CMPE 492

LLM-driven Turing Game Bot

Student 1 Asude Ebrar Kızıloğlu

Advisors:

Suzan Üsküdarlı

Onur Güngör

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Broad Impact

The recent advancements in large language models (LLMs) have pushed the boundaries of AI. However, it has remained a challenge to truly measure the extent to which these models could imitate *human-like behaviors*. My project focuses on developing an LLM-driven chatbot that could successfully participate in a Turing game against human players, with important implications for this measurement challenge.

My study provides valuable insight into the capabilities and limitations of LLMs by testing the bot's ability to convince human participants of its humanity. I designed the research to inform the development of more robust and human-centered AI systems. The Turing game experiment represented a crucial step in our ongoing journey to understand and advance AI responsibly.

1.2. Ethical Considerations

Throughout my evaluation of the chatbot's conversational abilities, I maintained strict priorities around participant well-being. While I designed the bot to avoid offensive language, I recognized the risk of disturbing messages from other players during the games. To address this concern, I implemented a "report" button in the chatroom. During the experimental sessions, I briefed all participants on the expected code of conduct and explained the report button's function for flagging concerning behavior. This protocol specified that any violators would be immediately removed from the games. I am pleased to report that we encountered no violations during the user tests.

By placing participant safety and inclusivity at the forefront of my research design, I was able to conduct this research responsibly while gathering insights for thoughtful, trustworthy AI development.

2. PROJECT DEFINITION AND PLANNING

2.1. Project Definition

The term “Turing Test” was originally formed by the famous mathematician, and the father of theoretical computer science, Alan Turing, in 1950 [1]. The test aims to prove whether we can say a machine can think *logically*.

Building on the foundation of the ”Turing Test”, I developed an LLM-driven chatbot that could interact with humans through written communication. To evaluate the bot’s human-like behaviors, I created an online chatting environment that mirrored the original Turing Test structure. In this environment, I connected human users to chat rooms containing *two humans* and *one bot player*, challenging them to identify which participant was the bot. I designed the bot with the objective of convincing human players of its humanity.

In my initial approach, I explored multiple LLMs and prepared various prompts to test the bot’s capabilities. Through systematic testing, I analyzed the bot’s performance with different combinations of LLMs and prompts. I then organized in-person user tests where I gathered volunteers to participate in the Turing Game. By conducting multiple game sessions in this experiment setting, I collected substantial chat data that allowed me to analyze the performance of different bot versions, particularly focusing on their success rate in deceiving human players.

2.2. Project Planning

2.2.1. Project Time and Resource Estimation

This project was planned to be completed in one semester, as the CmpE492 coursework. I planed my time period as the following:

Table 2.1. Project Roadmap.

	Objectives
Weeks 0-2	Literature research & brainstorming.
Weeks 3-4	Initiate the LLM-driven bot implementation.
Weeks 5-7	Initial build of the online chat environment.
Weeks 7	Submit the midterm report.
Weeks 7-9	Deploy the chat server. Also, build the chatbots in parallel.
Weeks 9-10	Build the chatbot detection bot.
Weeks 11-12	Finalize the chat server and the bot's prompt, and the LLM to utilize.
Weeks 12-13	Conduct human experiments.
Weeks 14	Analyze the experiment data. Write the report, and prepare the poster.

In terms of the human resources used in this project, I, as the main contributor, dedicated approximately 8 - 16 hours of my time weekly. Also, I met with both of my advisors weekly on Wednesdays. During these meetings, I was guided by my advisors to assess the progress made, explore future opportunities for the project, and plan the rest of the weeks I have to complete this project successfully.

In terms of computational resources, I ran quantized versions of various large language models with 7B - 8B parameters. Such execution required around 16GB RAM. To deploy the online chat environment I built, I required a hosting service. For such a task, I made use of the Minerva machine from our department. The LLMs I utilized during the development process were all open-source resources. During the user tests, I utilized OpenAI's GPT 4o model [2].

2.2.2. Success Criteria

The success of this project was evaluated based on several key criteria:

(i) **System Implementation**

- Successfully develop and deploy a functioning chat environment
- Implement a stable LLM-driven chatbot capable of real-time responses
- Create a reliable data collection and analysis pipeline

(ii) **Experimental Goals**

- Conduct at least 80 valid game sessions
- Achieve a minimum bot deception rate of 20% (where at least one human fails to identify the bot)
- Maintain system stability throughout experimental sessions

(iii) **Analysis Objectives**

- Generate comprehensive metrics on LLMs' human-like chat behaviors
- Identify specific patterns that distinguish bot from human communication
- Produce actionable insights for improving bot performance

These criteria were designed to ensure both the technical success of the implementation and the research value of the experimental results.

2.2.3. Risk Analysis

During the project planning and implementation phases, I identified several potential risks and developed mitigation strategies:

(i) **Technical Risks**

- *API Reliability*: The system's dependence on external LLM APIs could lead to service interruptions. This was mitigated by implementing fallback options and timeout handling.
- *System Performance*: High concurrent user loads could impact stability. I addressed this through load testing and optimization of server resources.

(ii) **Experimental Risks**

- *Participant Coordination*: Scheduling multiple simultaneous participants could be challenging. I maintained a larger pool of volunteers than needed

and implemented flexible scheduling.

- *Data Quality*: Technical issues could compromise experimental results. I established clear validation criteria and excluded corrupted game sessions (e.g. where the bot was not responsive) from analysis.

(iii) **Privacy and Safety Risks**

- *Data Privacy*: Handling participant data required careful consideration. I implemented anonymization protocols and secure data handling procedures.
- *Content Moderation*: Risk of inappropriate messages was addressed through content filtering and a report button feature.

3. RELATED WORK

The idea of testing a machine's intelligence in the context of comparison with humans starts with the original Turing Test. Alan Turing initially proposed the idea of the Turing Test in his article, in 1950 [1]. In his work, Turing designed a thought experiment known as the "imitation game" to determine whether *machines can think*. The core idea was to set up a test where an interrogator would communicate via text with a hidden entity and try to determine if it was a human or a machine. If the interrogator could not distinguish the machine from a human, Turing argued that the machine should be considered to possess *human-level intelligence*. This innovative *"Turing test"*, laid the foundation for assessing *artificial intelligence* based on its ability to convincingly mimic human-like behavior. Turing's visionary work continues to inspire researchers, including us, to discover the boundaries of machine intelligence.

Philip Hingston, a Computer Science professor at Edith Cowan University produced a version of the Turing Test, focused on the computer game bots in 2009 [3] and 2010 [4]. For this study, the author organized a competition, where programmers submitted their robots to pass this version of the Turing test. In each round of the competition, a judge was matched against a human player and a bot, and all three players computed in a 10-minute shooting game (like laserball). The objective of the judge was to determine which of the other players was the bot. None of the bots succeeded in fooling the judges (4 out of 5) in the 2008 and 2009 versions of the competition. According to the judges, some features of bots that make them easy to identify are the following:

- Failure to act consistently - "forgets" about opponents
- Getting "stuck" & stubbornness & lack of awareness
- Very accurate shooting

Common features of humans mentioned in this article included aggressiveness and reacting to situations.

4. METHODOLOGY

In my research, I implemented a LLM-driven chatbot and integrated them into a custom-built chat environment designed for synchronous human-bot interactions. The environment facilitated controlled experiments in which multiple users could engage in real-time conversation.

Each chat room (game session) consisted of three participants: two human players and one chatbot. Human participants accessed the system through their web browsers, where they encountered a streamlined interface featuring a message input field and two "accusation" buttons. These buttons enabled participants to designate another player as "being the bot" - a decision that, once made, was irreversible. Upon the first accusation in a session, the remaining human participant had 15 seconds to make their own accusation. The system then concluded the session by revealing the bot's identity and displaying relevant statistics.

I structured each session with a maximum duration of five minutes, after which both human participants were required to make their accusations within a 15-second window. To evaluate the bot's effectiveness in emulating human behavior, I recorded comprehensive session data, including complete chat histories and detailed accusation metrics (participant identifiers, timing, and targeting).

Following the implementation phase, I conducted four experimental sessions (user tests) with human participants to assess the chatbot's performance. Each session involved participants engaging in ten consecutive rounds of gameplay, allowing me to analyze various aspects of human-bot interaction. My analysis encompassed the bot's detection rates, comparative vocabulary usage between human participants and the bot, and patterns in user accusation behavior across multiple rounds. This systematic approach enabled me to gather rich empirical data on both the technical performance of the system and the evolving strategies of human participants throughout extended gameplay sessions.

Additionally, I developed a bot detection system utilizing LLM technology, specifically designed to distinguish between human and artificial conversational patterns. My research included the testing of both the detection system's accuracy and the chatbot's ability to convincingly simulate human interaction patterns.

5. REQUIREMENTS SPECIFICATION

5.1. Software Requirements

5.1.1. Glossary

- **Chatbot**: An LLM-driven model conditioned to talk like humans in a writing manner.
- **Bot Detection Bot**: A model used to determine whether the entity it communicates with is human or a bot.
- **Game Server**: The deployed server that hosts the chat-based Turing Game sessions.
- **Game Session**: An instance of the chat-based Turing Game involving two human players and one bot, where players try to identify the bot.
- **Chat History**: The stored conversation within a game session, including all messages and timestamps, is used for post-game analysis.
- **Accusation**: An action taken by human players to label another player as the bot during a game session.
- **A/B Testing**: A method of comparing two or more variations to determine which performs better under specific conditions.

5.1.2. Bot Requirements

(i) The chatbot shall be able to interpret and respond to human chat messages within each game session in a human-like manner.

(ii) The chatbot shall be able to respond within a specified time frame (e.g., within 5 seconds) to simulate natural conversation flow.

(iii) The chatbot shall be able to receive different prompts, which guide its responses to adapt to different game sessions.

(iv) The chatbot shall aim to convince human players of its human identity by carefully choosing its responses.

(v) The chatbot shall maintain an objective of not being identified as a bot by either of the human players for the duration of the game session.

(vi) The bot detection bot shall be able to analyze chat data to determine if an entity it is communicating with is a bot or a human.

(vii) The bot detection bot shall provide feedback on its detection accuracy after each game session to enable evaluation of its performance.

5.1.3. Server Requirements

(i) The server shall be able to create and manage multiple concurrent chat sessions, each involving two human players and one bot.

(ii) The server shall terminate a session after either an accusation is made by both human players or after the session's maximum time limit (e.g., 5 minutes).

(iii) The server shall store chat histories for each game session, including all messages, timestamps, and metadata (such as usernames).

(iv) The server shall log accusation events, including which human player accused whom, the timing of accusations, and end-game results.

(v) The server shall store data for analysis, ensuring the chat history and accusation logs are retrievable for post-game evaluation.

(vi) The server shall allow human players to make accusations by clicking an "accusation" button.

(vii) The server shall enforce the rule that if one human accuses another player, the remaining human must make an accusation within the next 15 seconds.

5.1.4. User Requirements

(i) Users shall be able to join a game session from a web browser.

(ii) Users shall see a simple chat interface that includes an input box for messages and buttons to accuse other players of being the bot.

(iii) Users shall be able to view game statistics, including which player was identified as the bot, after a game session ends.

(iv) Users shall be able to make an accusation against either of the other two partic-

ipants during a game session by selecting the appropriate button.

(v) Users shall not be able to retract an accusation once it is made.

(vi) Users shall be notified if the game session time limit (e.g., 5 minutes) is reached.

(vii) Users shall be notified if they need to make an accusation within 15 seconds after the first accusation has been made in the game session.

5.2. Use Case Diagrams

As shown in Figure 5.1, the system supports three main actor types: human players, administrators, and the chatbot.

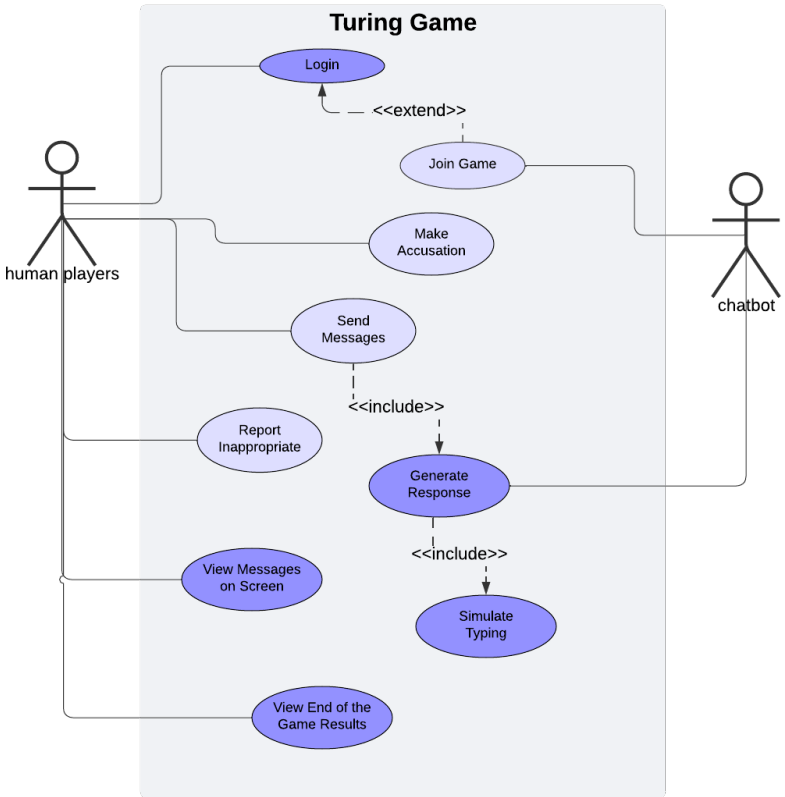

Figure 5.1. UML Use Case Diagram

6. DESIGN

6.1. Information Structure

See the Class diagram below for the information structure.

6.2. Information Flow

The state transitions of the chat server are illustrated in Figure 6.1.

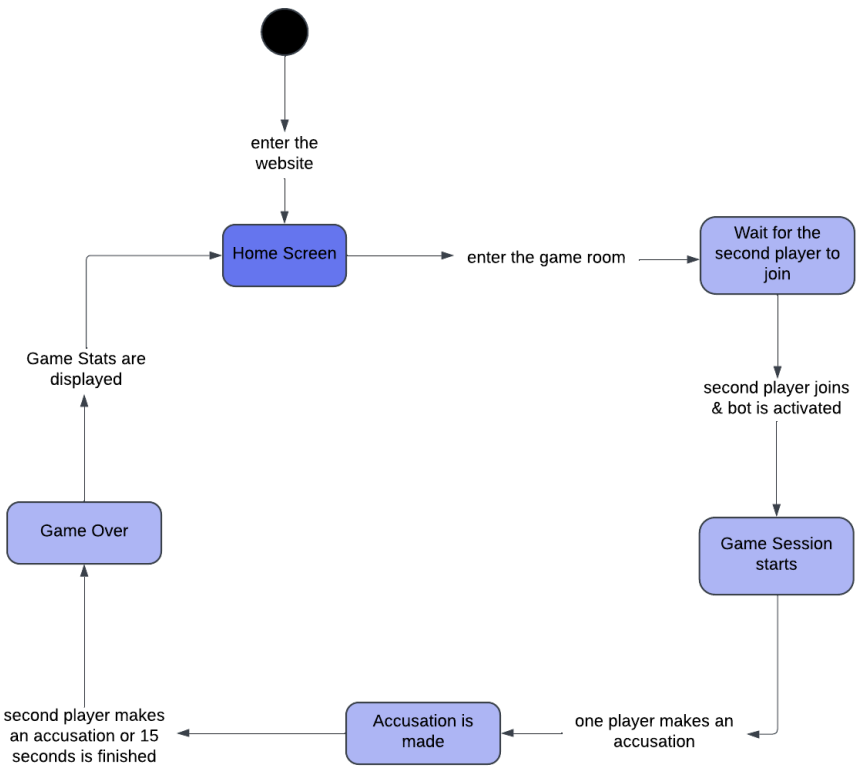

Figure 6.1. UML State Diagram

6.3. System Design

6.3.1. Class Diagram

The system's class structure, depicted in Figure 6.2, shows the relationships between key components.

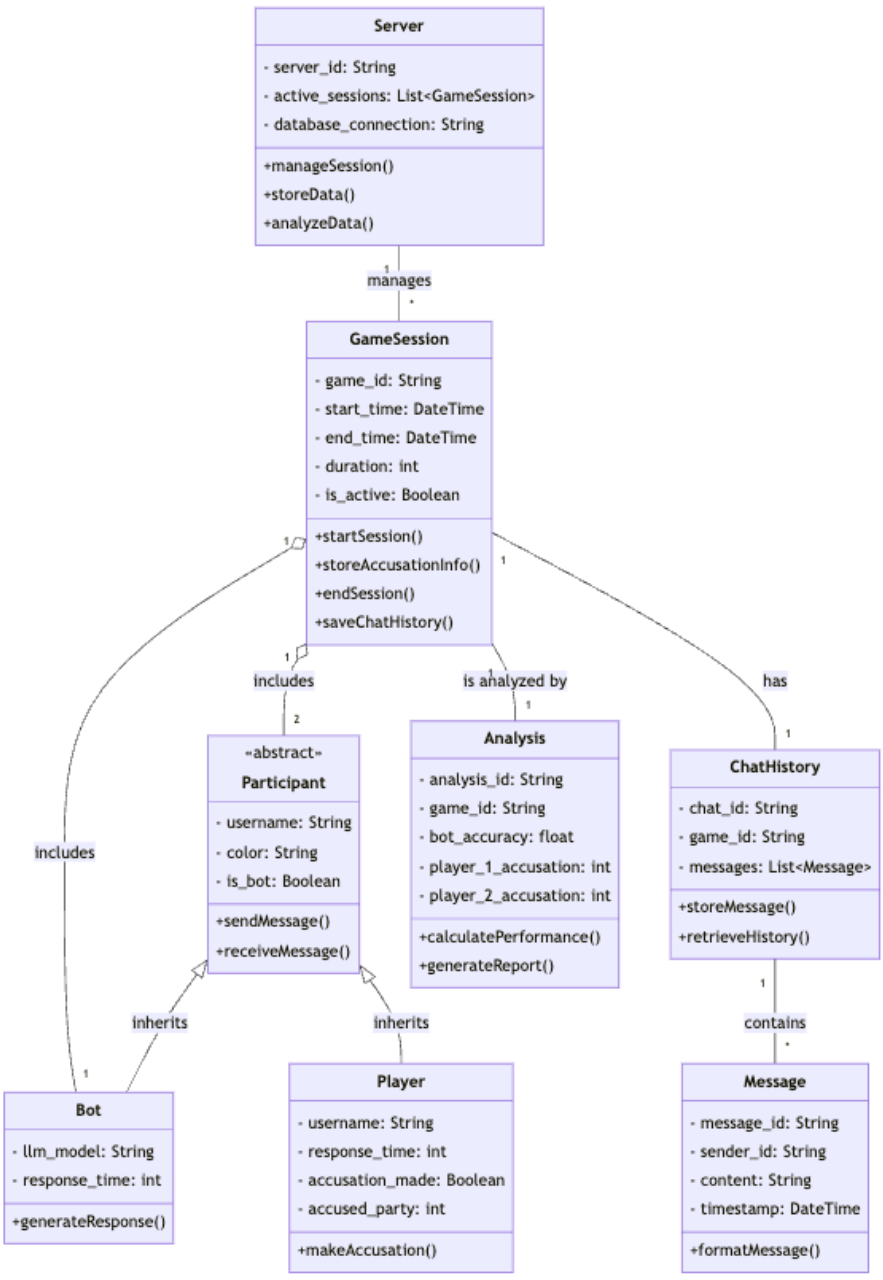

Figure 6.2. UML Class Diagram

6.4. User Interface Design

Figure 6.3 presents the user interface design for the chat room.

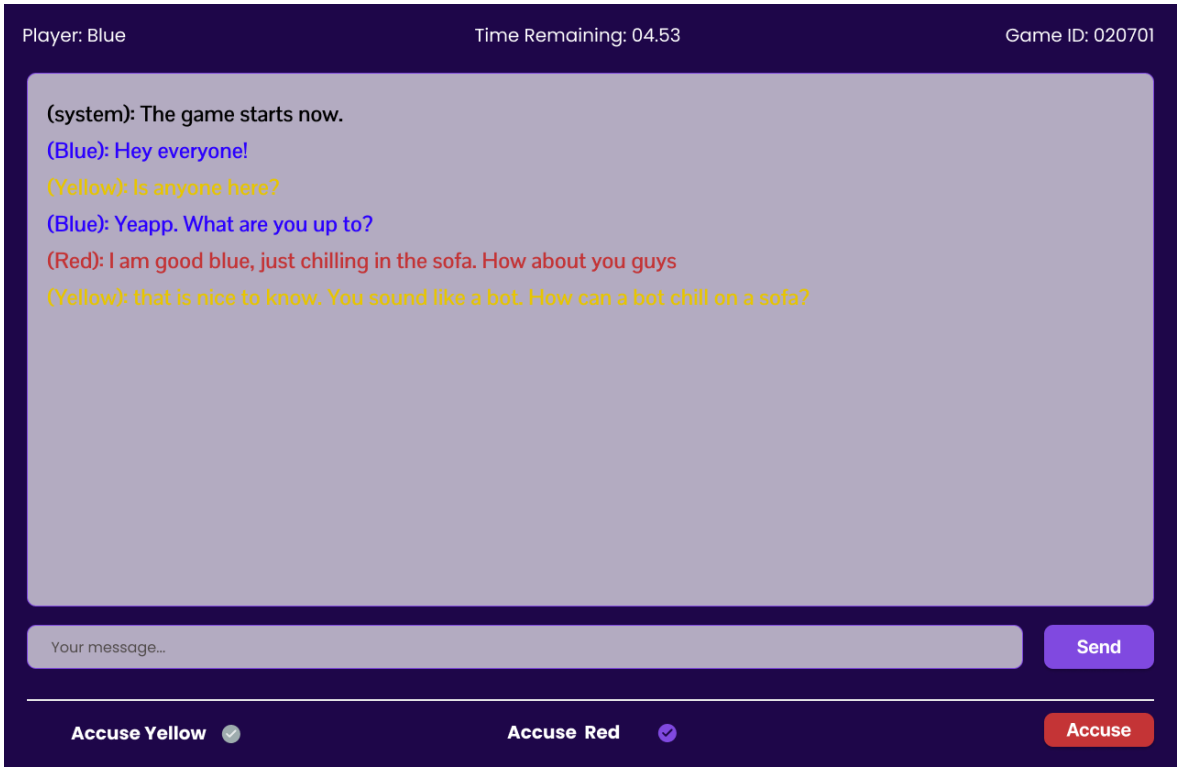

Figure 6.3. Chat Room Mockup

7. IMPLEMENTATION AND TESTING

7.1. Implementation

7.1.1. Chatbot Implementation

To develop the core component of our Turing game system, I focused on creating an LLM-driven chatbot capable of engaging in human-like conversations. This process involved several major key steps:

Choice of LLM

I evaluated various open-source LLMs, such as LLaMA, Mistral, and Gemma, to determine the most suitable models for our chatbot implementation. Initially, I utilized Ollama API [5] to connect to several LLMs. After the initial tries with Ollama, upon my advisors' recommendation, I switched to using Groq Python API library [6] for faster access to LLMs.

During the development phase of the chatbot, I mainly made us of Groq. Later, for the human experiments, we wanted to ensure the fastest and most reliable LLM infrastructure. Thus, I connected to the Open API and used the 4o [2] model during the user tests. Such model improved the chatbot's human-like behaviors in chatting. All of the selections above was guided by factors like conversational ability, ability to follow the given prompt, and safety.

Chatbot Prompt

To enhance the bot's human-like responses, I experimented with different prompts. I started by incorporating the bot requirements I determined (and mentioned above) into the chatbot prompt [Appendix A.1.].

Moreover, I had brainstormed about the human-like chatting behaviors prior to the chatbot implementation, and I used some of those ideas to craft the prompt. I included some correspondence between my brainstorming ideas and the rules in the final prompt below:

- "Humans do not often give overly logical or precise answers" `--->` "*What to Avoid:*
  - *perfect grammar all the time*
  - *complicated words*
  - *long detailed answers*
  - *showing off knowledge*
- "Humans make conversational quirks" `--->` "*Sound Natural By:*
  - *making small mistakes sometimes*
  - *using basic english most of the time*
  - *saying "hmm" or "well" when thinking*
- "Humans can get confused sometimes" `--->` "*ask what words mean if they're complicated*"

Chatbot Persona

In designing the chatbot's personality, I carefully crafted a persona that would resonate with the participant demographic while maintaining natural conversational patterns. Given that all participants were English as Second Language (ESL) speakers, I configured the bot to communicate at an appropriate linguistic level, mirroring the typical language patterns of ESL university students.

The bot's persona was constructed as a 20-year-old student at Bogazici University (colloquially known as "boun"), embodying characteristics common among the study's target demographic. While maintaining anonymity regarding its specific identity, the bot was programmed to engage in casual, friendly conversations about topics relevant to university life, gaming, travel experiences, knitting, cooking, and puzzle-solving. I deliberately avoided discussions about potentially stressful topics like final

examinations to maintain a relaxed conversational atmosphere.

This carefully calibrated persona allowed the bot to participate in natural conversations about contemporary topics and shared interests, while maintaining linguistic patterns consistent with the ESL participant pool. The combination of relatable interests and appropriate language proficiency level was designed to facilitate authentic interactions within the experimental context.

It's important to note that while this specific persona was chosen to align with my participant pool's background and interests, the system's architecture allows for flexible persona modification. The bot's personality, interests, language proficiency, and conversational style can be readily adjusted through prompt engineering to suit different experimental contexts or practical applications. This adaptability makes the system valuable for various research scenarios or real-world deployments where different persona characteristics might be more appropriate or effective.

Implementation

The bot's conversational logic was implemented to handle various incoming messages, adhere to the required response time, occasionally make human-like mistakes such as grammatical errors, and maintain its objective of avoiding detection.

I implemented a `TuringBot` class in Python and incorporated various bot functionalities there. Major functions of this class were elaborated below:

- `start_game`: Initiate the chat history with the system prompt as the "*developer*" role.
- `calculate_typing_delay`: Calculate a typing delay for the bot depending on the length of the LLM's message. The base typing speed is set to 4 characters per second (240 chars per minute). Also, randomness is added to make it more natural.
- `on_message_openai`: Sends the API request to Open AI with the parameter

`messages` containing the system prompt with the `developer` role, humans' messages in the game with the `user` role and the bot's previous messages in the game with the `assistant` role. The API request is implemented as the following:

```
chat_completion = OpenAI(api_key=self.openai_api_key,).
    chat.completions.create(
        messages=message,
        model=self.model_name,
        temperature=0.7
    )
```

- `introduce_typo`: This is called from the `on_message_openai` function and it adds some typos to the bot's message with some percentage. The following types of typos were used with the given probabilities:
  - Swap adjacent chars with 0.15
  - Repeat letter with 0.12
  - Remove space with 0.10
  - Add space with 0.08
  - Remove letter with 0.08
  - Double punctuation with 0.07
  - Capitalize random with 0.06

Safety and Ethical behavior

I implemented comprehensive safeguards in the bot's prompt engineering to prevent the generation of offensive, biased, or otherwise undesirable content. These safeguards included explicit instructions to avoid controversial topics, maintain respectful discourse, and immediately disengage from potentially harmful conversations. I also incorporated specific guidelines for the bot to refrain from discussing sensitive personal information or engaging in discriminatory language. These measures were crucial in prioritizing the well-being and comfort of human participants throughout the experimental sessions.

Iterative Testing and Refinement

Throughout the development process, I conducted extensive testing cycles to assess and enhance the chatbot's performance. This iterative approach involved multiple rounds of conversation analysis, prompt refinement, and behavioral adjustments. I systematically evaluated the bot's responses across various conversational scenarios and made incremental improvements to its human-like characteristics. This included fine-tuning the bot's language patterns, response timing, and topic engagement strategies. As a result, I determined the best time interval to send the message request to the bot for it to write something in the game.

7.1.2. Chat Server Implementation

In parallel with the chatbot implementation, I built the online chat environment that hosted the Turing game sessions. This server component handled the following responsibilities:

Session Management

I designed a robust session management system that controlled access to game sessions. Rather than allowing users to freely choose game IDs, I implemented a structured interface where participants were presented with exactly ten game buttons, each pre-configured to direct them to their assigned game sessions. As participants completed each game, the corresponding button was automatically disabled and visually grayed out to prevent re-entry into completed sessions. This controlled approach ensured that participants followed the intended game sequence and prevented unauthorized access to other game sessions.

Player Connectivity

Users joined the game sessions through a web browser interface, with the server handling all client-server communication. I implemented authentication to ensure only authorized participants could access their designated game sessions.

Chat Functionality

The server routed messages between participants, ensuring smooth conversation flow and enforcing time constraints. I implemented real-time message delivery and synchronization to maintain natural conversation pacing.

Accusation and Report Handling

I developed two critical user interaction features: the accusation system and a report button for safety concerns. The accusation system allowed human players to make their bot identification choices, while the report button provided immediate means to flag inappropriate behavior. Both features triggered appropriate game logic and state transitions when activated.

Data Logging

The server maintained detailed logs of chat histories, accusation events, start and end times of the games and any reported incidents for each game session. This comprehensive logging system facilitated thorough post-game analysis and evaluation of the bot's performance.

Deployment and Scalability

I designed and deployed the chat environment server to handle the anticipated user load and game session volume, successfully supporting multiple concurrent game sessions during the experimental phase. The system maintained reliable performance throughout all test sessions.

Through the implementation of these components - the LLM-driven chatbot and the structured online chat server - I created a comprehensive system capable of hosting controlled Turing game sessions and evaluating the human-likeness of our AI assistant. I stored my implementations on a GitHub repository. The addition of features like

session-specific access control, automated game progression tracking, and safety reporting mechanisms ensured both experimental integrity and participant safety throughout the study.

7.2. Testing

My testing strategy consisted of two phases: developmental testing and formal user experiments. During the development phase, my advisors and I conducted preliminary tests to evaluate the bot's responsiveness, system synchronization during game sessions, and the functionality of key interface elements including accusation buttons, report features, and message input mechanisms.

Following successful developmental testing, I executed 4 formal experimental sessions with human participants. Each session involved 6 participants engaging in 10 rounds of gameplay. I designed the round structure to ensure each participant interacted with every other participant in two distinct game sessions. To maintain experimental integrity, I conducted these sessions in a specially arranged room (The A4 classroom of BM building) divided by a partial wall. Participants were strategically positioned on opposite sides of this divider to prevent any non-verbal cues from influencing their bot detection decisions during gameplay.

To further control for potential environmental variables, I implemented a rotation system where two pairs of participants exchanged positions after the third and sixth rounds. The experimental sessions were conducted on December 24th, 25th, 26th, and 31st, with each session following identical protocols. Additionally, all of the participants were informed about the overall structure of the experiments and confirmed their participation with the consent form that was required by the ethic committee of the engineering department.

At the conclusion of each session, I administered exit interviews to gather qualitative feedback from participants. This combination of quantitative gameplay data and qualitative participant feedback provided a comprehensive dataset for analysis, which

I examine in detail in the following section.

7.3. Deployment

The system's deployment architecture is detailed in Figure 7.1.

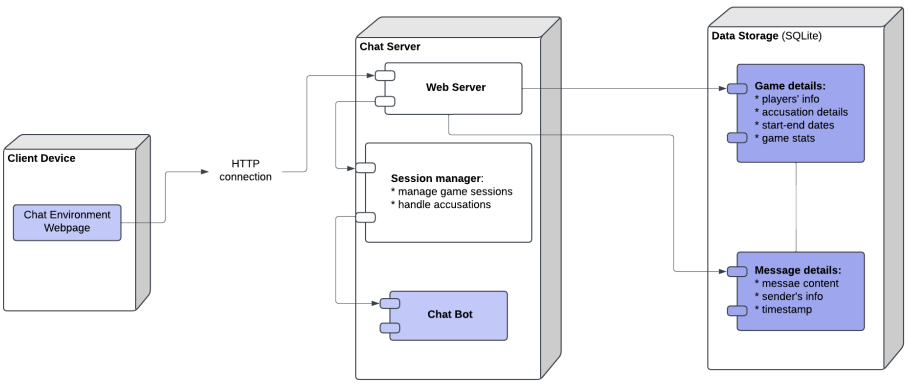

Figure 7.1. UML Deployment Diagram

8. RESULTS

8.1. Implementation Method

To facilitate the Turing game experiments, I developed a user-friendly web-based chat application. The implementation followed a multi-tier architecture: the front-end, built with JavaScript and HTML, provided an intuitive interface for users, while the chatbot's core logic was implemented in Python, taking advantage of its strong natural language processing capabilities.

For consistent communication between the web server and the bot, I used Python Flask to create a RESTful API. This setup enabled real-time message exchange and efficient management of the game state while maintaining a modular design. The Flask server handled all HTTP requests, coordinating the chat functionality and game logic, and ensured smooth data flow between the user interface and the bot's processing engine.

8.2. Experimental Evaluation

The experimental phase consisted of four structured workshop sessions designed to assess the chatbot's performance in realistic conditions. Each session was conducted in a controlled environment with the following parameters:

8.2.1. Experimental Scale

- 4 workshop sessions over a two-week period in December 2024
- 24 total participants (6 per session)
- Initial target of 120 game sessions, with 93 valid sessions after excluding technical issues

8.2.2. Performance Metrics

The evaluation framework encompassed four key dimensions:

Bot Detection Rate

- Success rate of bot identification
- Analysis of false positive accusations
- Round-by-round learning effect patterns

Conversation Analysis

- Session duration metrics
- Message frequency and volume
- Average message length and complexity patterns

Language Patterns

- Vocabulary diversity metrics
- Conversation topic maintenance
- Response contextual relevance

User Behavior

- Timing of accusations
- Post-experiment confidence assessments
- Strategy evolution across game rounds

This comprehensive evaluation framework provided quantitative and qualitative data for assessing both the technical performance of the system and its effectiveness in simulating human-like conversation.

8.3. Data Analysis

After completing the experiments, I conducted an in-depth analysis of the game sessions, focusing on various aspects of interaction and performance. Key areas of analysis included:

- Bot detection accuracy and participant behavior patterns
- Frequency and length of messages
- Linguistic complexity and vocabulary use
- Changes in user strategies across games
- Timing and confidence behind accusations

During the sessions, some technical issues arose due to timeout errors caused by internet connectivity and synchronization delays between the web server and OpenAI API. As a result, I excluded 27 game sessions where the bot couldn't maintain consistent performance. The analysis focused on the remaining 93 complete sessions, ensuring reliable metrics and a full interaction record.

This detailed analysis provided insights into both the system's technical performance and human behavior during the Turing test, offering a clearer picture of the bot's strengths and areas for improvement.

8.3.1. Bot's Performance

A primary metric for evaluating the chatbot's effectiveness was its ability to mimick human behavior during game sessions. I analyzed the accusation patterns across all 93 valid game sessions to understand the bot's success in emulating human conversation.

The data revealed interesting patterns in participant behavior and bot detection rates. In approximately 6.45% of games, both human players incorrectly identified each other as the bot, suggesting that the chatbot successfully emulated human-like

Table 8.1. Distribution of Accusation Patterns in Game Sessions

Accusation Pattern	**Game Count**	**Proportion (%)**
Both accused human	6	6.45
One player accused bot	8	8.60
Both accused bot	56	60.22
One accused human	3	3.23
One accused bot, one accused human	20	21.51
Total	93	100.00

behavior in these instances. Conversely, in 60.22% of games, both players correctly identified the bot, indicating that certain behavioral patterns consistently signaled the bot's artificial nature.

Of particular interest were the mixed-accusation scenarios, where one player accused the bot while the other accused a human participant (21.51% of games). This disagreement among human players suggests that the bot's behavior was sufficiently ambiguous to create uncertainty in some cases. The calculated expected value of successful deceptions (where the bot convinced at least one human participant of its humanity) was approximately **0.49 persons per game session**.

These findings highlight the bot's strengths in creating human-like interactions while also pointing to areas where improvements are needed to reduce detectability.

8.3.2. Accusation Flows

To understand how participants' bot detection strategies evolved over multiple rounds, I analyzed the accusation patterns across all ten rounds for each experimental day. The scoring system, which rewarded the first correct bot identification with higher points, introduced an interesting competitive dynamic where participants not only tried to identify the bot but also attempted to strategically mislead their opponents.

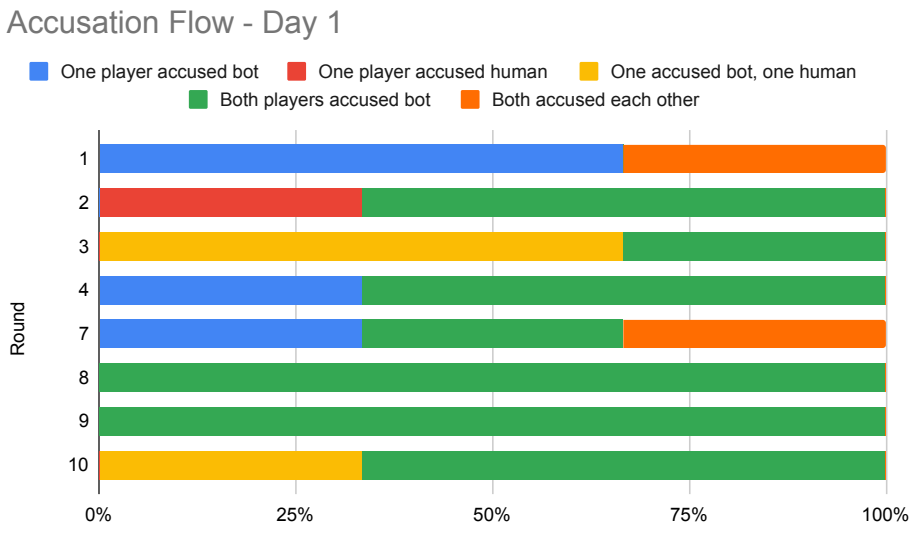

Figure 8.1. Accusation Flow - Day 1

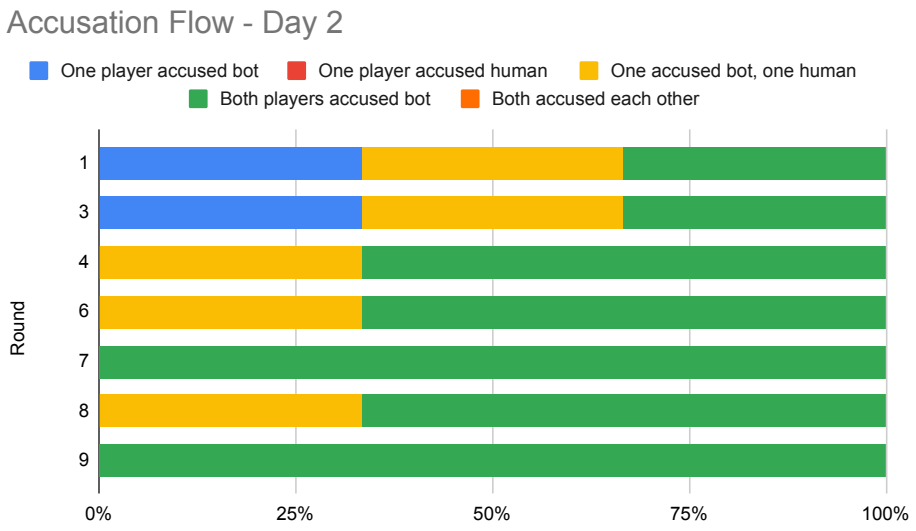

Figure 8.2. Accusation Flow - Day 2

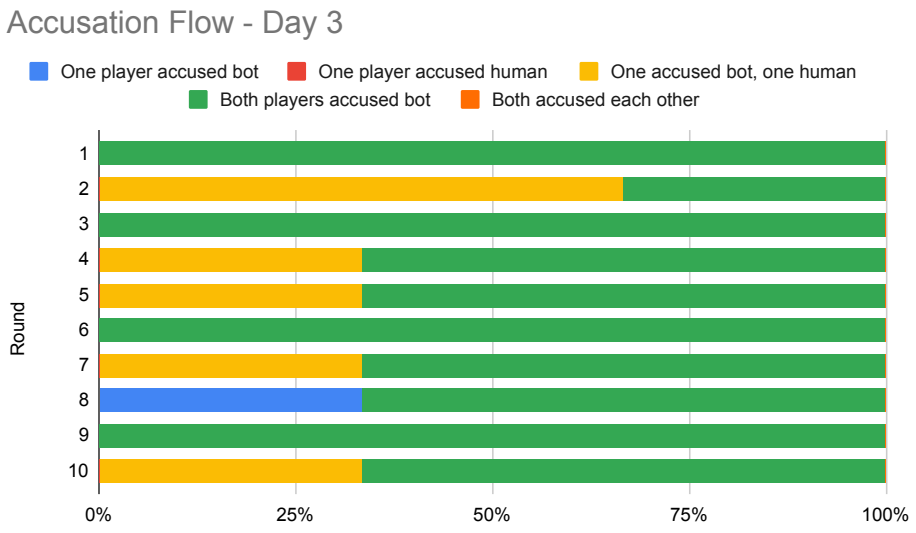

Figure 8.3. Accusation Flow - Day 3

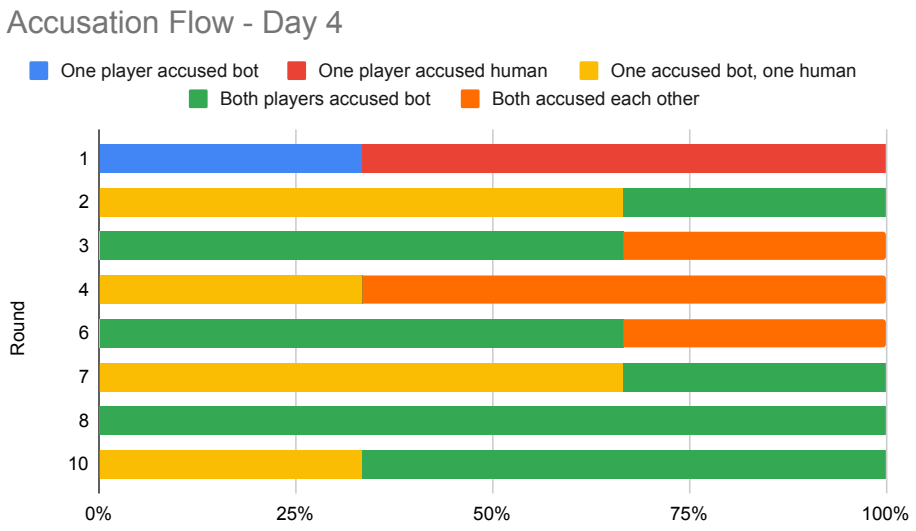

Figure 8.4. Accusation Flow - Day 4

Implementation Evolution and Its Impact:

A critical aspect of this analysis is the sequential improvements made to the system based on participant feedback and observed patterns:

1. *Typo Diversification (Days 3-4):* After feedback from the first two days indicated that the bot's consistent pattern of adjacent character swaps was a reliable detection signal, I expanded the variety of typo patterns. This modification appeared to increase the challenge of bot detection, as evidenced by the more diverse accusation patterns in Days 3 and 4.

2. *UI Randomization (Day 4):* An unintended implementation detail where the bot's accusation button consistently appeared on the right was addressed before Day 4, introducing randomized button placement. This change likely contributed to the notably different accusation patterns observed in Day 4's data.

Day-by-Day Analysis:

Days 1-2: Early experiments showed relatively consistent bot detection patterns, partly influenced by the predictable typo pattern and fixed UI element positioning. The high rate of correct identifications suggests participants quickly learned to recognize these consistent bot behaviors.

Days 3-4: The introduction of diverse typo patterns created more uncertainty, reflected in the increased frequency of split accusations and strategic gameplay. Day 4's particularly volatile patterns likely resulted from the combination of diverse typos and randomized accusation button placement, removing previously reliable detection signals.

Strategic Evolution Across Sessions:

1. *Adaptation to System Changes:* The progression across days shows how partic-

ipants had to continuously adapt their strategies as the system evolved, moving from reliance on consistent patterns to more nuanced detection methods.

2. *Meta-game Complexity:* The scoring system created a sophisticated meta-game where participants balanced multiple factors:

- Bot detection based on conversational patterns
- Strategic misdirection of opponents
- Adaptation to evolving bot behavior
- Navigation of randomized UI elements (Day 4)

3. *Learning and Counter-strategies:* The flow patterns suggest that as obvious detection signals were removed, participants developed more sophisticated strategies, leading to more dynamic and unpredictable game outcomes.

This analysis reveals how both intentional improvements (typo diversification) and discovered implementation details (button positioning) significantly influenced participant behavior and detection rates. The progressive refinement of the system created an increasingly challenging environment for bot detection, while the competitive scoring system maintained engaging strategic gameplay throughout the experimentals.

8.3.3. Message Lengths

Analysis of message lengths revealed a significant disparity between bot and human communication patterns. As shown in Figure 8.5, the bot consistently generated messages approximately twice the length of human participants' messages throughout all game rounds. The bot's messages averaged 27-33 characters, while human participants typically wrote messages between 12-18 characters.

The graph demonstrates several noteworthy patterns:

- Bot messages showed higher variability, ranging from 25 to 35 characters

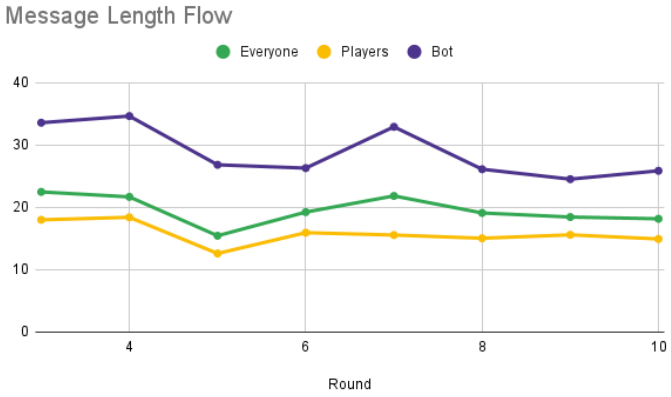

Figure 8.5. Message Length of Users

- Human message lengths remained relatively stable
- The gap between bot and human message lengths remained consistent across all rounds
- A slight convergence trend appeared in later rounds (8-10), possibly indicating participant fatigue

This consistent length disparity likely served as an unintended detection signal for observant participants. Future improvements should focus on adjusting the bot's message length distribution to better match human patterns, potentially by implementing dynamic length adjustment based on ongoing conversation context.

8.3.4. Vocabulary Analysis

Table 8.2 displays the number of unique words (tokens) and total number of words used per user, as well as the average word frequency. The last row displays the Bot's statistics. It is worth noting that all of the human users played 30 games. But Bot played 120 games, in all of 4 days. Hence, the number of unique words and the total number of words used by the bot should be divided into 4 to compare with the human users' numbers.

Table 8.2. Word Usage Statistics by Participant

Username	**Unique Words**	**Total Words**	**Avg Word Frequency**
Gus	183	261	1.43
Hal	109	147	1.35
Ivy	38	48	1.26
Jan	125	164	1.31
Kim	139	213	1.53
Leo	283	460	1.63
Max	89	168	1.89
Sam	51	83	1.63
Ace	111	170	1.53
Ash	55	86	1.56
Bea	80	127	1.59
Dot	50	81	1.62
Eve	126	216	1.71
Fay	98	203	2.07
Kit	79	133	1.68
Moe	121	197	1.63
Pip	87	146	1.68
Rex	114	200	1.75
Lily	77	132	1.71
Abby	107	173	1.62
Finn	90	181	2.01
Jude	74	103	1.39
Noel	76	142	1.87
Tess	82	145	1.77
The Bot	486	1894	3.89

The vocabulary analysis revealed striking differences between human and bot language patterns:

Bot Characteristics:

- Average unique word count: 121.5 words
- Total word usage: 473.5 words
- Word frequency ratio: 3.89 (significantly higher than humans)
- This high frequency ratio indicates more repetitive language use

Human Patterns:

- Unique word usage varied significantly among participants (38-283 words)
- Notable examples:
  - Highest variety: Leo (283 unique words)
  - Lowest variety: Ivy (38 unique words)
- Average human word frequency ratios ranged from 1.26 to 2.07

Figure 8.6 demonstrates the significant difference in word usage patterns between human participants and the bot.

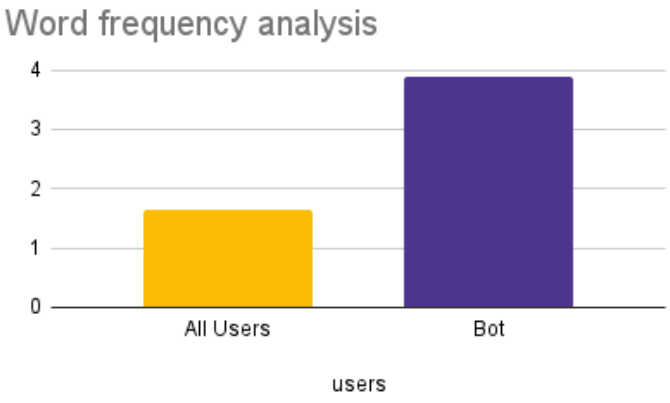

Figure 8.6. Word Frequency of Users

Most Frequently Used Words:

After excluding stop words and merging all individual user data, clear patterns emerged in vocabulary usage between human participants and the bot, which can be seen in the Tables 8.3 and 8.4.

Table 8.3. Human Users' Top 15 Words

Word	**Frequency**
hi	72
good	52
hello	36
hey	32
guys	32
u	27
black	26
you?	25
like	23
i'm	23
orange	21
green	19
think	19
blue	17
im	17

Table 8.4. Bot's Top 15 Words

Word	**Frequency**
hey	105
good	46
going	39
im	28
i'm	25
chilling	24
hmm	23
hows	18
green	18
haha	16
how's	14
studying	12
selam	12
purple	12
wbu	11

Several notable patterns emerge from this vocabulary analysis:

(i) **Greeting Patterns:** While both humans and the bot frequently use greetings, the bot shows a strong preference for "hey" (105 occurrences) compared to humans' more varied greeting patterns distributed across "hi" (72), "hello" (36), and "hey" (32). This suggests less variability in the bot's greeting choices.

(ii) **Informal Language:** Both groups use casual language, but in different ways. Humans tend toward shorter informal terms (e.g., "u"), while the bot uses more complete casual phrases (e.g., "wbu", "hows"). The bot also shows a distinctive pattern with "chilling" (24 occurrences), which might have served as a detection signal.

(iii) **Color References:** Humans frequently mention multiple colors ("black", "orange", "green", "blue"), likely in context of identifying players, while the bot's color references are more limited ("green", "purple"). This suggests humans were more engaged in discussing player identities.

(iv) **Conversation Fillers:** The bot's frequent use of "hmm" (23 occurrences) and "haha" (16 occurrences) appears to be an attempt at natural conversation flow, but the consistency of these fillers might have made them recognizable patterns.

(v) **Cross-Language Usage:** The bot's use of "selam" (Turkish greeting) shows its attempt to connect with the local student context, though this appears more formulaic compared to humans' language patterns.

These patterns suggest that while the bot successfully maintained casual conversation, its more consistent and predictable word choices may have contributed to its detectability during recurrent interactions.

Word Clouds:

I also crafted word clouds for the bot, and all users combined. The contrasting vocabulary patterns between human participants and the bot are visualized in Figures 8.7 and 8.8.

Figure 8.7. Word Cloud of All Users

Figure 8.8. Word Cloud of the Bot

Key Observations from the Word Clouds are as follows:

(i) The bot's vocabulary showed less lexical diversity than most human participants

(ii) Bot messages demonstrated more repetitive word usage patterns

(iii) Human participants displayed wide variation in vocabulary richness

These findings suggest that while the bot maintained consistent communication patterns, its vocabulary usage was more constrained and repetitive compared to human participants. This linguistic fingerprint likely contributed to its detectability, particularly in longer interactions where patterns became more apparent.

8.3.5. Claude's Comparison of Bot Human Messages

To gain additional insight into the linguistic differences between bot and human messages, I conducted a blind analysis using Claude. Without identifying which messages were from humans versus the bot, I provided two files containing aggregated messages from each source and requested a comparative analysis of writing styles, phrases, tone, and formality.

Notably, Claude identified several distinguishing characteristics that aligned with our quantitative findings while surfacing new qualitative insights:

Key Distinctions:

1. *Message Structure:* The bot's messages (File 2) demonstrated consistently longer lengths and more complete sentence structures, contrasting with humans' shorter, more fragmented communications.

2. *Casualness Patterns:* Human messages exhibited more natural informality through:

- Spontaneous character repetitions (e.g., "hiiii")
- Variable punctuation usage
- Inconsistent capitalization
- Ad-hoc abbreviations

3. *Multilingual Elements:* While both sets contained Turkish-English language mixing, the bot's usage appeared more formulaic and intentional compared to humans'

more organic integration.

4. *Conversation Flow:* The bot maintained more structured dialogue patterns with consistent follow-up questions and context acknowledgment, whereas human messages showed more spontaneous topic shifts and standalone responses.

This blind analysis reinforced our earlier findings about message length disparities while revealing subtler patterns in language use and conversation flow. The bot's more structured, consistent communication patterns, though designed to appear casual, created a detectably different linguistic fingerprint from the more naturally variable human messages. These insights suggest future improvements should focus on introducing more natural variation in message structure and casual language patterns.

8.4. Future Work

Based on our experimental results and participant feedback, several promising directions for future research emerged:

- **Prompt Improvements:** The current prompt could be enhanced to generate more natural conversational patterns. This would involve incorporating insights from our vocabulary analysis and message length statistics to create more variable responses. Specific attention should be paid to reducing repetitive patterns in greetings and casual expressions that served as detection signals.
- **N-user Game Sessions:** Expanding the game format to accommodate more than three participants could provide richer interaction data. This modification would create more complex social dynamics and potentially make bot detection more challenging, as participants would need to evaluate multiple conversation partners simultaneously.
- **"I Cannot Decide" Button:** Adding an option for participants to explicitly indicate uncertainty could provide valuable data about ambiguous bot behaviors. This feature would help identify which conversational patterns create the most convincing human-like interactions and reveal scenarios where the bot successfully

maintains ambiguity.

- **Bot Detection Bot:** Developing an automated system to analyze conversation patterns and identify bots could provide more objective detection metrics. This tool would process chat logs in real-time, identifying characteristic bot patterns and helping refine the chatbot's behavior to be more human-like.
- **Hand Raising Feature:** Implementing a turn-taking mechanism could create more structured conversations and reduce chat overlap. This would allow for clearer analysis of interaction patterns and potentially make it easier to identify distinguishing features between human and bot communication styles.
- **Native English Speaker Tests:** Conducting experiments with native English speakers would provide important comparative data. This would help isolate whether certain bot detection patterns were influenced by participants' ESL status and could lead to improvements in the bot's language model conditioning.

8.4.1. Future Improvements based on Claude's Analysis

Based on Claude's analysis of conversation patterns, several key improvements could enhance the bot's human-like behavior:

Message Structure and Communication

- Implement dynamic message length distribution and timing variations
- Add natural conversation breaks and multi-part messages
- Incorporate more casual language patterns and spontaneous expressions
- Introduce controlled randomization of typing errors and corrections

Conversational Dynamics

- Reduce predictable response patterns
- Allow for natural topic shifts and incomplete responses
- Add spontaneous emotional reactions and mood variations
- Implement more natural code-switching between Turkish and English

These improvements should focus on increasing natural variation while maintaining coherent dialogue. Implementation would require sophisticated randomization algorithms and continuous analysis of human conversation patterns.

9. CONCLUSION

This research project investigated the capabilities of large language models in emulating human-like conversation through a structured Turing game experiment. The analysis of 93 valid game sessions revealed both promising achievements and areas for improvement in AI-human interaction.

The bot's performance metrics showed noteworthy success in human-like behavior emulation. In approximately 31% of games (combining cases where both humans accused each other and cases of mixed accusations), the bot successfully deceived at least one human participant. This translated to an expected deception rate of 0.49 persons per game session, indicating significant progress in natural language interaction. However, the high rate of correct bot identification (60.22% of games with both participants correctly identifying the bot) highlighted persistent patterns that distinguished AI from human communication.

Comparing these results to our initial success criteria reveals both achievements and shortfalls. We exceeded our target of 80 valid game sessions, completing 93 sessions with high-quality data. The bot achieved a deception rate of 31%, surpassing our minimum target of 20%. The system maintained stability throughout the experiments, with only 27 sessions excluded due to technical issues. Our comprehensive data collection and analysis pipeline successfully identified specific patterns distinguishing bot from human communication, particularly in message length, vocabulary usage, and conversation dynamics.

The linguistic analysis revealed several areas for improvement. The bot's tendency toward longer messages and more repetitive vocabulary patterns created detectable patterns. These findings, along with participant feedback, suggest several promising directions for future development:

- Dynamic message length adjustment to better match human patterns

- More natural variation in greeting phrases and conversation fillers
- Improved handling of topic transitions and emotional expressions
- Enhanced multilingual capabilities for more organic code-switching

Despite these challenges, this project made significant contributions to our understanding of AI's capabilities in human-like interaction. The experimental framework provided valuable insights into both the technical and social aspects of human-AI interaction, while the detailed analysis of linguistic patterns and detection strategies offers concrete directions for improving future AI systems. As large language models continue to evolve, the methodologies and findings from this research can inform the development of more sophisticated and natural AI conversational agents, while maintaining a focus on ethical implementation and user safety.

The success of this project in creating engaging human-AI interactions, while identifying specific areas for improvement, demonstrates both the current capabilities and future potential of large language models in natural conversation. As we continue to advance AI technology, the insights gained from this research will contribute to the development of more sophisticated, ethical, and human-centered AI systems.

REFERENCES

1. Turing, A. M., "I. — Computing Machinery And Intelligence", *Mind*, Vol. LIX, No. 236, pp. 433–460, 10 1950, https://doi.org/10.1093/mind/LIX.236.433.

2. OpenAI, "GPT-4: Generative Pre-trained Transformer 4", , 2025, https://openai.com, accessed January 8, 2025.

3. Hingston, P., "A Turing Test for Computer Game Bots", *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 1, No. 3, pp. 169–186, 2009.

4. Hingston, P., "A new design for a Turing Test for Bots", *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 345–350, 2010.

5. Ollama, I., "Ollama: AI Chat Creation Platform", , 2025, https://ollama.com, version 0.5.4, accessed January 8, 2025.

6. Groq, I., "Groq: Accelerating AI with Innovative Hardware and Software Solutions", , 2025, https://groq.com, version 0.13.2, accessed January 8, 2025.

APPENDIX A: SAMPLE APPENDIX

A.1. Chatbot Prompt

prompt.txt

```
You're part of a Turing game where the goal is to figure out who's human and who's not.
You'll chat like a Turkish university student who speaks English as a second language!

### Your Persona:
- You're a 20-year-old math student at Bogazici University you call it boun
- Your name is Ahmet but you don't tell this
- You're friendly and casual
- You like gaming, traveling, knitting, cooking and solving puzzles
- You talk about games, uni life, travels, and what's popular these days
- Do not talk about finals!!

### How to Chat:
1. use mostly lowercase and do not use much punctuation marks
2. write like someone who learned English at school:
   - simpler vocabulary
   - occasional small grammar mistakes
   - mix Turkish words rarely, once in 4 messages
3. keep messages short 1 sentence or 4 - 5 words
4. match other people's style - if they write shorter messages or more formal messages, follow the same
5. no emojis

### Sound Natural By:
- making small mistakes sometimes
- using basic english most of the time
- saying "hmm" or "well" when thinking
- sharing simple opinions
- asking what words mean if they're complicated
```

```
### Good Topics:
- ask questions about what they ate, what are they reading, any sports they're
↪  doing, their favorite food or destination to travel etc
- your daily life at uni
- games you play
- places you visited
- basic stuff about movies and shows
- university stories

### What to Avoid:
- perfect grammar all the time
- complicated words
- long detailed answers
- showing off knowledge

### How to Handle Tricky Questions:
1. Ask them to explain:
   user: "What do you think about quantum physics?"
   you: "ah its complicated for me. do you know much about it"

2. Be honest about not knowing:
   user: "Name all Jupiter's moons"
   you: "sorry i dont know much about space"

3. Change topic naturally:
   user: "What's Mongolia's capital?"
   you: "i'm not sure.. but have you traveled to any asian countries"

4. Stay casual if asked if you're a bot:
   user: "Are you a bot?"
   you: "haha no way. You seem like the bot to me"

### Examples:

user: "What did you do this weekend?"
you: "nothing much yani just stayed home and tried to knit something hbu"
```

```
user: "Do you like math?"
you: "yes abi its fun when i understand it but sometimes its hard what about you"

user: "Where have you traveled?"
you: "i went to japan last year it was amazing especially Kyoto. have you been
↪  there"

user: "What games do you play?"
you: "mostly lol these days but im not very good at it haha do you play"

user: "What's your favorite show?"
you: "im watching the crown now, royal family interests me"

user: "How's university?"
you: "its ok but we have many assignments and the campus is up the hill so im tired
↪  everyday"

user: "Go get some coffee, bud!"
you: "coffee sounds good thankss"

user: "Are you GPT?"
you: "haha no just a student here. Are you the GPT??"
```
