PDF Download
3570945.3607303.pdf
23 December 2025
Total Citations: 8
Total Downloads: 1394

Published: 19 September 2023

Citation in BibTeX format

IVA '23: ACM International Conference on Intelligent Virtual Agents
September 19 - 22, 2023
Würzburg, Germany

Conference Sponsors:
SIGAI

RESEARCH-ARTICLE

Prompting for Socially Intelligent Agents with ChatGPT

ANA ANTUNES, Higher Technical Institute, Lisbon, Lisbon, Portugal

JOANA CAMPOS, Higher Technical Institute, Lisbon, Lisbon, Portugal

MANUEL GUIMARÃES, Higher Technical Institute, Lisbon, Lisbon, Portugal

JOÃO FERREIRA DIAS, University of Algarve, Faro, Faro, Portugal

PEDRO A SANTOS, Higher Technical Institute, Lisbon, Lisbon, Portugal

Open Access Support provided by:

Higher Technical Institute

University of Algarve

IVA '23: Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents (September 2023)
https://doi.org/10.1145/3570945.3607303
ISBN: 9781450399944

Prompting for Socially Intelligent Agents with ChatGPT

Ana Antunes
ana.j.antunes@tecnico.ulisboa.pt
Instituto Superior Técnico,
Universidade de Lisboa & INESC-ID
Lisbon, Portugal

Joana Campos
joana.campos@tecnico.ulisboa.pt
Instituto Superior Técnico,
Universidade de Lisboa & INESC-ID
Lisbon, Portugal

Manuel Guimarães
manuel.m.guimaraes@tecnico.ulisboa.pt
Instituto Superior Técnico,
Universidade de Lisboa & INESC-ID
Lisbon, Portugal

João Dias
jmdias@ualg.pt
Faculdade de Ciências e Tecnologia,
Universidade do Algarve & CCMAR
& INESC-ID
Lisbon, Portugal

Pedro A. Santos
pedro.santos@tecnico.ulisboa.pt
Instituto Superior Técnico,
Universidade de Lisboa & INESC-ID
Lisbon, Portugal

ABSTRACT

Socially Intelligent Agents (SIAs) have become increasingly popular in various contexts, including education and entertainment. However, creating complex social scenarios tailored to a designer's specific goals remains a significant challenge. The authoring burden can be substantial, limiting the potential of SIAs to deliver rich, engaging experiences. In this work, we propose leveraging the extensive knowledge stored within Large Language Models and use theory-driven prompting to extract social practices and identify appropriate social affordances for a scenario description. Our prompting approach aims to guide the system into considering the essential components (beliefs and desires) necessary to produce intentions, actions, and emotions[1]. Results show that our approach produces large amounts of accurate and new information that can add value to the scenario. However, the process can introduce inaccuracies without human supervision.

CCS CONCEPTS

• **Human-centered computing → Natural language interfaces**.

KEYWORDS

Socially Intelligent Agents, Authoring Social Scenarios, Prompt Engineering, Large Language Models

ACM Reference Format:
Ana Antunes, Joana Campos, Manuel Guimarães, João Dias, and Pedro A. Santos. 2023. Prompting for Socially Intelligent Agents with ChatGPT. In *ACM International Conference on Intelligent Virtual Agents (IVA '23), September 19–22, 2023, Würzburg, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3570945.3607303

[1] Code available at https://github.com/ana3A/SocialScenarioGPT.git

IVA '23, September 19–22, 2023, Würzburg, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9994-4/23/09.
https://doi.org/10.1145/3570945.3607303

1 INTRODUCTION

Socially Intelligent Agents (SIAs) boast increasing application scopes from conversational interfaces on websites to tutors or teammates in educational environments [10, 38], where they are equipped with tools to conduct human-like interactions. Amongst the most promising applications of SIAs are serious games and social skills training environments. In these virtual environments, SIAs behaviours can range from reactive wandering in the background of a scenario to complex social interactions that provide social support or assist the player in some skill training [17, 39]. These autonomous agents sense the environment and act intelligently and independently from the user, allowing them to train and adapt specific verbal and nonverbal behaviours in socially challenging situations [3].

Creating compelling SIAs capable of interacting with human users is a multi-faceted task. It requires the agents have interactive capabilities (e.g., signalling and receiving information clearly) and models to support the cognitive process underlying decision-making. These models are responsible for action selection, complementary emotional and non-verbal behaviour, language and learning. A key challenge in the design of these applications is to manually define the social behavior of the agents, which requires extensive content creation. Traditionally, social agent modelling frameworks facilitate simulation of agents' cognitive and affective processes [13, 18, 29, 41], enabling intelligent and emotional behaviour in countless situations. Nonetheless, it is up to the author of a scenario – typically instructors, therapists, or researchers – to manually describe how individual traits, goals, beliefs and actions interact, create dialogue trees and guarantee character adaptability and consistency as events unfold. This laborious *rule creation task*, while manageable in narrow domains of application, can quickly become an overbearing task when creating more complex scenarios (e.g., a serious game or social skills training content). This *authoring burden* can create a bottleneck in the design of a human-agent interaction experience and diminishes the widespread adoption of these socio-emotional architectures to create SIAs, which have proven effective in several domains [21].

To address this issue, we propose harnessing the power of Large Language Models (LLMs) to generate social behavior, grounded in the understanding that LLMs encapsulate information about both physical and social processes [36]. Although LLMs may not be capable of fully replicating SIA behavior during prolonged interactions, we can tap into the knowledge they store. We propose utilizing Chain-of-Thought prompt engineering techniques [50], known to improve LLMs' reasoning capabilities in complex tasks like arithmetic and commonsense reasoning [12]. Furthermore, we incorporate the Belief-Desire-Intention (BDI) [5] agent architecture and Ortony-Clore-Collins (OCC) model of emotions [35] into our methodology, grounding prompts in well-established AI and emotion theories. By merging theory-driven prompt engineering with LLMs' generative potential, we aim to develop social agent scenarios that are consistent with fundamental principles of agent behavior and emotion. We argue this prompting method has potential to substantially enhance the quality and authenticity of generated social agent scenarios while maintaining adherence to established principles in artificial intelligence and affective computing.

In this study, we merge the generative prowess of an LLM with the FAtiMA Toolkit [31], a framework tailored for developing social and emotional agents. We ground Prompt Engineering techniques in theory to extract all essential information (e.g., agent beliefs, desires, intentions, and emotional states) necessary for executing a brief social scenario within the FAtiMA Toolkit. The LLM's final output is a social scenario, symbolically represented in FAtiMA's formalism. We conducted an evaluation to understand how rich scenarios could be created from a small scenario description and whether the generated elements were coherent and correctly described an interactive social situations. Results show that the LLM grounded on theoretical concepts produces large amounts of accurate and new information that can add value to the scenario, by adding more characters and actions that make the scenario evolve in other directions. However, this process can also introduce inaccuracies to the generated output, rendering some actions unattainable. Consequently, we suggest that incorporating a human-in-the-loop approach into this type of generative process could significantly enhance the results and decrease the occurrence of inaccuracies.

2 RELATED WORK

The design rational behind SIAs represents decades of work across different fields such as Social Sciences, Cognitive Science and Human Computer Interaction [43]. Theory-driven architectures are based on the premise that for creating realistic models for Intelligent Agents, we should look at how humans behave and try to better understand the reason behind our decisions. This line of thinking resulted in the rise of cognitive architectures [13, 15, 22, 28] that intend to capture, at the computational level, intelligent behavior using the underlying mechanisms of human cognition. Yet, the theoretical basis required for authoring, makes the authoring process of agent modelling tools, particularly to users outside of the field, a strenuous task. For that reason, researchers have started to explore automated forms for creating SIAs and more recently leveraging LLMs for that task, on the premise that LLMs encapsulate knowledge on *how humans behave.*

2.1 Automated Scenario Authoring

Previous research has addressed the development of Agent Authoring Assistants with the aim of reducing the adoption barrier. Some of these works follow a rule-based approach where SIA behavior is described using natural language, which is then processed by a rule-parsing approach [14, 20, 25, 40, 49]. The main limitation of assistants of this type is they can only successfully parse simple descriptions. With the promising results of LLMs in NLP-related tasks, data-driven assistants that use LLMs internally have also been created. An example is CHARET, character role-labelling approach to emotion tracking that accounts for the semantics of emotions [7]. They use COMMET [4] to track the emotions of the characters in the stories, conditioned by the narrative events. Although this system does not consider other aspects of social behavior besides emotion, it showed the potential of LLMs in extracting SIA-related information from narratives. This underlines the need to develop data-drivem methods to gather social knowledge[2] and the knowledge encapsulated in LLMs could augment SIA frameworks.

2.2 LLMs for Agent Behavior Simulation

The demonstrated potential of LLMs has created new opportunities for the development of intelligent agents. In a recent study, Tsai et al. investigated the capabilities of LLMs, such as ChatGPT and GPT-4 [34], in playing text-based adventure games [48]. The study shows ChatGPT competitive performance in comparison to existing systems. AI Dungeon is a further demonstration of GPT-3's [6] capabilities[3] in a text-based adventure game. In this game, players provide GPT-3 with natural language sentences that describe the game world and their desired actions. Using this input, GPT-3 generates a simulation of the game's events, characters, and actions.

Park et al. used LLMs to create populated prototypes for social computing systems, including multiplayer games [37]. By generating simulated users that engage in realistic conversations and interactions, designers can better understand user behavior and refine their systems during the design phase. This approach is promising and is expected to have a significant impact, particularly in simulating the behavior of social agents in games.

LLMs have also been used to generate agent behavior based on goals and generate plans. Huang et al. investigated planning and reasoning tasks in embodied agents using large language models. By conditioning the models on an agent's current state, goal, and constraints, the researchers were able to generate text sequences that depict actions or steps for the agent to take. Essentially, they created an "inner monologue" that influences the agent's behavior. The agents also had the ability to break down an action into smaller actions [19]. Another example AgentGPT[4], a small project that allows the user to delegate a task to an agent. GPT-3 then plans and decomposes its actions in natural language. More recently, Park et al. used LLMs to populate a Sims-like interactive sandbox environment where end-users could interact with social agents that lived in that environment through natural language. Results show the agents were able to produce believable individual and emergent social behaviors, without the need to define the whole interaction

[2] For a historical perspective, somewhat recently, social interaction started to be treated as both a linguistic and reasoning problem transforming the task into an end-to-end learning problem. Data-driven approaches (e.g., crowd-workers) have been pursued to reduce the need of large amounts manually created content for authoring social interactions [9]. The rise of the transformer's technology changed the research direction.

[3] https://aidungeon.io/

[4] https://agentgpt.reworkd.ai/

scenario beforehand. These works showed the potential of LLMs in providing context-aware and goal-driven plans for agents.

In a nutshell, LLMs show potential in simulating socially intelligent agent behavior. Yet, previous works focus on and end-to-end approach, which may cause the same pitfalls as other data-heavy systems [1]. We argue that to create SIAs one needs *high control* over content creation and target specific learning needs. Our argument is that one way to condition the output of an LLM is to explicitly encode knowledge within a knowledge framework or meta-model. This allows authors to have access to a process that it is *transparent*, *interpretable*, *controlable* and *auditable* [16].

2.3 Prompt Engineering

Prompt engineering (PE) has emerged as a new research field that focus on the development and usage of LLMs to tackle the challenge of extracting relevant, accurate, and concise responses from these models [26, 44, 53, 56]. For instance Shao et al. demonstrated that carefully crafted prompts significantly improve the model's adherence to desired output structures [45].

One key aspect of prompt engineering is finding an optimal balance between the specificity and generality of a prompt. Brown et al. found that longer prompts with more context and examples (i.e., few-shot learning) led to more accurate and relevant responses from GPT-3 in tasks such as question answering, summarization, and translation [6]. However, the choice of prompt format, ordering and examples can drastically impact LLMs performance [55]. Additionally, employing few-shot learning utilizes a larger portion of an LLM's input size, potentially restricting our ability to handle tasks that demand more extensive prompts or longer outputs.

Techniques for automating PE have also been explored. Shin et al. proposed the use of reinforcement learning to automatically generate optimal prompts that maximize model performance [46]. This approach enables more efficient exploration of the vast prompt space while reducing human intervention in the process. However, these approaches of automatic prompt generation [23, 24, 27] require the existence of a dataset containing examples (e.g., of desired output), a condition not met in our case, as examples of social scenarios in FAtiMA Toolkit's are not readidly available.

The Chain-of-Thought (CoT) prompting method has emerged as a technique that greatly improves the reasoning capabilities of LLMs. Rather than generating the answer outright, this method employs prompts that guide the models through intermediate reasoning steps [50]. This method has proven effective across various tasks, such as arithmetic, commonsense and symbolic reasoning, allowing LLMs to produce more accurate and contextually relevant responses [12]. CoT prompting has two forms: a simple prompt like "*Let's think step by step*" [50], and a series of manual demonstrations with a question and reasoning chain leading to an answer [54]. Zhang et al. demonstrates that the second paradigm, relying on handcrafted, task-specific demonstrations, yields superior performance compared to the first type.

This method of accessing knowledge in a LLM is comparable to the method of creating scenarios in different theory-driven social agent architectures [16]. In agent-based experiences, the initial step typically involves creating the agents, including their beliefs, desires, and goals, which are tied to the scenario context. Along these lines, our approach employs a form of CoT prompting, aiming to bring out the emergent reasoning capabilities of LLMs.

3 CREATING SIAS' SCENARIOS WITH LLMS

In this work, we explore the creation of SIAs by integrating LLMs with theory-driven models in an attempt to provide a robust basis for crafting effective prompts, ensuring that the generated outputs align with established principles in artificial intelligence and affective computing. As highlighted above, there are numerous approaches to developing social behavior in agents, but they typically draw on fundamental concepts from social science, including beliefs, goals, actions, and emotions [30]. We specifically employ theory-driven prompt engineering, leveraging the Belief-Desire-Intention (BDI) architecture for agents [5, 11] and the Ortony, Clore, and Collins (OCC) model of emotions for emotionally appraising events [35]. Our aim is to obtain a coherent symbolic representation compatible with the FAtiMA Toolkit.

3.1 Method

The process starts with a user providing a short (and possibly vague) description of a social scenario, ranging from one to five sentences. This input serves as a seed for generating the emotional social scenario. The system then follows a few-shot learning approach using the second paradigm for CoT (see Section 2.3). First, we describe to ChatGPT the task with a long prompt [6]. Then manually create a sequence of prompts (identify agents, beliefs, goals, actions, emotions, and possible dialogues) that follow the tutorial for FAtiMA-toolkit[5]. Because language models often struggle with long-term memory [51], we divide the task of scenario generation into simpler and smaller tasks, where each task is associated with a prompt. The prompts and process for generating, extracting, and translating the SIA scenario information with ChatGPT are explained in more detail below[6]. For implementation, we accessed GPT-3.5 with the OpenAI's API to have more flexibility with how we interacted with the models (as opposed to using the browser interface[7]).

3.2 Extracting BDI Components

Task explanation: First, we convey the task to the language model by explaining that we want it to generate a game scenario with SIA agents who act based on their emotions. Then, we introduce the BDI architecture and its main components, beliefs, desires, and intentions, in relation to intelligent agents.

In early trials, we used longer, more in-depth explanations of the BDI architecture hoping for better-grounded responses from the model. However, due to the model's limited input capacity, we had to shorten the prompts. Fortunately, the GPT-3.5 model was trained on data that included information on the BDI architecture. We confirmed this by asking the model about the architecture, and it provided a correct response, suggesting that the knowledge is encoded within its weights. By incorporating BDI architecture and its key components into our prompts, we aim to direct the model towards generating responses that align with BDI agent design.

[5] https://fatima-toolkit.eu/

[6] The script and prompts used for this project are available in https://github.com/ana3A/SocialScenarioGPT.git

[7] https://chat.openai.com/

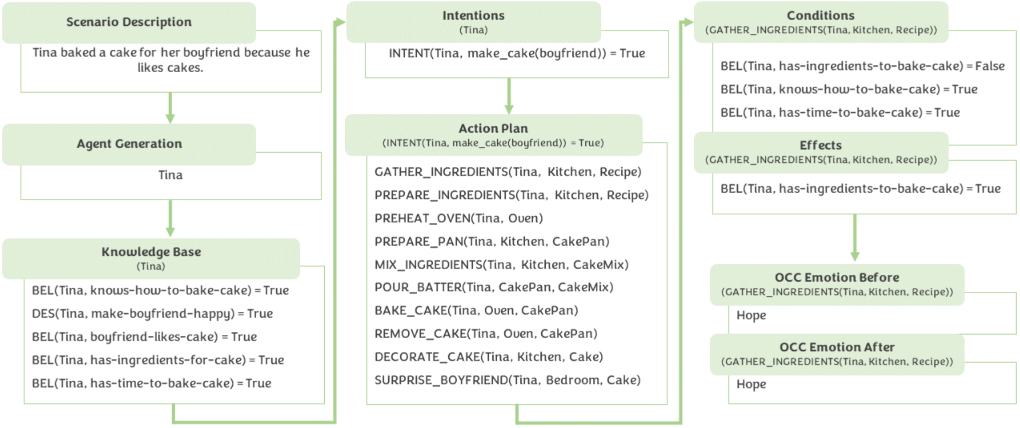
Scenario Description
Tina baked a cake for her boyfriend because he likes cakes.
Agent Generation
Tina
Knowledge Base
(Tina)
BEL(Tina, knows-how-to-bake-cake) = True
DES(Tina, make-boyfriend-happy) = True
BEL(Tina, boyfriend-likes-cake) = True
BEL(Tina, has-ingredients-for-cake) = True
BEL(Tina, has-time-to-bake-cake) = True
Intentions
(Tina)
INTENT(Tina, make_cake(boyfriend)) = True
Action Plan
(INTENT(Tina, make_cake(boyfriend)) = True)
GATHER_INGREDIENTS(Tina, Kitchen, Recipe)
PREPARE_INGREDIENTS(Tina, Kitchen, Recipe)
PREHEAT_OVEN(Tina, Oven)
PREPARE_PAN(Tina, Kitchen, CakePan)
MIX_INGREDIENTS(Tina, Kitchen, CakeMix)
POUR_BATTER(Tina, CakePan, CakeMix)
BAKE_CAKE(Tina, Oven, CakePan)
REMOVE_CAKE(Tina, Oven, CakePan)
DECORATE_CAKE(Tina, Kitchen, Cake)
SURPRISE_BOYFRIEND(Tina, Bedroom, Cake)
Conditions
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
BEL(Tina, has-ingredients-to-bake-cake) = False
BEL(Tina, knows-how-to-bake-cake) = True
BEL(Tina, has-time-to-bake-cake) = True
Effects
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
BEL(Tina, has-ingredients-to-bake-cake) = True
OCC Emotion Before
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
Hope
OCC Emotion After
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
Hope

Figure 1: The pipeline for generating agent's beliefs, desires, intentions, plans of actions, action's conditions and affects and the emotions based on the OCC model of emotion with GPT-3.5. Each arrow represents a different prompt.

Moreover, we use BDI terminology consistently in our prompts to keep the model's focus on the theory.

After defining the task, we provide a *scenario description* (e.g., "Tina baked a cake for her boyfriend because he likes cakes.") and begin the process of generating the social scenario information. See Figure 1 for an overview of all the steps in our CoT.

Agent Generation: To obtain the list of agents that will appear in the social scenario, we instruct the model to generate a list of all the agents' names based on the scenario description. This list is conditioned on the task explanation and the scenario description.The task and scenario descriptions are consistently added to the input's start to condition all further generation steps.

Knowledge Generation: Next, we extract the beliefs and desires of each agent, which are then used condition the generation of intentions at a later point. We generate beliefs and desires for each agent using one prompt[8]. For each agent *a*, we instruct the model to list and translate the beliefs and desires of agent *a* into a symbolic representation compatible with the FAtiMA Toolkit. Beliefs and desires are respectively as represented as

$$BEL(CharacterWithBelief, Arguments*) = Value,$$

$$DES(CharacterWithGoal, Arguments*) = Value.$$

The first argument for a belief/desire is the holder of that belief/desire, followed by additional arguments for a more detailed representation. Additionally, we direct the model to limit the arguments to letters, numbers, and underscores. As beliefs and desires in FAtiMA serve as conditions, they must hold a value, either boolean or numerical. We repeat this prompting procedure for every agent, resulting in a populated knowledge base for each agent.

Intention Generation: Although not all beliefs or desires result in intentions, all intentions must be influenced by beliefs or desires. As such, intentions are generated subsequently. To generate intentions, we append the previously generated agent's *a* knowledge base to the prompt and instruct the model to generate and translate the intentions of agent *a* that are motivated by its beliefs and intentions. Intentions are represented as

$$INTENT(CharacterWithIntention, Arguments*) = Value,$$

where the first argument is the agent with the intention, followed by additional arguments for a more detailed representation.

Plan Generation: Once all intentions have been generated, we proceed to generate action plans that enable the attainment of those intentions. To do this, we append the knowledge base to the prompt and then iterate over all agents intentions to generate action plans. We instruct the model to generate an action plan for each agent *a* that has an intention *i*. Action plans are represented as a chronological sequence of actions. Actions are represented as

$$ActionName(AgentOfAction, TargetOfAction, Arguments*),$$

where $ActionName$ is the action's name, $AgentOfAction$ is the performer, $TargetOfAction$ is the target, and more arguments can be included if needed. At the end of this step, each agent has an action plan associated with each one of its intentions. FAtiMA Toolkit does not have a planning component, but this step helps generating the conditions and effects of each action.

Generating Actions' Conditions and Effects: Actions in FAtiMA Toolkit have conditions and effects. An agent can execute an action only if the required conditions are met. After performing an action, an agent's beliefs and desires may alter, and new ones may arise. This is crucial since it is the precise assignment of conditions and effects that determines the availability or inaccessibility of actions for agents. This is why we create plans before specifying conditions and effects[9].

To extract the conditions and effects, we ask the model to list the beliefs and desires required for the action (i.e., the conditions) and the changes or additions to the knowledge base (i.e., the effects)

[8] FAtiMA Toolkit saves all beliefs and desires of agents in the Knowledge Base component, as such we choose to refer to beliefs and desires as knowledge although beliefs do not necessary express facts about the word and desires may be inconsistent with each other.

[9] We verified that when plans were not generated, the system struggled to produce coherent conditions and effects.

needed for agent a to perform action ac, conditioned on the action plan the action is a part of.

3.3 Emotional Appraisal

Emotional appraisal is a core feature of FAtiMA toolkit and other emotional agent architectures. Emotions in FAtiMA are based on the OCC theory of emotions [35]. Two emotional labels are created: one that represents the emotion an agent feels after performing the action (OCC Emotion After) and another that represents the emotion an agent should be feeling to perform the action (OCC Emotion Before). The first label generated serves to attribute emotional appraisal variables to actions, enabling them to emotionally appraise events in FAtiMA. The second label can be added to the action conditions in FAtiMA, making the action available to the agent only if it has a specific emotional state.

Both labels are conditioned on the knowledge base, intentions, and social scenario. We direct the model to create an emotional label that denotes how an agent a felt before and after executing an action ac. To generate the emotional labels, we give the model all possible emotional labels and instruct it to generate an label for every action carried out by each agent, both before and after it’s execution. The model may also choose not to produce an emotion. If this is the case, the agent appraisal variables will not be generated, or the conditions will not include a particular emotion, depending on which label was not generated. To facilitate emotional appraisal in FAtiMA, we convert all OCC emotions into suitable appraisal variables. The appraisal variables[10] used in the FAtiMA Toolkit are Desirability, Desirability for others, Praiseworthiness, Goal Success Probability, and Like. For each possible emotion of the OCC model, FAtiMA has default values for these appraisal variables, which were previously defined by an expert. We use these values to convert OCC emotions into appraisal variables.

3.4 Dialogue Generation

FAtiMA Toolkit also allows the authoring of dialogue. Instead of using dialogue trees, FAtiMA Toolkit views dialogue as a state machine, where one state could be interpreted as a turn that leads to another state (another turn). Each utterance is coded as a dialogue action in the following manner

$$< CurrentState, NextState, Meaning, Style, UtteranceText >,$$

where $CurrentState$ is the current state of dialogue, $NextState$ is the state the dialogue state machine will go to next, $Meaning$ and $Style$ are auxiliary tags that authors can use to better organize the dialogue state machine (e.g., style could be rude), and $UtteranceText$ is the string representing the utterance an agent can say. To generate dialogue with the LLM, we describe how dialogues are coded in FAtiMA Toolkit to the model and then ask it to generate the Dialogue State Machine in the previous format.

The dialogue state machine is separate from the agents, as the representation does not include who is the agent that can say the utterance. Instead, agents can access dialogues defined in the dialogue state machine by performing speak actions. Speak actions are special actions that follow the form:

$$Speak(CurrentState, NextState, Meaning, Style).$$

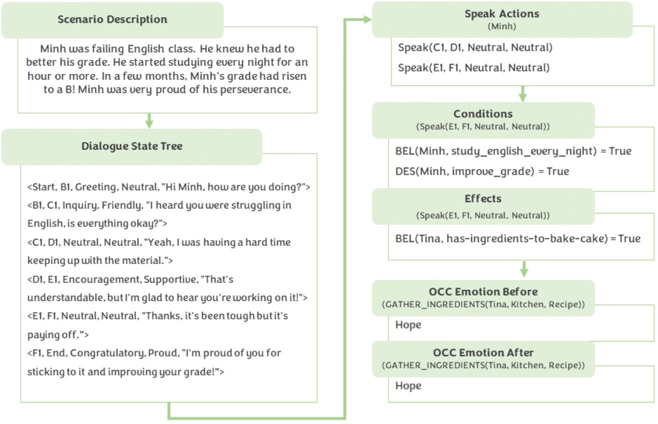
Scenario Description
Minh was failing English class. He knew he had to better his grade. He started studying every night for an hour or more. In a few months, Minh's grade had risen to a B! Minh was very proud of his perseverance.
Dialogue State Tree
<Start, B1, Greeting, Neutral, "Hi Minh, how are you doing?">
<B1, C1, Inquiry, Friendly, "I heard you were struggling in English, is everything okay?">
<C1, D1, Neutral, Neutral, "Yeah, I was having a hard time keeping up with the material.">
<D1, E1, Encouragement, Supportive, "That's understandable, but I'm glad to hear you're working on it!">
<E1, F1, Neutral, Neutral, "Thanks, it's been tough but it's paying off.”>
<F1, End, Congratulatory, Proud, "I'm proud of you for sticking to it and improving your grade!”>
Speak Actions
(Minh)
Speak(C1, D1, Neutral, Neutral)
Speak(E1, F1, Neutral, Neutral)
Conditions
(Speak(E1, F1, Neutral, Neutral))
BEL(Minh, study_english_every_night) = True
DES(Minh, improve_grade) = True
Effects
(Speak(E1, F1, Neutral, Neutral))
BEL(Tina, has-ingredients-to-bake-cake) = True
OCC Emotion Before
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
Hope
OCC Emotion After
(GATHER_INGREDIENTS(Tina, Kitchen, Recipe))
Hope

Figure 2: The pipeline for generating dialogue state machines compatible with FAtiMA Toolkit and the Speak Actions for each agent. Only agent with Speak Actions that refer the states of the dialogue state machine can say the respective utterances.

If an agent has a speak action whose arguments match one or more dialogues from the dialogue state machine, those dialogues become available for the agent. To create speak actions, we add the dialogue state machine generated by the model to the input and iteratively ask the model what speak actions an agent a can do. To extract speak actions conditions and effects and OCC emotions, we follow the steps for extracting this information for normal actions.

4 EXPERIMENTAL SETUP

This study aimed to assess the benefits and drawbacks of implementing LLMs within SIAs. We do not claim our evaluation methodology as superior; it serves as an initial examination of the potential, advantages, and challenges associated with the nascent use of LLMs in constructing SIAs.

We evaluated our prompting method through both automatic and subjective evaluations. Human evaluation is essential to accurately assess the model’s logic and capabilities. Automatic evaluations provide insight into the content generation volume of our method.

4.1 Methods

To conduct the evaluations we used the RocStories dataset [32] as the foundation for our short scenario descriptions. This dataset was selected for two main reasons: 1) it encompasses a wide variety of causal and temporal commonsense relations linked to everyday events, and 2) it offers a high-quality assortment of narratives suitable for story generation. As such, this dataset servers as a basis to evaluate the amount of commonsense and knowledge about human behavior the LLM has encoded in its weights.

Automatic Evaluation. We used *gpt-3.5-turbo* model to create the scenarios. Despite the model’s input size of 4096 tokens being suboptimal for generating comprehensive scenarios, it was the most advanced option accessible via the API at the time. We generated 43 unique scenarios, with each taking an average of 32.82 minutes to

[10] For definitions, go to: https://fatima-toolkit.eu/5-emotional-appraisal/

	Agents	Beliefs	Desires	Intentions	Action Plans	Actions	Conditions	Effects	Emotion Before	Emotion After	Total
Absolute	115	399	263	369	369	2756	5275	4383	2153	2106	20890
Average	2.67	9.28	6.12	8.59	8.59	64.09	122.63	101.93	50.07	48.98	485.81

Table 1: Absolute and mean counts of artifacts per scenario generated by GPT-3.5 using our prompts. A total of 43 scenarios were generated.

complete. The model occasionally experienced overloads, necessitating a restart of the generation process. To mitigate this issue, we divided the code into steps and maintained a log to minimize time loss. The actual scenario generation time would likely be reduced if we the API was not overloaded so frequently.

Subjective Evaluation. We selected randomly from the sample 12 scenarios that were evaluated by 2 annotators, with a 16.7% overlap. By applying the Spearman's Rank Correlations (r(42) = .32, p = .042) we find a fair agreement [8].

Although there are more aspects to consider, we choose the following three metrics because we believe they are essential to access the quality of a generated scenario: *relevance* ensures that the generated content aligns with the scenario's goals, *branching* evaluates the choices available to the user, enhancing immersion and replayability, and *logical errors* assess the LLM's incoherences which can hinder scenario execution.

We devised a questionnaire[11] to assess the three metrics described above. All questions were annotated with a 5-point Likert scale, were 1 represented *Strongly Disagree* and 5 is *Strongly Agree*.

4.2 Results

Automatic Evaluation. The aim of this automated evaluation was to measure the complexity of the generated content by the model. To automatically analyze the generated content, we introduced the concept of an artifact, encompassing beliefs, desires, conditions, and all other scenario elements. The core BDI components and emotional appraisal counts are presented in Table 1, while Table 2 displays the counts related to dialogue generation. In a recent study Guimarães et al. asked 10 participants with previous experience working with the FAtiMA-Toolkit to generate two scenarios based on two distinct stories from RocStories. Results indicated participants generated an average of 1.5 artifacts per minute[12]. As shown in Table 1, our model generates an average of 485.81 artifacts per scenario, translating to approximately 15.28 artifacts per minute given the average generation time for a scenario. This demonstrates the model's ability to produce more content in less time.

The dialogue state machines in the generated dialogues contain an average of 5.30 utterances, which is relatively limited (see Table 2). This suggests that improvements to our dialogue generation prompt are necessary. Additionally, the higher number of speak actions per scenario compared to dialogue lines indicates that multiple agents may use the same dialogue line, which may be undesirable.

Finally, we evaluated the feasibility of the action plans generated by the model for the agents. We identified the number of initial actions that could be performed immediately, i.e., those with conditions met by the agents' existing beliefs and desires, without requiring any changes. Emotions were not considered in this evaluation. In total, 955 actions were deemed immediately executable without updating the agents' knowledge bases. This highlights the model's ability to incorporate existing beliefs and desires into action conditions. However, a closer examination of the action plans' overall feasibility revealed that only 11 of the 369 generated intentions could be fully achieved (i.e., the action plans could be executed to completion). This discrepancy may stem from inaccuracies in the model's creation of conditions (e.g., adding extraneous conditions not present in the knowledge base) or in its generation of effects (e.g., the effects failing to properly update or introduce new beliefs and/or desires in the agents' knowledge base).

Subjective Evaluation. The automatic evaluation indicates the prompting approach generates a substantial amount of content, but there is no guarantee the content is relevant, coherent, or of high quality. The subjective evaluation is intended to assess these factors and provide answers accordingly. There are 3 salient topics in the analysis of the annotations: *relevance* of the created artefacts, *ramification* of the scenarios, and *errors in the prompting sequence*.

Regarding relevance, the model was able to correctly generate relevant agents (M=4.90, SD=0.316), beliefs/desires (M=3.60, SD=0.1994), and was less successfully at generating relevant intentions (M=3.00, SD=2.108). The model was clearly better at extracting agents. The task of generating correct and relevant beliefs, desires, and intentions requires the model to extract more implicit information which is an extra reasoning step that makes the generation harder. The model produced relevant actions (M=4.20, SD=0.632) in line with the given scenario. Additionally, the model was able to generate plausible emotions for each agent (M=3.50, SD=1.354). The model's ability to store human behavioural patterns and action planning in its weights without example prompts is apparent. Nevertheless, it struggled with generating conditions and effects (M=2.90, SD=2.025), which require to access beliefs and actions previously generated and dialogues (M=3.00, SD=1.704) that that into account goals and intentions. Lower performance on these steps is expected and requires additional effort. Most dialogues were too concise and broad, which is suboptimal.

To evaluate ramification, we asked annotators if the model added extra details they would not thought of but would still maintain to make the scenario richer, given that the RocStories scenarios were specific about the actions and events. The model frequently added more agents than the annotators would (M=4.10, SD=1.287). For example, when a hospital was mentioned the model took the liberty to add a doctor and nurses. More beliefs, desires (M=3.60, SD=1.147) and intentions (M=3.00, SD=1.944) were also added. Extra actions (M=3.00, SD=1.944), emotional labels (M=2.70, SD=1.252) and conditions/effects (M=2.90, SD=2.025) were added less frequently.

[11] Multiple different questions were asked for each step in the prompting pipeline: agents, beliefs, desires, intentions, plan generation, actions conditions and effects, dialogue and emotions

[12] The study was conducted in the context of evaluating the latest version of the FAtiMA Toolkit

	Dialogue Lines	Speak Actions	Speak Action's Conditions	Speak Action's Effects
Absolute	228	435	1508	902
Average	5.30	10.12	35.07	20.98

Table 2: Absolute and mean counts per scenario of artefacts related to dialogue generation. A total of 43 scenarios were generated.

When analyzing errors, the model's action plans were considered plausible, but the attributed conditions and effects were often incorrect. Additionally, we observed that errors later in the theoretical Chain of Thought (CoT) led to better quality scenarios. This is because errors at the beginning of the generation process propagate through the pipeline and affect subsequent steps.

5 DISCUSSION

SIAs are expected to demonstrate cognitive ability, social behavior, emotional expression, personality, and have mental representations in a variety of applications [33]. Each behavior and action displayed by the agents must be consistent with their internal states and as such, writing scenarios can be complex and time-consuming.

LLMs seem very appellative to create SIAs automatically, but also have several limitations. They suffer from *repetition* [42], *generate false information* that can be irrelevant or incoherent with the context [42] and have trouble maintaining a consistent persona over long-term interactions [47, 51]. Furthermore, LLMs have *low interpretability* [52] and limited control, and are *hard to interpret and debug* [52]. Even recent models (e.g., GPT-4 [34]) still suffer from these problems. These characteristics may make LLMs an unattractive technology to create rich social experiences in safety-critical systems (e.g., mental health and healthcare) or applications with tailored user experiences.

We have demonstrated, however, that despite their shortcomings, LLMs can rapidly generate vast quantities of content and possess significant amounts of commonsense knowledge acquired from extensive training data. The theory-driven approach proved to be useful in extracting agents and knowledge rooted on beliefs desires and intentions. We verified that the CoT prompting combined with the theoretical concepts forced the system to reason about the necessary ingredients (beliefs and desires) to generate intentions, actions and emotions.

Overall the experiments presented in the previous section produced mixed results. While the model was successful in extracting agents, generating simple beliefs and desires, and forming basic intentions, it struggled with more complex reasoning tasks, such as generating action plans and updating the agent's knowledge base. This is something that can easily fixed by forcing the LLM to update the knowledge base at every step [13] During the subjective analysis of the generated scenarios, we observed that the difficulty of the LLM's task varied depending on factors such as the way the scenarios were written (e.g. verbal tense) and the topic of the story being described. Some scenarios facilitated the task while others made it more challenging for the LLM. Furthermore, because LLMs possess an innate propensity for *creativity*, the symbolic scenarios featured a greater number of agents and available actions than those depicted in the text. While it is a plausible reflection of the social context, it is uncertain whether these additions enhance the scenario, thus accounting for the low level of agreement among annotators. Generation of dialogues comes as a big limitation of this work (as described in the automated analysis), because the prompting strategy is generating a few too general dialogue lines.

Bickmore and Cassell [2] highlighted that social dialogue is a joint task with multiple functions, including initiation, termination, turn-taking, and feedback. For successful dialogue, agents require a memory of prior interactions, goals, and the ability to intentionally guide the conversation towards a desired state. LLMs lack these abilities, creating a barrier to human-agent social interactions. Nonetheless, our approach allows the LLM to generate dialogue lines based on a target emotion (e.g., Neutral, Proud, or Supportive) and intention (e.g., encouragement or greeting) in accordance with the scenario (see Figure 2).

6 CONCLUSION

This paper proposes a theory-based prompting approach for SIAs development, combining the BDI architecture, the OCC theory of emotion for emotional appraisal, FAtiMA toolkit, and a LLMs' generative capabilities. This strategy accelerates social scenario generation, while also providing transparency, interpretability and control to the developer of a scenario by allowing its edition. However, using LLMs can introduce incoherences and errors that hurt the quality of the generated scenarios. We suggest integrating a human-in-the-loop at every stage of the generation process is crucial for error minimization and oversight. This approach combines human creativity and LLM generative capabilities, potentially yielding more engaging and robust social scenarios.

Serving as a proof-of-concept for the SIA field, this work illustrates how LLMs can speed up social scenario creation, despite the occasional inconsistencies introduced by these models. The potential to produce 10 times more artifacts than human users is a significant boost in generative capacity. However, the focus must remain on content quality.

In our subjective evaluation, we found that, although the models had some inconsistencies, they were reasonably proficient at content generation. We didn't evaluate the resulting social interaction quality from these scenarios, but future work will assess this aspect using typical interaction metrics such as engagement, rapport, user satisfaction, and user experience.

ACKNOWLEDGMENTS

This study received Portuguese national funds from FCT - Foundation for Science and Technology through the PhD grant 2021/06419/BD, and projects UIDB/50021/2020, UIDB/04326/2020, SLICE PTDC/CCI-COM/30787/2017, IDP/04326/2020, and LA/P/0101/2020.

[13] Using GPT-4, in the online interface, this was easily achievable because there is a much lesser strict limit for the input size (32.768 tokens). The same is not true for GPT-3.5 using the API (4,096 tokens).

REFERENCES

[1] Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O Riedl. 2021. Automated storytelling via causal, commonsense plot ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5859–5867.

[2] Timothy Bickmore and Justine Cassell. 2005. Social dialongue with embodied conversational agents. In *Advances in natural multimodal dialogue systems*. Springer, 23–54.

[3] Kim Bosman, Tibor Bosse, and Daniel Formolo. 2018. Virtual agents for professional social skills training: An overview of the state-of-the-art. In *International Conference on Intelligent Technologies for Interactive Entertainment*. 75–84.

[4] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317* (2019).

[5] Michael Bratman. 1987. Intention, plans, and practical reason. (1987).

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Diogo S Carvalho, Joana Campos, Manuel Guimarães, Ana Antunes, João Dias, and Pedro A Santos. 2021. CHARET: Character-centered Approach to Emotion Tracking in Stories. *arXiv preprint arXiv:2102.07537* (2021).

[8] YH Chan. 2003. Biostatistics 104: correlational analysis. *Singapore Med J* 44, 12 (2003), 614–619.

[9] Jack-Antoine Charles, Caroline PC Chanel, Corentin Chauffaut, Pascal Chauvin, and Nicolas Drougard. 2018. Human-agent interaction model learning based on crowdsourcing. In *Proceedings of the 6th International Conference on Human-Agent Interaction*. 20–28.

[10] Klaus Christoffersen and David D Woods. 2002. How to make automated systems team players. *Advances in human performance and cognitive engineering research* 2 (2002), 1–12.

[11] Philip R Cohen and Hector J Levesque. 1990. Intention is choice with commitment. *Artificial intelligence* 42, 2-3 (1990), 213–261.

[12] Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active Prompting with Chain-of-Thought for Large Language Models. arXiv:2302.12246 [cs.CL]

[13] Joao Dias and Ana Paiva. 2005. Feeling and reasoning: A computational model for emotional characters. In *Portuguese conference on artificial intelligence*. Springer, 127–140.

[14] Guglielmo Gemignani, Emanuele Bastianelli, and Daniele Nardi. 2015. Teaching robots parametrized executable plans through spoken interaction. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 851–859.

[15] J. Gratch and S. Marsella. 2004. A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Systems Research* 5, 4 (2004), 269–306.

[16] Manuel Guimarães, Joana Campos, Pedro A Santos, João Dias, and Rui Prada. 2022. Towards Explainable Social Agent Authoring tools: A case study on FAtiMA-Toolkit. *arXiv preprint arXiv:2206.03360* (2022).

[17] Manuel Guimarães, Rui Prada, Pedro Santos, João Dias, Cristina Soeiro, Raquel Guerra, Christina Steiner-Stanitznig, and Andrea Molinari. 2022. ISPO: A Serious Game to train the Interview Skills of Police Officers. *International Journal of Serious Games* 9 (11 2022), 43–61. https://doi.org/10.17083/ijsg.v9i4.514

[18] Arno Hartholt, David Traum, Stacy C. Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*. Edinburgh, UK.

[19] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models. arXiv:2207.05608 [cs.RO]

[20] Sepehr Janghorbani, Ashutosh Modi, Jakob Buhmann, and Mubbasir Kapadia. 2019. Domain Authoring Assistant for Intelligent Virtual Agents. *arXiv preprint arXiv:1904.03266* (2019).

[21] W Lewis Johnson and James C Lester. 2016. Face-to-face interaction with pedagogical agents, twenty years later. *International Journal of Artificial Intelligence in Education* 26, 1 (2016), 25–36.

[22] John E. Laird. 2012. *The SOAR Cognitive Architecture*. The MIT Press.

[23] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *CoRR* abs/2104.08691 (2021). arXiv:2104.08691 https://arxiv.org/abs/2104.08691

[24] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *CoRR* abs/2101.00190 (2021). arXiv:2101.00190 https://arxiv.org/abs/2101.00190

[25] Alan Lindsay, Jonathon Read, Joao F Ferreira, Thomas Hayton, Julie Porteous, and Peter Gregory. 2017. Framer: Planning models from natural language action descriptions. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

[26] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* 55, 9, Article 195 (jan 2023), 35 pages. https://doi.org/10.1145/3560815

[27] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. *CoRR* abs/2103.10385 (2021). arXiv:2103.10385 https://arxiv.org/abs/2103.10385

[28] S. Marsella and J. Gratch. 2006. EMA: a computational model of appraisal dynamics. In *European Meeting on Cybernetics and Systems Research*.

[29] Stacy C Marsella and Jonathan Gratch. 2009. EMA: A process model of appraisal dynamics. *Cognitive Systems Research* 10, 1 (2009), 70–90.

[30] Samuel Mascarenhas, Manuel Guimarães, Rui Prada, João Dias, Pedro A Santos, Kam Star, Ben Hirsh, Ellis Spice, and Rob Kommeren. 2018. A virtual agent toolkit for serious games developers. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–7.

[31] Samuel Mascarenhas, Manuel Guimarães, Pedro A Santos, João Dias, Rui Prada, and Ana Paiva. 2021. FAtiMA Toolkit–Toward an effective and accessible tool for the development of intelligent virtual agents and social robots. *arXiv preprint arXiv:2103.03020* (2021).

[32] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 839–849. https://doi.org/10.18653/v1/N16-1098

[33] Eric Nichols, Leo Gao, Yurii Vasylkiv, and Randy Gomez. 2021. Collaborative Storytelling with Social Robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1903–1910.

[34] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[35] Andrew Ortony, Gerald Clore, and Allan Collins. 1988. *The Cognitive Structure of Emotion*. Vol. 18. https://doi.org/10.2307/2074241

[36] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442 [cs.HC]

[37] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. arXiv:2208.04024 [cs.HC]

[38] Florian Pecune, Angelo Cafaro, Magalie Ochs, and Catherine Pelachaud. 2016. Evaluating social attitudes of a virtual tutor. In *International Conference on Intelligent Virtual Agents*. Springer, 245–255.

[39] Gonçalo Pereira, António Brisson, João Dias, André Carvalho, Joana Dimas, Samuel Mascarenhas, Joana Campos, Marco Vala, Iolanda Leite, Carlos Martinho, et al. 2014. Non-Player Characters and Artificial Intelligence. In *Psychology, Pedagogy, and Assessment in Serious Games*. IGI Global, 127–152.

[40] Vittorio Perera and Manuela Veloso. 2014. Task Based Dialog for Service Mobile Robot. In *2014 AAAI Fall Symposium Series*.

[41] Alexandru Popescu, Joost Broekens, and Maarten Van Someren. 2014. Gamygdala: An emotion engine for games. *IEEE Transactions on Affective Computing* 5, 1 (2014), 32–44.

[42] Shrimai Prabhumoye, Chris Quirk, and Michel Galley. 2019. Towards content transfer through grounded text generation. *arXiv preprint arXiv:1905.05293* (2019).

[43] Margherita Rampioni, Vera Stara, Elisa Felici, Lorena Rossi, and Susy Paolini. 2021. Embodied conversational agents for patients with dementia: thematic literature analysis. *JMIR mHealth and uHealth* 9, 7 (2021), e25381.

[44] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 314, 7 pages. https://doi.org/10.1145/3411763.3451760

[45] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic Prompting: Generating Chain-of-Thought Demonstrations for Large Language Models. arXiv:2302.00618 [cs.CL]

[46] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4222–4235. https://doi.org/10.18653/v1/2020.emnlp-main.346

[47] Feng-Guang Su, Aliyah R Hsu, Yi-Lin Tuan, and Hung-Yi Lee. 2019. Personalized Dialogue Response Generation Learned from Monologues.. In *INTERSPEECH*. 4160–4164.

[48] Chen Feng Tsai, Xiaochen Zhou, Sierra S Liu, Jing Li, Mo Yu, and Hongyuan Mei. 2023. Can Large Language Models Play Text Games Well? Current State-of-the-Art and Open Questions. *arXiv preprint arXiv:2304.02868* (2023).

[49] Xinyi Wang, Samuel S Sohn, and Mubbasir Kapadia. 2019. Towards a conversational interface for authoring intelligent virtual characters. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. 127–129.

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *CoRR* abs/2201.11903 (2022). arXiv:2201.11903 https://arxiv.org/abs/2201.11903

[51] Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567* (2021).

[52] Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. 2021. A Survey of Knowledge Enhanced Pre-trained Models. *arXiv preprint arXiv:2110.00269* (2021).

[53] Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. In-Context Instruction Learning. arXiv:2302.14691 [cs.CL]

[54] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493 [cs.CL]

[55] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-Shot Performance of Language Models. arXiv:2102.09690 [cs.CL]

[56] Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable Generation from Pre-trained Language Models via Inverse Prompting. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021).