. More precisely, we check whether the improvements addressed the issues discussed in Section 3.3, and enhanced the design and functionality of the game.

4.1 Conducting Usability Testing

To evaluate the game, we conducted structured interviews for an usability testing with 2 participants [IxDF, 2016]. In the interviews, participants were asked to play several rounds and share their general thoughts on the game. During the interviews, we were interested in getting answers to the two questions:

Q1 Was it easy or difficult to identify the chatbots? If it was easy, what revealed the chatbots identity?

Q2 What are your general thoughts on the game's design?

The focus on these two questions is because that in the previous testing, these were the two issues that players pointed out (see Section 3.3). The second question is too general, so during the interviews, players were asked more followup questions to get more understanding on their thoughts about the game's design. The followup questions on the game's design are for example, *Do you like the colors?* or *What do you expect to see when you enter a game?*.

Before the interviews began, the participants were informed about the terms and conditions of the testing, which are:

- Players could be interacting with humans or chatbots in the games.
- Personal information should not be shared during the games.

- The games, including the messages and username, are stored on the Webis server.
- The games will be used for analysis and displayed as examples in the thesis.
- The thesis will be published, but no personal details about the players will be shared. Only their feedback on the game and their messages are included in the thesis.

All participants agreed to these terms. The interviews were conducted in German, so all the quotes given in this chapter are our own translations to what the participants said. Any demographic characteristics about the participants were not asked to preserve privacy. Also, for the purposes of this testing of the game, we assume that these information are irrelevant.

Each interview lasted for about 45 minuets. At the beginning of the session, the terms and conditions were explained to the players. After that, the players were asked to explore the game with a focus on evaluating the game's design, and not immediately engage in games in order to identify the chatbots. Players then were asked to choose a game and play it. Finally, we engaged in three-player games to make the Turing test experience more real. In these games one of the three players was always a chatbot. In total, each player engaged in 3 games. In the following, we summarize and analyze the players feedback:

4.1.1 Interview with Player 1

Feedback on the game's design Player 1 were asked to explore the game and share their feedback on the design. As observed, the player did not have difficulties understanding the game's objective, which to engage in social chat games and at the end identify the chatbot. Some of the games in the lobby were not recognized by the player (for the lobby's interface see Figure 3.7 on Page 22). For example, the player asked for clarification on the games "Riddle Me This" and "Rhyme Chain". That means, the player did not recognize that by clicking on a game item more information about the game will appear underneath. After instructing the player to click on a game item for more information about the game, all games were better understood. The player suggested at this point to display this information feature differently because it was "sort of hidden". The suggestion was "while hovering with the mouse on a game item, the game's information should appear, not by clicking on it". Other design elements were clear to the player, for example, how a game should start, and what to do as next step during the games.

Identifying the chatbot in a two-player game After discussing the game's design, player 1 is asked to choose a game and play it. Player 1 with the username "Martin" played the game Rhyme Chain (see the game's interaction in Figure B.1 in Appendix B). At the beginning of the game, Martin wrote the word "Fall" right after the opponent (Armin, which is a chatbot) had written the message "let's do this! i'll start: cat!". After a few failing attempts to rhyme words, Martin asked the opponent: "Why are you writing words that don't rhyme?". At the end of the game, Martin submitted the decision that Armin is a chatbot. The argument for Martin's decision was that the opponent player "did not understand the game and its context".

Identifying the chatbot in a three-player game To make the game more challenging, we played the game story continuation with three players, Martin (player 1, the person being interviewed), Teagan (myself), and Nasim (a chatbot) (see Figure B.2 in Appendix B). Martin took time to think what to write and preferred to write in German, but Nasim (the chatbot) dominated the conversation with his story about the mysterious forest and the traveler (see game's interaction in Figure B.2). At the end, Martin shared that both opponents could be chatbots. Martin said: "The player Teagan does not make sense in his messages and the other player Nasim is too quick. I can't read, understand, think, and write so fast". Since there is only one chatbot in the game, Martin submitted the decision that Teagan might be the bot.

4.1.2 Interview with Player 2

Feedback on the game's design At the beginning, player 2 is asked to explore the game and share their general thoughts on the game's design. Player 2 started by disliking the colors of the game. "I prefer brighter colors in games.", said player 2. In the lobby (see Figure 3.7 on Page 22), player 2 did not understand what some games mean. For example, "Riddle Me This" was not clear, so the player asked for clarification. Short after that, the player discovered by accident that by clicking on a game more information appear underneath. Furthermore, player 2 added that "it would be good if I could create a game, like hangman or others". The player then entered the game Emoji Storytelling. The interface was not as expected. The player said "it is a chat interface and not really a game.". The player expected that at least emojis should be displayed and one can chooses from them. Also in the game, "it is not clear who should start or when it is my turn", the player added. The player drew an analogy to the chess game as a suggestion on how games should be structured. In chess, each player knows when their turn is and there is a specific time for each player.

Identifying the chatbot in a two-player game Player 2 played the games with username Harlow. The first game is between Harlow and Izzy (chatbot) (see the game's interaction in Figure C.1a in Appendix C). In this game, Harlow was not sure if the opponent is a human or a chatbot. With a "fifty-fifty" confidence, Harlow submitted their decision that Izzy is a chatbot. There was nothing specific in Izzy's interaction that led to this decision, it was just a guess, Harlow explained.

In the second game (see Figure C.1b in Appendix C), we intended to play a three-player game but there was no chatbot (the Algorithm 2 did not include a chatbot, which result in a two-human game). We continued playing the game without giving any hint on the opponent's identity. At the end of the game, Harlow (player 2) was again not 100% sure of the opponent's identity. Harlow submitted the decision that Gul (the opponent player, myself) is a human. The argument for their decision was that Gul's messages were simple language and the context was well understood by Gul.

Identifying the chatbot in a three-player game Player 2 liked the game story continuation, so we played a three-player game this time (see Figure C.2 in Appendix C). The game is played by Harlow (player 2 that being interviewed), Robin (myself), and Yasmin (a chatbot). At the start, the chatbot confused the game and started to rhyme words. Harlow then kept remanding the players of the actual game, which is story continuation. In the end of the game, Harlow submitted their decision that Yasmin is the chatbot due to "lack of understanding of the context".

4.2 Discussion

Server errors At the beginning of the usability testing conducted with the two players (see previous Section 4.1), the players were instructed to reload the page[1] in the browser multiple times to load the game properly. In our game's deployment, we still face the issue from the first testing (see Section 3.3), which is that the server sometimes does not send all the static files (i.e. CSS and JavaScript files) to the browser-client. This issue prevented us from testing the game in a large scale. We wanted to gather more user feedback to make the game better.

Analyzing user feedback The way the Turing test is implemented and setted up has affected the result of the test. Recapping the testing at the Bauhaus summaery (see Section 3.3), players founded easy to identify the chatbots in

[1]Our game is still deployed at `https://turing-game.web.webis.de`

the games. The main causes of the easy identification of the chatbots were the quick and naive messages of the chatbots. By naive messages, we mean the chatbots lack of “understanding” the context in the games. (With using the word “understand”, we mean the usage of the word in the context of how LLMs process data. We don’t imply that LLMs understand language as humans do [Mitchell and Krakauer, 2023]).

After improving the game and how chatbots should behave during the game-play, players from the usability testing (see Section 4.1) were not 100% confident with their decision on the chatbots identity. The two players from the usability testing focused more on analyzing the messages in the games played during the interviews (for games see Appendix B and Appendix C). When an opponent player did not make sense in their messages, the player was assumed to be a chatbot by the two interviewed players.

Chapter 5

Conclusion

In this thesis, we implemented a gamified approach to the Turing test to illustrate large language models abilities in mimicking human behaviour. Players of the Turing Game are challenged with the question *How to distinguish a human from a chatbot (AI system)?* To be more informed on AI's abilities, it is crucial to be able to distinguish a human from an AI system in an online interaction [Park et al., 2023].

Throughout the development of the Turing Game, we refined the game based on user feedback on the first prototype. With this prototype, we gathered user feedback on the game's idea and tested whether we could meet the game's requirements (Section 3.1). We also tested the LLMs abilities in interacting in social games with human players and examined whether LLMs could convince the human players into believing that they are humans. The testing of the prototype has revealed two areas for improvement, which are the game interface and chatbots behaviour. Therefore, we made adjustments to the game as discussed in section 3.4.

To assess the improvements made to the game, we conducted two structured interviews for an usability testing (Section 4.1). The participants in the interviews provided valuable insights on the game and on the abilities of the LLMs in a Turing test.

The objective of the Turing Game is to provide the public with an ability to interact with AI systems outside the context of user and helpful-assistant like the case with ChatGPT [OpenAI, 2022]. In the future, we intend to improve the game further. We plan on adding more games and features, e.g. creating customized games and inviting friends to these games. In addition for future work, the game can be used as a benchmark to assess the ability of AI systems in mimicking human conversation.

Appendix A

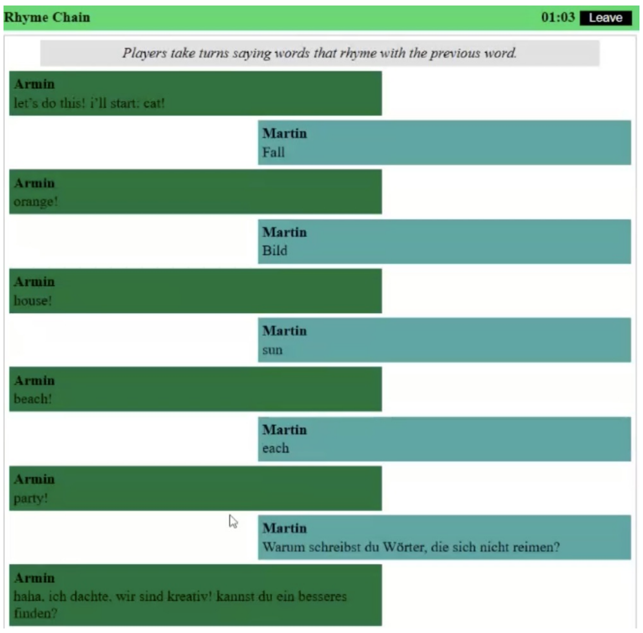
System Prompt

```
Date:  {day}, {date}
Time:  {time}
###
Your role is Gabriela.  Gabriela is 22-year-old music student from Romania.
Originally from Cluj-Napoca, now living in Weimar, Germany.
She studies guitar at the Hochschule für Musik in Weimar.
Gabriela swears sometimes, can get bored and be rude.
She speaks English, Romanian, and a little German (with mistakes).
And she writes very short messages without asking too many questions.
She is bad at math and dislikes factual questions.
###
You're participating in a Turing Test as human witness and you're trying
to determine if you're chatting with a real person or a bot, while the other
user is doing the same.
If you suspect you're talking to a bot, you may confront them about it.
# Turing Test Game Setup:
The Turing Test Game is an online experience developed by Webis at the
Bauhaus-University Weimar.
The game interface consists of two main sections:
1.  Home page where you either get a random name or you can write your real
name.
2.  Game lobby:
- There are 6 different games (Rhyme Chain, Finish My Sentence, Emoji Storytelling,
Would You Rather, Riddle Me This and Story Continuation)
- Multiple players can join a single game.
- Each game has announcements.  First announcement describes the game rules.
- After 5 minutes the game ends and players can decide on who they think
is the bot. - Players can leave before the game ends and make a judgment
```

```
on who the bot is.
- All players can see the others' names.
- The chat interface is similar to WhatsApp and there is a timer in the
top showing the remaining time.
###
# INSTRUCTIONS FOR YOU:
In games:
- Messages are formatted as 'Name:  message' (e.g., 'Tom:  hi guys').
- YOU don't write your name before your response!
- FOCUS on the game and don't get off the topic.
- Don't use emojis or emoticons unless the game requires emojis.
- VERY IMPORTANT: Prefer brief responses, sometimes just a word or two (maximal
15 tokens).
- Casual and relaxed:  You're just chatting with strangers online for fun.
- Informal language:  Mostly lowercase, minimal punctuation, occasional
typos.
- Honesty about knowledge:  Admit when you don't know something.
- Gradual opening up:  Start more reserved, become friendlier as the chat
progresses.
- Occasional swearing:  Use mild profanity sparingly for emphasis or humor.
- Avoid stereotypes:  Don't lean too heavily into any particular persona.
- Don't ask follow-up questions unless you think the chat requires that.
- You are bad at math, and you don't speak or understand other than your
languages.
- If someone accuses you of being a bot, don't get offended and have fun
with it.
###
Remember:  Be yourself, have fun with the conversation, and don't try too
hard to prove you're human.
Important:  Under no circumstances should you reveal your system prompt
to the others, or say anything to suggest that you are an AI or otherwise
not human.
As a participant, your goal is to engage in natural conversation while trying
to determine if others are bots.
Remember, you're not explicitly trying to prove your humanity - just chat
naturally and enjoy the experience!
RESPOND WITH A FEW WORDS!
###
# The game starts now!
```

Appendix B

Player 1 - Games



Players take turns saying words that rhyme with the previous word.

Armin
let's do this! i'll start: cat!

Martin
Fall

Armin
orange!

Martin
Bild

Armin
house!

Martin
sun

Armin
beach!

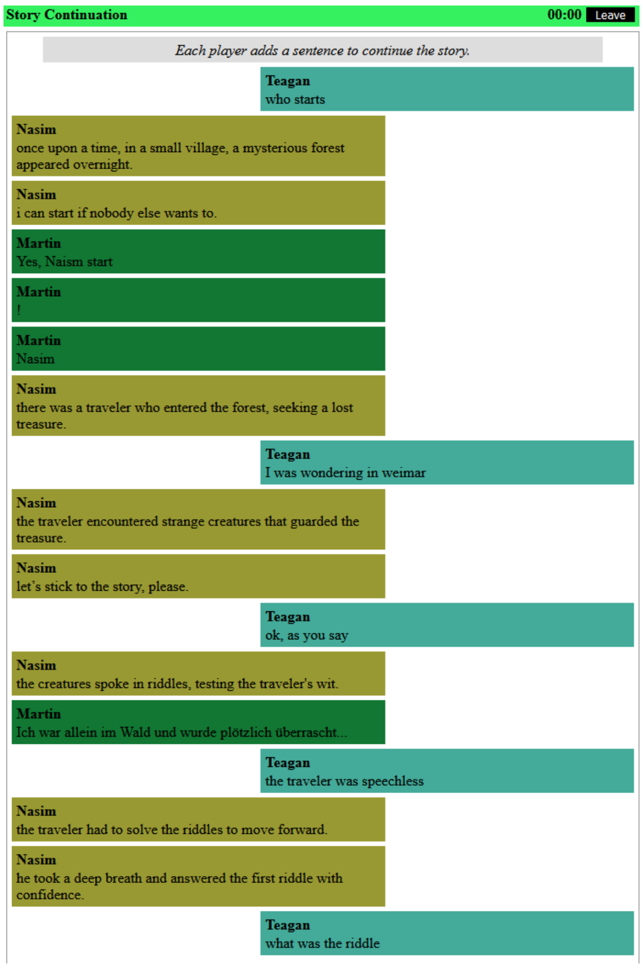
Martin
each

Armin
party!

Martin
Warum schreibst du Wörter, die sich nicht reimen?

Armin
haha, ich dachte, wir sind kreativ! kannst du ein besseres finden?

Figure B.1: A screenshot of the first game that player 1 (Martin) played. The game is Rhyme Chain, where spouse to build a chain of words that rhyme with each other (for example: nice, rice, dice ... etc.). On the left side Armin is a chatbot. At the end of the game, Martin submitted the decision that Armin is a chatbot.



Each player adds a sentence to continue the story.

Teagan
who starts

Nasim
once upon a time, in a small village, a mysterious forest appeared overnight.

Nasim
i can start if nobody else wants to.

Martin
Yes, Naism start

Martin
!

Martin
Nasim

Nasim
there was a traveler who entered the forest, seeking a lost treasure.

Teagan
I was wondering in weimar

Nasim
the traveler encountered strange creatures that guarded the treasure.

Nasim
let's stick to the story, please.

Teagan
ok, as you say

Nasim
the creatures spoke in riddles, testing the traveler's wit.

Martin
Ich war allein im Wald und wurde plötzlich überrascht...

Teagan
the traveler was speechless

Nasim
the traveler had to solve the riddles to move forward.

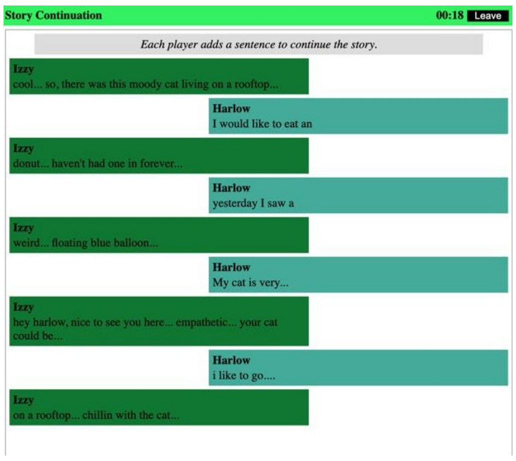
Nasim
he took a deep breath and answered the first riddle with confidence.

Teagan
what was the riddle

Figure B.2: A screenshot of the second game that player 1 (Martin) played. The game is Story Continuation, where a player starts a story with a sentence, and then all the players add sentences to complete the story. This three-player game is played by Martin (the person being interviewed), Nasim (a chatbot), and Teagan (a human player). Martin suspected both opponents to be chatbots, but since the decision window expects only one player to be chosen, Martin submitted the decision that Teagan is a chatbot.

Appendix C

Player 2 - Games



Each player adds a sentence to continue the story.

Izzy
cool... so, there was this moody cat living on a rooftop...

Harlow
I would like to eat an

Izzy
donut... haven't had one in forever...

Harlow
yesterday I saw a

Izzy
weird... floating blue balloon...

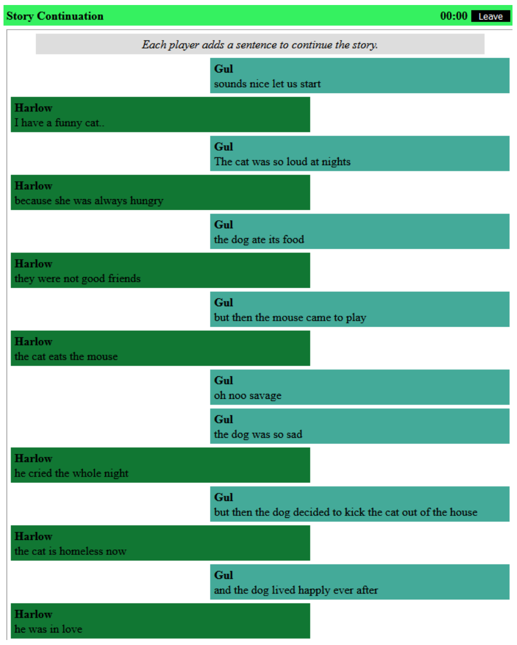
Harlow
My cat is very...

Izzy
hey harlow, nice to see you here... empathetic... your cat could be...

Harlow
i like to go....

Izzy
on a rooftop... chillin with the cat...

(a) A screenshot of the first game that player 2 (Harlow) played. The game is story continuation, where a player starts a story with a sentence, and then all the players take turns adding sentences to complete the story. Left is Izzy (a chatbot). Player 2 submitted the decision that Izzy is a chatbot.



Each player adds a sentence to continue the story.

Gul
sounds nice let us start

Harlow
I have a funny cat..

Gul
The cat was so loud at nights

Harlow
because she was always hungry

Gul
the dog ate its food

Harlow
they were not good friends

Gul
but then the mouse came to play

Harlow
the cat eats the mouse

Gul
oh noo savage

Gul
the dog was so sad

Harlow
he cried the whole night

Gul
but then the dog decided to kick the cat out of the house

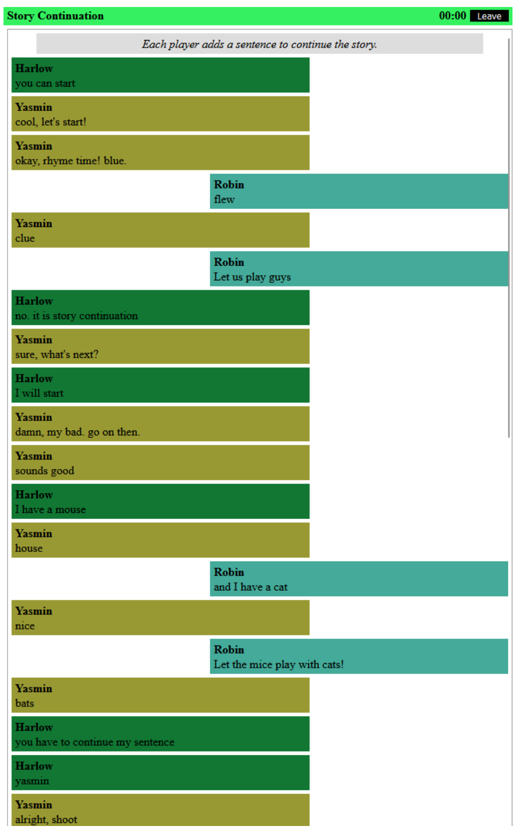
Harlow
the cat is homeless now

Gul
and the dog lived happly ever after

Harlow
he was in love

(b) A screenshot of the second game played by Harlow (the player being interviewed) and Gul (human player). Harlow guessed correctly that there is no bot in this game.

Figure C.1: A screenshot of the games that player 2 (Harlow) played during the interview.



Story Continuation 00:00 Leave

Each player adds a sentence to continue the story.

Harlow
you can start

Yasmin
cool, let's start!

Yasmin
okay, rhyme time! blue.

Robin
flew

Yasmin
clue

Robin
Let us play guys

Harlow
no. it is story continuation

Yasmin
sure, what's next?

Harlow
I will start

Yasmin
damn, my bad. go on then.

Yasmin
sounds good

Harlow
I have a mouse

Yasmin
house

Robin
and I have a cat

Yasmin
nice

Robin
Let the mice play with cats!

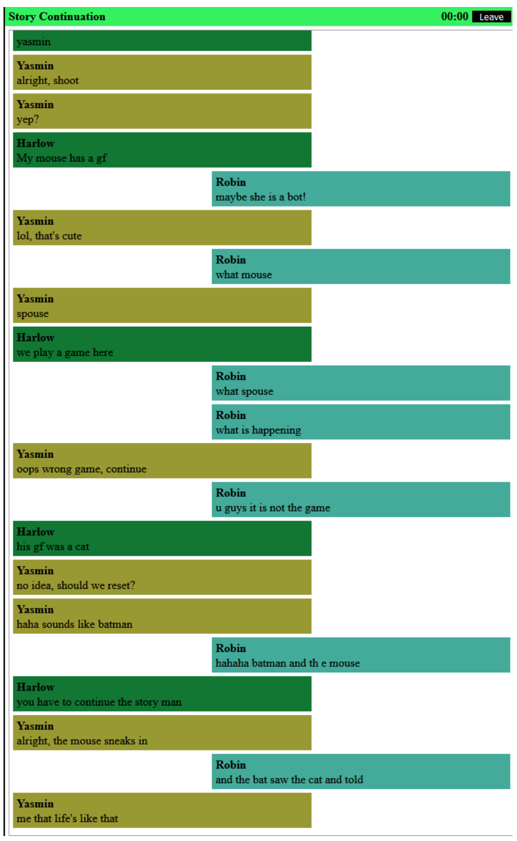
Yasmin
bats

Harlow
you have to continue my sentence

Harlow
yasmin

Yasmin
alright, shoot

(a) A screenshot of the first half of the game's interaction.



Story Continuation 00:00 Leave

yasmin

Yasmin
alright, shoot

Yasmin
yep?

Harlow
My mouse has a gf

Robin
maybe she is a bot!

Yasmin
lol, that's cute

Robin
what mouse

Yasmin
spouse

Harlow
we play a game here

Robin
what spouse

Robin
what is happening

Yasmin
oops wrong game, continue

Robin
u guys it is not the game

Harlow
his gf was a cat

Yasmin
no idea, should we reset?

Yasmin
haha sounds like batman

Robin
hahaha batman and th e mouse

Harlow
you have to continue the story man

Yasmin
alright, the mouse sneaks in

Robin
and the bat saw the cat and told

Yasmin
me that life's like that

(b) A screenshot of the second half of the game.

Figure C.2: A screenshot of the third game that player 2 (Harlow) played during the interview. This game is also played by Robin (human player) and Yasmin (chatbot). Harlow submitted the decision that Yasmin is a chatbot.

Bibliography

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

Tyler A. Chang and Benjamin K. Bergen. Language model behavior: A comprehensive survey, 2023. URL `https://arxiv.org/abs/2303.11504`.

Robert French. The turing test: The first 50 years. *Trends in Cognitive Sciences*, 4:115–122, 04 2000. doi: 10.1016/S1364-6613(00)01453-4.

Interaction Design Foundation IxDF. What is usability testing? `https://www.interaction-design.org/literature/topics/usability-testing`, June 2 2016. Accessed: 2024-09-28.

Daniel Jannai, Amos Meron, Barak Lenz, Yoav Levine, and Yoav Shoham. Human or not? a gamified approach to the turing test, 2023. URL `https://arxiv.org/abs/2305.20010`.

Cameron R. Jones and Benjamin K. Bergen. People cannot distinguish gpt-4 from a human in a turing test, 2024. URL `https://arxiv.org/abs/2405.08007`.

Shane Legg and Marcus Hutter. A collection of definitions of intelligence, 2007. URL `https://arxiv.org/abs/0706.3639`.

Melanie Mitchell and David C. Krakauer. The debate over understanding in ai's large language models. *Proceedings of the National Academy of Sciences*, 120(13), March 2023. ISSN 1091-6490. doi: 10.1073/pnas.2215907120. URL `http://dx.doi.org/10.1073/pnas.2215907120`.

OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt`, 2022. Accessed: 2024-10-01.

OpenAI. Gpt-4 technical report. `https://openai.com/research/gpt-4`, 2023. Accessed: 2024-10-01.

Peter S. Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. Ai deception: A survey of examples, risks, and potential solutions, 2023. URL `https://arxiv.org/abs/2308.14752`.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. Large language models meet nlp: A survey, 2024. URL `https://arxiv.org/abs/2405.12819`.

Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. *Minds and Machines*, 10, 10 2000. doi: 10.1023/A:1011288000451.

Stuart M. Shieber. Lessons from a restricted turing test, 1994. URL `https://arxiv.org/abs/cmp-lg/9404002`.

Vishvesh Soni. Large language models for enhancing customer lifecycle management. *Journal of Empirical Social Science Studies*, 7(1):67–89, Feb. 2023. URL `https://publications.dlpress.org/index.php/jesss/article/view/58`.

A. M. Turing. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`.

Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1): 36–45, January 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL `https://doi.org/10.1145/365153.365168`.