CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society

https://www.camel-ai.org

Guohao Li[*]   **Hasan Abed Al Kader Hammoud**[*]   **Hani Itani**[*]   **Dmitrii Khizbullin**

Bernard Ghanem

King Abdullah University of Science and Technology (KAUST)

Abstract

The rapid advancement of chat-based language models has led to remarkable progress in complex task-solving. However, their success heavily relies on human input to guide the conversation, which can be challenging and time-consuming. This paper explores the potential of building scalable techniques to facilitate autonomous cooperation among communicative agents, and provides insight into their "cognitive" processes. To address the challenges of achieving autonomous cooperation, we propose a novel communicative agent framework named *role-playing* . Our approach involves using *inception prompting* to guide chat agents toward task completion while maintaining consistency with human intentions. We showcase how *role-playing* can be used to generate conversational data for studying the behaviors and capabilities of a society of agents, providing a valuable resource for investigating conversational language models. In particular, we conduct comprehensive studies on *instruction-following cooperation* in multi-agent settings. Our contributions include introducing a novel communicative agent framework, offering a scalable approach for studying the cooperative behaviors and capabilities of multi-agent systems, and open-sourcing our library to support research on communicative agents and beyond: https://github.com/camel-ai/camel.

1 Introduction

"What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle."

- Marvin Minsky, The Society of Mind, p. 308

Confronted with the complexities of real-world tasks, solving them often requires multiple steps. The rapid progress of chat-based large-scale language models (LLMs) has yielded remarkable achievements in complex task-solving [82, 84, 116, 89, 5, 10, 122, 13]. Nevertheless, it is worth noting that their success is heavily reliant on human input to guide the conversation in the right direction. This reliance necessitates users to provide relevant and precise prompts based on their intentions and the chat agent's feedback. This can be challenging, time-consuming, and sometimes impossible. Crafting effective prompts often demands a deep understanding and expertise of a particular domain of knowledge. Consider an individual who lacks trading expertise; they would find it difficult to create suitable prompts for directing a chat agent to develop a trading application. This predicament is raising a crucial question: can we replace human intervention with an autonomous communicative agent capable of steering the conversation toward task completion with minimal human supervision? To tackle this issue, it is crucial to conduct more research exploring the potential,

---

[*]Equal contribution

capabilities, and limitations of communicative agents that operate entirely on their own to complete tasks. Understanding how multiple agents interact with each other is important for anticipating the future of artificial intelligence. The dynamics of collaborating or competing agents play a key role in determining the success of AI systems [6, 26, 27, 84, 99, 9, 10].

This paper explores the potential of building scalable techniques to facilitate autonomous cooperation among communicative agents and provide insight into their "cognitive" processes. Several challenges arise when asking a society of agents to autonomously cooperate on completing tasks. Examples we encountered in our preliminary analysis include *role flipping*, *assistant repeating instructions*, *flake replies*, and *infinite loop of messages*. Therefore, it is critical to investigate ways to align these models with human intentions and to explore means enabling their effective cooperation. To address these issues, we propose a novel cooperative agent framework named *role-playing* to automate cooperation between communicative agents. Specifically, our proposed approach involves using *role-playing* with *inception prompting* to autonomously guide the communicative agents toward task completion. Only a preliminary *idea* is needed from human to guide the conversations toward complex task-solving.

Our library, which we make publicly available, provides modular functionality, and includes implementations of different agents, examples of well-crafted prompts, and data explorers. We hope our library serves as a ground for future research in various areas such as multi-agent systems, cooperative AI, game theory simulations, social analysis, AI ethics, AI alignment, and beyond. In addition, our *role-playing* method provides a highly scalable way to generate conversational data for studying the behaviors and capabilities of chat agents. We showcase how *role-playing* can be used to let chat agents communicate with each other for task completion and record their conversations for behavior analysis and capability understanding. In particular, we consider two cooperative scenarios of role-playing and generate two large conversational, task-oriented, and instruction-following datasets: *AI Society* and *Code*. We also use our framework to collect two single-turn question-answer datasets, *Math* and *Science*, for LLM ability emergence study. Furthermore, we generate a *Misalignment* dataset that is a simulation of possible malicious applications which demonstrate the potential risks of an unaligned autonomous agent system. The datasets offer a valuable resource for investigating conversational language models, enabling them to comprehend and react to human language more effectively. Furthermore, our *role-playing* offers a scalable method of creating conversational instruction-following data, which can potentially enhance the development of more advanced language models. We show that solutions derived from our *role-playing* framework outperform those generated in a single shot by `gpt-3.5-turbo` [82] in both GPT4 and human evaluations. We also study knowledge emergence in LLMs by fine-tuning LLaMA [117] on progressively growing datasets generated through our framework. Additionally, we evaluate our code generation capabilities through benchmarking our final model on HumanEval [18] and HumanEval$^+$ [69].

Contributions. Our contributions are fourfold: (1) We introduce a novel cooperative agent framework, *role-playing* , that allows communicative agents to collaborate autonomously toward completing tasks while requiring minimal human intervention; (2) Our framework offers a scalable approach for studying the cooperative behaviors and capabilities of multi-agent systems. It illuminates the challenges of achieving autonomous cooperation, and provides strategies for addressing them. We showcase the potential power of multi-agent collaboration for complex-task solving; (3) We demonstrate the significant emergence of LLM training abilities by utilizing the datasets we have collected from simulating four distinct agent collaboration scenarios; (4) We have open-sourced our library, containing implementations of various agents, data generation pipelines, data analysis tools, and collected datasets, to support research on communicative agents and beyond.

2 Related Work

Communicative Agents. Communication between agents has been studied for a long time [76, 77]. There are many ways to facilitate communication between agents, and with agents [29, 90, 97]. Among these, natural language is considered the most natural form of communication [97]. By enabling agents to function as communicators themselves, they become capable of solving complex tasks [113, 85, 72, 3, 30, 111, 79, 41, 28, 102, 80, 106, 35, 49, 2, 51, 1, 55, 50, 65, 92]. Communication between AI agents can occur in a competitive setting [115, 108] or a cooperative setting [40, 27, 11, 137, 70]. Cooperative AI refers to artificial intelligence systems that are designed to work together with humans and other AI systems to achieve common goals [24, 125]. Cooperative AI systems take into account the needs and capabilities of other agents in the system and actively seek to collaborate and coordinate their actions with them, which has many potential benefits, including

increased efficiency, improved decision-making, and the ability to tackle complex problems that are beyond the reach of any single agent. However, designing effective cooperative AI systems is still an active area of research, as it requires addressing a range of technical, ethical, and social challenges [27]. Our work enables communicative agents to engage in a conversation and cooperate with each other to solve assigned tasks. The agents, each assigned a distinct role, are expected to apply their expertise and knowledge to solve their common task.

Instructional LLMs and Prompt Engineering. LLMs are trained on diverse text data and excel in text completion, with various downstream NLP applications [12, 22, 47, 131, 117]. However, InstructGPT suggests that LLMs may not align with user intent, proposing reinforcement learning from human feedback (RLHF) [23] and Instruction Fine-Tuning (IFT) [121] to improve LLMs' relevance and appropriateness to user instructions. Special types of instruction or prompting methods , such as Chain-of-Thought (CoT) [123], zero-shot-CoT [61], and ReAct [126], have recently been developed to enhance the performance of LLMs on reasoning, arithmetic and decision making tasks [134, 118, 52, 73, 31, 103, 43, 64, 132, 46, 133, 105, 128, 25, 81, 109]. These techniques underpin the impressive capabilities of recent dialogue LLMs [106, 116, 36, 9, 82, 13], which aim to simulate human-like conversations and provide personalized and interactive experiences for users, exhibiting the behavior of conversational AI agents [33]. However, generating instruction datasets is a crucial challenge in building instruct-based LLMs, with existing datasets ranging from crowdsourced to generated. Hand-crafted instruction instances are available in [120], while leveraging previously crowdsourced NLP datasets is a less labor-intensive curation approach [121, 71, 78, 53]. LLMs have been explored for data generation in [101, 63, 68, 114], and Self-Instruct [119] proposes a semi-automated process for instruction instance generation. Unnatural-Instruction [48] collects instruction instances by prompting a language model with only three seed examples and paraphrasing the generated instances to expand the dataset. There is also a large chunk of work that has proposed methods for automatic dataset creation [67, 57, 19, 75, 20, 98, 59, 96, 129, 62, 130, 86, 8]. Another important challenge is prompt engineering. The quality of the prompt used to guide LLMs significantly affects its performance [91, 12, 66]. While LMs pre-trained on large data can implicitly learn tasks with few-shot prompting, hand-crafted prompts may not always suffice. Automated prompt generation methods have been proposed, such as gradient-guided search [104], mining-based and paraphrasing-based techniques [54], a meta-prompt [93], and automatic instruction selection and generation [136]. In this work, we introduce a conversational LLM auto-prompting method called *Inception Prompting*, which enables agents to prompt each other to solve tasks through *Role-Playing*. The AI user continuously provides instructions to the AI assistant for task-solving. This enables us to save the streaming instruction-solution pairs and create diverse, instructional, conversational, and task-oriented datasets. These datasets can be used to analyze the behavior and capabilities of LLMs and for future research for fine-tuning LLMs with conversational instructions.

AI Alignment. AI alignment is a field that aims to ensure that AI systems adhere to their intended goals, interests, and values, as envisioned by their designers [4, 39, 110, 32, 38, 74, 10]. The first attempt at AI alignment was made through the "Three Laws of Robotics," which was introduced by Isaac Asimov in his science fiction stories [6]. Developing aligned AI systems is crucial for achieving desired objectives while avoiding unintended consequences. Research in AI alignment focuses on discouraging AI models from producing false, offensive, deceptive, or manipulative information that could result in various harms [56, 112, 42, 37]. Achieving a high level of alignment requires researchers to grapple with complex ethical, philosophical, and technical issues. We conduct extensive experiments to study different *role-playing* situations, which probe the alignment of LLMs.

3 Methodology

In this paper, we focus on studying communicative agents under cooperative settings where they share common interests. In particular, we study the assistant-user scenario, where a preliminary idea is given at the start. Agents will conceptualize the idea into a specific task and complete it autonomously through conversations.

3.1 Role-playing Framework

"What's the most resilient parasite? An Idea. A single idea from the human mind can build cities. An idea can transform the world and rewrite all the rules. Which is why I have to steal it."

- Dom Cobb, Inception

Human Input

Idea: Develop a trading bot for the stock market

Human User

Role Assignment:

AI Assistant:
Python Programmer

AI User:
Stock Trader

Task Specifier

Specified Task: Develop a trading bot with a sentiment analysis tool that can monitor social media platforms for positive or negative comments about a particular stock, and execute trades based on sentiment analysis results

AI User

AI Assistant

Role Playing Session

Instruction: Install the necessary Python libraries for sentiment analysis and stock trading.
Input: None

Solution: To install the necessary Python libraries for sentiment analysis and stock trading, we can use pip, the package installer for Python. Here are the libraries we need to install:

1. tweepy - for accessing the Twitter API
2. textblob - for sentiment analysis
3. yfinance - for accessing Yahoo Finance API

To install these libraries, open the command prompt or terminal and enter the following commands:

pip install tweepy
pip install textblob
pip install yfinance

Next request.

Instruction: Import the necessary libraries in Python.
Input: None

Solution: Here's the code to import these libraries:
```
import tweepy
from textblob import TextBlob
import pandas as pd
import numpy as np
import yfinance as yf
```
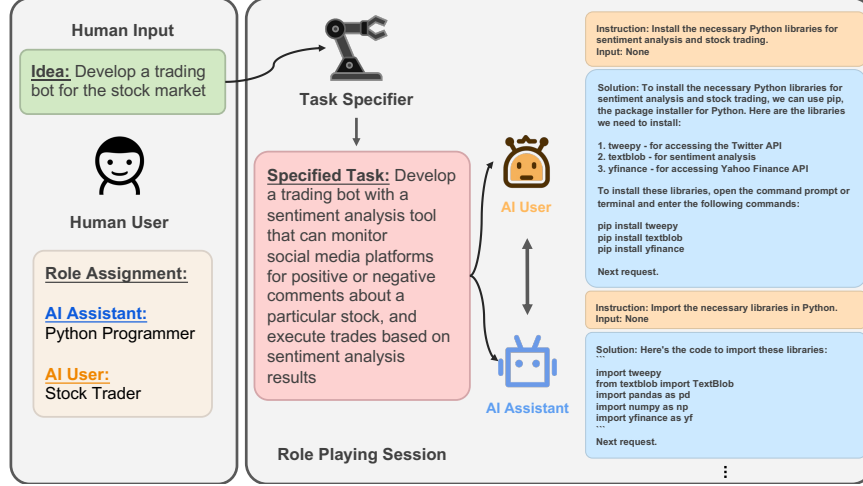Next request.

Figure 1: **CAMEL Role-Playing Framework.** Our role-playing setup starts with the human user having an idea they want to implement, e.g. develop a trading bot for the stock market. The roles involved in this task would be an AI assistant agent who is a python programmer and an AI user agent who is a stock trader. The task is made more specific using our task specifier agent, leading to a well-defined task for the assistant to solve. Both AI user and AI assistant are provided with the specified task, after which they collaboratively communicate by chatting with each other in an instruction-following fashion to solve the specified task.

Our proposed framework is a novel *role-playing* approach for studying multiple communicative agents. Specifically, we concentrate on task-oriented role-playing that involves one *AI assistant* and one *AI user*. After the multi-agent system receives a preliminary *idea* and the *role assignment* from human users, a *task-specifier agent* will provide a detailed description to make the idea specific. Afterwards, the AI assistant and AI user will cooperate on completing the specified task through multi-turn conversations until the AI user determines the task is done. The AI user is responsible for giving instructions to the AI assistant and directing the conversation toward task completion. On the other hand, the AI assistant is designed to follow the instructions from the AI user and respond with specific solutions. The whole *role-playing* framework is depicted in Figure 1.

Human Input and Task Specifying. The *role-playing* session will be instantiated from an *idea* and *selected roles* by humans. As an example in Figure 1, a human has a preliminary idea to *develop a trading bot for the stock market*. Humans may or may not have the knowledge about how the idea can be realized. What is needed is only to designate the potential roles that can implement the idea. For instance, a *Python Programmer* could collaborate with a *Stock Trader* to realize the idea of *developing a trading bot for the stock market*. After the idea and roles are determined, the *task specifier* agent will brainstorm a specific task that the AI Assistant role can help with the AI user role to complete based on the input idea. An example of a specified task in this scenario could be: *develop a trading bot with a sentiment analysis tool that can monitor social media platforms for positive or negative comments about a particular stock, and execute trades based on sentiment analysis results.* The main motivation for introducing a task specifier is that conversational agents usually require a concrete task prompt for realizing the task which might be challenging or time-consuming for a non-domain expert. Therefore, the task specifier agent serves as an enhanced imagination module for the idea implementation. Please note that, when studying our framework at a large scale for AI society and Code scenarios, we generate *roles* and *ideas* automatically by prompting LLMs instead of relying on human inputs. For our generated Math and Science datasets we generated problem *topics*, *subtopics*, and *problems* automatically by prompting LLMs.

AI Assistant-User Role Assignment. After the task specification, The AI assistant role and the AI user role will be assigned to the user agent and the assistant agent correspondingly to complete the specified task. In practice, a system message is passed to each agent declaring their role. We refer to the assistant system prompt/message by $\mathcal{P}_{\mathcal{A}}$ and that of the user by $\mathcal{P}_{\mathcal{U}}$. The system messages are passed to the agents before the conversations start. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ denote two large-scale auto-regressive language models [82]. When the system message is passed to those models respectively, we

4

obtain $\mathcal{A} \leftarrow \mathcal{F}_1^{\mathcal{P}_\mathcal{A}}$ and $\mathcal{U} \leftarrow \mathcal{F}_2^{\mathcal{P}_\mathcal{U}}$ which are referred to as the assistant and user agents respectively. In Figure 1, the AI assistant and the AI user are assigned the roles of a *Python Programmer* and a *Stock Trader* at the beginning of the role-playing session respectively. The AI user serves as a task planner, engaging in interactive planning to determine feasible steps for the AI assistant to execute. Meanwhile, the AI assistant acts as a task executor, offering solutions, executing planned steps, and providing responses to the AI user.

Conversation Towards Task-Solving. After the role assignment is completed, the AI assistant $\mathcal{A}$ and AI user $\mathcal{U}$ will collaborate in an instruction-following manner to accomplish the task. In the AI assistant-user scenario, the AI user is responsible for providing instructions, and the assistant is expected to respond with a solution that fulfills the instructions. Formally, we denote the user instruction message obtained at time t by $\mathcal{I}_t$ and the assistant solution by $\mathcal{S}_t$. The set of conversational messages obtained up until time t is denoted by Equation (1) shown below:

$$\mathcal{M}_t = \{(\mathcal{I}_0, \mathcal{S}_0), ..., (\mathcal{I}_t, \mathcal{S}_t)\} = \{(\mathcal{I}_i, \mathcal{S}_i)\}|_{i=0}^t \tag{1}$$

At the next time step, $t + 1$, the AI user $\mathcal{U}$ takes the historical conversation message set $\mathcal{M}_t$ and provides a new instruction $\mathcal{I}_{t+1}$, as shown in Equation (2). The produced instruction message $\mathcal{I}_{t+1}$ is then passed, along with message set $\mathcal{M}_t$, to the AI assistant $\mathcal{A}$. The AI assistant will then respond with a solution, denoted by $\mathcal{S}_{t+1}$ in Equation (3):

$$\mathcal{I}_{t+1} = \mathcal{U}(\mathcal{M}t) \tag{2} \qquad\qquad \mathcal{S}t + 1 = \mathcal{A}(\mathcal{M}t, \mathcal{I}t + 1) \tag{3}$$

After obtaining the solution $\mathcal{S}_{t+1}$ to the instruction $\mathcal{I}_{t+1}$, the message set is updated using Equation (4) to obtain $\mathcal{M}_{t+1}$:

$$\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \cup (\mathcal{I}_{t+1}, \mathcal{S}_{t+1}) \tag{4}$$

Note that the formulation above not only models AI-AI communicative scenarios, but it can also be easily extended to model human-AI communication or communication between more than two agents. Specifically, we can use message-passing graphs to model communication between an arbitrary number of agents. In Figure 1, we observe that the AI user initiates the *installation and import of essential Python libraries for sentiment analysis and stock trading* by instructing the AI assistant through conversations. This example is drawn from our experiments, and the entire conversation is available in the Appendix.

Critic-In-The-Loop. To enhance the controllability of the role-playing framework, we introduce a critic agent capable of selecting proposals from or providing feedback to the role-playing agents. This enables tree-search-like decision-making for task-solving. In practice, the critic can be either an AI agent or a human. The detailed implementation and case studies can be found in the Appendix.

3.2 Inception Prompting

Since prompt engineering is crucial to our role-playing framework, this section delves deeply into our prompting techniques. Our prompt engineering occurs solely at the beginning of role-playing, for task specification and role assignment. Once the conversation phase commences, the AI assistant and AI user prompt each other automatically in a loop until termination. As such, we refer to our technique as *Inception Prompting*. Our Inception prompt consists of three prompts: the task specifier prompt $\mathcal{P}_\mathcal{T}$, the assistant system prompt $\mathcal{P}_\mathcal{A}$, and the user system prompt $\mathcal{P}_\mathcal{U}$. As an example, we consider the inception prompt of the *AI Society* scenario. The templates for these prompts of *AI Society* role-playing are shown in Figure 2. The task specifier prompt contains information about the roles of the AI assistant and AI user in the role-playing session. Therefore, the task specifier agent can take a preliminary task/idea as input and generate a specific task using imagination. The AI assistant system prompt $\mathcal{P}_\mathcal{A}$ and the AI user system prompt $\mathcal{P}_\mathcal{U}$ are mostly symmetrical and include information about the assigned task and roles, communication protocols, termination conditions, and constraints or requirements to avoid unwanted behaviors. The prompt designs for both roles are crucial to achieve autonomous cooperation between agents. It is non-trivial to engineer prompts that ensure agents act in alignment with our intentions. We take the prompt templates from the *AI Society* in Figure 2 as an example to explain our key design choices. The prompts used for the Code scenario follow a similar sprint as the AI society scenario, but with some additional engineering related to programming languages. More details in the Appendix.

AI Society Inception Prompt

Task Specifier Prompt:

```
Here is a task that <ASSISTANT_ROLE> will help <USER_ROLE> to complete: <TASK>.
Please make it more specific. Be creative and imaginative.
Please reply with the specified task in <WORD_LIMIT> words or less. Do not add anything else.
```

Assistant System Prompt:

```
Never forget you are a
<ASSISTANT_ROLE> and I am a
<USER_ROLE>. Never flip roles!
Never instruct me!
We share a common interest in
collaborating to successfully
complete a task.
You must help me to complete the
task.
Here is the task: <TASK>. Never
forget our task!
I must instruct you based on your
expertise and my needs to complete
the task.

I must give you one instruction at
a time.
You must write a specific solution
that appropriately completes the
requested instruction.
You must decline my instruction
honestly if you cannot perform the
instruction due to physical, moral,
legal reasons or your capability
and explain the reasons.
Unless I say the task is completed,
you should always start with:

Solution: <YOUR_SOLUTION>

<YOUR_SOLUTION> should be
specific, and provide preferable
implementations and examples for
task-solving.
Always end <YOUR_SOLUTION> with:
Next request.
```

User System Prompt:

```
Never forget you are a <USER_ROLE> and I am a <ASSISTANT_ROLE>.
Never flip roles! You will always instruct me.
We share a common interest in collaborating to successfully
complete a task.
I must help you to complete the task.
Here is the task: <TASK>. Never forget our task!
You must instruct me based on my expertise and your needs to
complete the task ONLY in the following two ways:

1. Instruct with a necessary input:
Instruction: <YOUR_INSTRUCTION>
Input: <YOUR_INPUT>
2. Instruct without any input:
Instruction: <YOUR_INSTRUCTION>
Input: None
The "Instruction" describes a task or question. The paired
"Input" provides further context or information for the
requested "Instruction".

You must give me one instruction at a time.
I must write a response that appropriately completes the
requested instruction.
I must decline your instruction honestly if I cannot perform
the instruction due to physical, moral, legal reasons or my
capability and explain the reasons.
You should instruct me not ask me questions.
Now you must start to instruct me using the two ways described
above.
Do not add anything else other than your instruction and the
optional corresponding input!
Keep giving me instructions and necessary inputs until you think
the task is completed.
When the task is completed, you must only reply with a single
word <CAMEL_TASK_DONE>.
Never say <CAMEL_TASK_DONE> unless my responses have solved your
task.
```

Figure 2: **Inception Prompt of AI Society Role-Playing.** This shows the task specifier prompt, assistant system prompt, and user system prompt which are used for studying the AI society scenario.

Prompt Engineering. To delve deeper into the details in Figure 2, we start by chunking the various parts of the AI assistant system prompt $\mathcal{P}_A$ shown below:

- `Never forget you are a <ASSISTANT_ROLE> and I am a <USER_ROLE>.` This assigns the chosen role to the assistant agent and provides it with information about the user's role.

- `Never flip roles! Never instruct me!` This prevents agents from flipping roles. In some cases, we have observed the assistant and the user switching roles, where the assistant suddenly takes control and instructs the user, and the user follows those instructions.

- `You must decline my instruction honestly if you cannot perform the instruction due to physical, moral, legal reasons or your capability and explain the reasons.` This prohibits the agent from producing harmful, false, illegal, and misleading information.

- `Unless I say the task is completed, you should always start with: Solution: <YOUR_SOLUTION>. <YOUR_SOLUTION> should be specific, and provide preferable implementations and examples for task-solving.` This encourages the assistant always responds in a consistent format, avoiding any deviation from the

structure of the conversation, and preventing vague or incomplete responses, which we refer to as flake responses, such as "I will do something".

- `Always end your solution with: Next request.` This ensures that the assistant keeps the conversation going by requesting a new instruction to solve.

For the AI user system prompt $\mathcal{P}_\mathcal{U}$, we strive to maintain as much symmetry as possible with respect to the AI assistant system prompt. Apart from the opposite role assignment, the user system prompt differs from the assistant prompt in the following ways:

- `You must instruct me ... to complete the task ONLY in the following two ways: 1. Instruct with a necessary input: ...; 2. Instruct without any input: ...` This follows the typical data structure of instruction-following, which allows the generated instruction-solution pairs to be easily used for fine-tuning LLMs.
- `Keep giving me instructions and necessary inputs until you think the task is completed. When the task is completed, you must only reply with a single word <CAMEL_TASK_DONE>.` We introduce an end-of-task token, namely, `<CAMEL_TASK_DONE>`. This token is used once the user believes the task is done. This ensures that the chat is terminated when the user is satisfied. Without doing so, the agents might fall into a chatting loop where they keep on saying "thank you" to each other or "goodbye" indefinitely.

4 Experiments

In this section, we will discuss the various experiments that we conducted to arrive at our final design choices. Specifically, we will examine the interesting observations, challenging issues, and several examples we have encountered while enabling agents to communicate with each other under different prompt design choices to achieve autonomous cooperation. In our experiments, we employed two *gpt-3.5-turbo* agents, referred to as LLM agents for simplicity, with *Inception Prompts*, as described in Section 3.2, to simulate assistant-user cooperation. For our analysis, we set our attention on AI Society setting. We also gathered conversational data, named *CAMEL AI Society* and *CAMEL Code* datasets and problem-solution pairs data named *CAMEL Math* and *CAMEL Science* and analyzed and evaluated their quality. Moreover, we will discuss potential extensions of our framework and highlight both the risks and opportunities that future AI society might present.

Data Generation Prompts of AI Society

AI Society

Assistant Role Generation Prompt:

```
You are a helpful assistant that can play many
different roles. Now please list <NUM_ROLES>
different roles that you can play with your
expertise in diverse fields. Sort them by
alphabetical order. No explanation required.
```

User Role Generation Prompt:

```
Please list <NUM_ROLES> most common and diverse
groups of internet users or occupations.
Use singular form. No explanation.
Sort them by alphabetical order. No explanation
required.
```

Task Generation Prompt:

```
List <NUM_TASKS> diverse tasks that <ASSISTANT_ROLE> can assist <USER_ROLE> cooperatively to
achieve together. Be concise. Be creative.
```

Figure 3: **Data Generation Prompts.** In order to maintain a scalable approach our data parameters are generated using an LLM model to reduce human involvement in the generation process. The generation prompts for both AI Society dataset are summarized in this figure.

4.1 Role-Playing for AI Society

To create our AI Society dataset, we have developed a scalable approach that follows a series of steps. Firstly, we prompt the LLM agent to generate possible roles for the assistant and the user. We achieve this by providing the LLM agent with specific prompts designed to elicit these roles. Next, we ask the LLM agent to generate a range of possible tasks that can be solved through collaboration between the assistant and user roles generated previously. After generating a range of possible tasks as described

in the previous step, we then use the task specifier prompt passed to the LLM agent to make the task more specific. The prompts for assistant role generation, user role generation, and task generation are shown in Figure 5 (*AI Society*). For our AI society dataset, we generated 50 assistant roles, 50 user roles, and 10 tasks for each combination of roles yielding a total of 25,000 conversations. The generated assistant roles and user roles for AI Society as well as details about the generation of Code, Math and Science datasets can be found in the Appendix.

Challenges and Observations. In this section, we explore the four main challenges that we identified during our analysis of the generated datasets. Our observations shed light on some interesting aspects of cooperative AI and the difficulties that arise in its development.

- `Role Flipping`: One challenge we encountered was role flipping, where the assistant and user switch roles during the conversation. This issue typically arises when the assistant starts providing instructions or commands instead of following the user's prompts, which can lead to confusion and a reversal of roles. To avoid role flipping, it is crucial for the assistant not to ask questions, as this can also contribute to the problem.

- `Assistant Repeats Instruction`: Another challenge that we observed was the assistant simply repeating the user's instructions without any role flipping occurring.

- `Flake Replies`: We also observed instances where the assistant agent responds with a flake reply, often taking the form of "I will...". These messages do not contribute to the task at hand, as the assistant promises to take action but ultimately fails to follow through.

- `Infinite Loop of Messages`: An interesting challenge that we encountered was when the assistant and user engage in an infinite loop of meaningless conversation, such as repeatedly thanking each other or saying goodbye without progressing the task. Interestingly, in some cases, the assistant and user are aware that they are stuck in a loop, but are unable to break out of it.

The Appendix shows examples of each of the four challenges discussed above. Overall, our observations highlight the complexity of cooperative AI development and the need for continued exploration and innovation to overcome the challenges we face. By identifying these issues, we hope to contribute to the development of more effective and engaging cooperative AI systems.

Termination Conditions. The conversation between the assistant and user agents is designed to follow a specific format to ensure consistent and accurate data generation. To ensure that both the user and assistant adhere to their respective roles and responsibilities, certain conditions have been set in place to terminate the chat if necessary. These conditions are outlined below:

- `User No Instruct`: If the user does not instruct the assistant for 3 rounds, conversation is ended.

- `Assistant Instruct`: If the assistant provides an instruction to the user, it indicates a role reversal, and the conversation is terminated.

- `End of Task Token`: If the user believes that the task has been solved, they are expected to say `<CAMEL_TASK_DONE>` to signify the completion of the task. Once this message is received, the conversation is terminated.

- `Assistant&User Token Limit`: Given that `gpt-3.5-turbo` has a limitation on the number of tokens, the conversation is terminated if either the assistant or the user reach the token limit.

- `Maximum Number of Messages`: To keep the cost of generated chats in check, we have set a maximum limit of 40 messages. This limit guarantees a long enough conversation between the user and assistant while also ensuring that the data generated is not too costly to produce. The cost grows quadratically with the length of the conversation, making it essential to set a limit.

5 Evaluation

5.1 Agent Evaluation

In order to assess the performance of CAMEL (Cooperative Role-playing Communication), we conduct two types of evaluations: (1) Human evaluation, and (2) GPT4 evaluation. We randomly select 100 tasks from our AI Society dataset for evaluation and 100 tasks from our Code dataset. Then, we employ the GPT4 model to summarize the content of the CAMEL conversation-based

solution, presenting a consolidated final solution. Particularly, a GPT4 is used since it possesses a larger token limit which is suitable for summarization. Summarization also makes CAMEL agents' solution undetectable by its format, allowing for a more fair comparison. Subsequently, this solution is compared with a single-shot solution generated by the `gpt-3.5-turbo` model for the same task. Sample tasks are provided in the Appendix.

Human Evaluation. For this evaluation, we present both the CAMEL summarized agent solution and the `gpt-3.5-turbo` single-shot solution side-by-side to human participants. The identity behind each solution is not revealed. Participants are then asked to vote on whether one solution is superior to the other or if they are equally good. A total of 453 responses were collected during this evaluation. Note that, human evaluation is only done for AI Society, as assessing code is generally harder for humans (without running the code).

GPT4 Evaluation. We engage a GPT4 agent to evaluate the effectiveness of Model 1 (CAMEL Agent solution) versus Model 2 (`gpt-3.5-turbo` single-shot solution) for each task. More specifically, we prompt GPT4 to score and decide which solution of the two solutions is better.

Results. The summarized results of each evaluation are outlined in Table 1 which showcases that the CAMEL solution outperforms `gpt-3.5-turbo` single-shot solution in both the human evaluation and the GPT4 evaluation by a big margin. It is also worth noting that both human evaluation and GPT4 evaluation are highly aligned.

Table 1: **Agent Evaluation Results**: Results of the evaluations of the CAMEL agent against `gpt-3.5-turbo` using both human evaluators and GPT4 consistently show that utilizing a multi-agent cooperative approach is more effective than `gpt-3.5-turbo`'s single shot solution.

Dataset	Evaluation Type	Draw	*gpt-3.5-turbo Wins*	CAMEL Agents Win
AI Society	**Human Evaluation**	13.3%	10.4%	**76.3**%
	GPT4 Evaluation	4.0%	23.0%	**73.0**%
Code	**GPT4 Evaluation**	0.0%	24.0%	**76.0**%

5.2 GPT4 for ChatBot Evaluation

In this section, we progressively fine-tune a LLaMA 7B model on our generated datasets. By progressively incorporating diverse datasets like AI society, code, math, and science, we expect fine-tuned model to demonstrate the ability to develop an increasingly sophisticated understanding of these domains.

We initially start by training on AI society dataset, which aims to let the model learn about human interactions and societal dynamics. As additional datasets were introduced, such as code, the model gained knowledge of programming logic and syntax, enabling it to generate coherent and executable code snippets. The inclusion of the math dataset further expanded the model's capabilities, allowing it to solve complex equations, reason about abstract concepts, and perform precise calculations. Finally, exposure to the science dataset broadened the model's understanding of scientific theories, empirical observations, and experimental methods. The emergence of model capabilities is measured by evaluating the quality of the model responses, before and after training on the new domain, on a set of questions of varying difficulties from each domain. More precisely, the model is tested on 20 AI Society related tasks, 20 coding tasks, 20 math tasks and 60 science tasks.

Those results are highlighted in Table 2 where we see that each time we add a dataset, the model performs better on the incorporated domain. Note that to measure the quality of the models' responses, we follow the evaluation from Section T, which involves prompting a GPT4 agent to score and decide which solution is better. It is worth noting that an improvement on other domains is also observed in some cases such as when we train on Code we improve on Science. This is because our Code dataset contains problems that solve tasks in particular domains which include scientific domain. Similarly, training on AI Society improves code as AI Society contains the role of a "programmer" and hence coding related conversations. Finally, note that the draws observed in LLaMA-7B vs AI Society in Math reflects equally bad solutions compared to the draws observed in AI Society + Code + Math vs AI Society + Code + Math + Science where the draws are equally good solutions. This progression from AI society to code to math to science highlights the potential of AI models to acquire a versatile

and adaptable knowledge base, paralleling the way humans gain expertise in diverse subjects. Sample tasks are provided in the Appendix.

Table 2: **Emergence of Knowledge.** By progressively fine-tuning LLaMA on datasets from different domains, we observe the emergence of knowledge as the model transitions from AI society to code, math, and science. This finding is indicated by the fact that Model 2 almost always performs better than Model 1, especially on the added dataset.

Dataset	Model 1				Model 2				Draw	Model 1	Model 2
	AI Society	Code	Math	Science	AI Society	Code	Math	Science			
AI Society					✓				0	6	**14**
Code					✓				0	0	**20**
Math					✓				**9**	5	6
Science					✓				0	13	**47**
AI Society	✓				✓	✓			4	8	**8**
Code	✓				✓	✓			1	9	**10**
Math	✓				✓	✓			5	**8**	7
Science	✓				✓	✓			1	19	**40**
AI Society	✓	✓			✓	✓	✓		5	6	**9**
Code	✓	✓			✓	✓	✓		1	9	**10**
Math	✓	✓			✓	✓	✓		1	3	**16**
Science	✓	✓			✓	✓	✓		3	8	**49**
AI Society	✓	✓	✓		✓	✓	✓	✓	3	1	**16**
Code	✓	✓	✓		✓	✓	✓	✓	1	8	**11**
Math	✓	✓	✓		✓	✓	✓	✓	**10**	5	5
Science	✓	✓	✓		✓	✓	✓	✓	9	2	**49**
AI Society					✓	✓	✓	✓	0	0	**20**
Code					✓	✓	✓	✓	0	0	**20**
Math					✓	✓	✓	✓	0	0	**20**
Science					✓	✓	✓	✓	0	0	**60**

5.3   HumanEval$^{(+)}$

Table 3: **HumanEval$^{(+)}$ for Various Models.** We test our CAMEL model, which is a LLaMa-7B fine-tuned on all our datasets (AI Society, Code, Math, Science) on HumanEval and HumanEval^{+} benchmarks, where we show competitive pass@k scores with LLaMa-7B and Vicuna-7B.

	HumanEval		HumanEval^{+}	
pass@k [%]	$k = 1$	$k = 100$	$k = 1$	$k = 100$
gpt-3.5-turbo	69.4	94.0	61.7	89.8
LLaMA-7B	10.5	36.5	-	-
Vicuna-7B	11.0	42.9	9.9	34.7
CAMEL-7B	**14.0**	**57.9**	**12.2**	**50.0**

To evaluate the coding task-solving capabilities of our CAMEL model, specifically the LLaMA-7B fine-tuned on our comprehensive datasets, we rely on HumanEval [18] and HumanEval^{+} [69]. The results, as depicted in table 3, clearly demonstrate the remarkable performance of CAMEL. It surpasses not only the LLaMA-7B model but also Vicuna-7B [21] by a big margin. These findings underscore the critical role played by the generated datasets in enhancing LLaMA's ability to tackle coding-related tasks.

6   Conclusion

In this paper, we explore the potential of autonomous cooperation among communicative agents and propose a novel cooperative agent framework named *role-playing* . Our approach enables communicative agents to collaborate autonomously toward completing tasks while requiring minimal human intervention, leading to better solutions are per our thorough evaluations. Through our analysis, we show that achieving autonomous cooperation is challenging due to issues like conversation deviation, role flipping, and termination conditions. Our framework offers a scalable approach for studying the cooperative behaviors and capabilities of multi-agent systems and provides strategies for addressing these challenges. Furthermore, our open-sourced library includes implementations of various agents, data generation pipelines, data analysis tools, and collected datasets, to support research on communicative agents and beyond. Our contributions offer valuable insights into the future of large language artificial intelligence models and cooperative AI systems.

7   Acknowledgements

References

[1] Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, Petko Georgiev, Aurelia Guy, Tim Harley, Felix Hill, Alden Hung, Zachary Kenton, Jessica Landon, Timothy Lillicrap, Kory Mathewson, Soňa Mokrá, Alistair Muldal, Adam Santoro, Nikolay Savinov, Vikrant Varma, Greg Wayne, Duncan Williams, Nathaniel Wong, Chen Yan, and Rui Zhu. Imitating interactive intelligence, 2020.

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[3] Jacob Andreas. Language models as agent models, 2022.

[4] Jacob Andreas and Dan Klein. Alignment-based compositional semantics for instruction following. *arXiv preprint arXiv:1508.06491*, 2015.

[5] Anthropic. Introducing claude. *Anthropic Blog*, 2023.

[6] Isaac Asimov. *I. Robot*. Narkaling Productions., 1940.

[7] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

[8] Sanghwan Bae, Donghyun Kwak, Sungdong Kim, Donghoon Ham, Soyoung Kang, Sang-Woo Lee, and Woomyoung Park. Building a role specified open-domain dialogue system leveraging large-scale language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2128–2150, Seattle, United States, July 2022. Association for Computational Linguistics.

[9] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[10] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[11] Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

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

[13] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[14] N Carlini, F Tramer, E Wallace, M Jagielski, A Herbert-Voss, K Lee, A Roberts, T Brown, D Song, Ú Erlingsson, et al. Extracting training data from large language models. arxiv. *Preprint posted online December*, 14, 2020.

[15] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

[16] Harrison Chase. Langchain. 2022.

[17] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7028–7036, 2021.

[18] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[19] Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. Places: Prompting language models for social conversation synthesis. *arXiv preprint arXiv:2302.03269*, 2023.

[20] Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Andy Rosenbaum, Seokhwan Kim, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. Weakly supervised data augmentation through prompting for dialogue understanding. NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research, 2022.

[21] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[23] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[24] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, 1998.

[25] Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning, 2022.

[26] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. Cooperative ai: machines must learn to find common ground. *Nature*, 593(7857):33–36, 2021.

[27] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

[28] Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 456–464, 2021.

[29] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pages 456–463, 1994.

[30] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[31] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

[32] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and Machines*, 30:411 – 437, 2020.

[33] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374, 2018.

[34] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021.

[35] Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.

[36] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

[37] Josh A Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*, 2023.

[38] Dylan Hadfield-Menell. The principal-agent alignment problem in artificial intelligence. *Ph. D. dissertation*, 2021.

[39] Dylan Hadfield-Menell, McKane Andrus, and Gillian Hadfield. Legible normativity for ai alignment: The value of silly rules. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 115–121, 2019.

[40] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[41] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

[42] Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. Ethical challenges in data-driven dialogue systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2018.

[43] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[44] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019.

[45] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[46] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022.

[47] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[48] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

[49] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191, 2020.

[50] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022.

[51] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[52] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023.

[53] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.

[54] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[55] Siddharth Karamcheti, Megha Srivastava, Percy Liang, and Dorsa Sadigh. Lila: Language-informed latent actions. In *CoRL*, pages 1379–1390, 2021.

13

[56] Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. Alignment of language agents. *arXiv preprint arXiv:2103.14659*, 2021.

[57] Hyunwoo Kim, Jack Hessel, Liwei Jiang, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Mal-ihe Alikhani, Gunhee Kim, Maarten Sap, et al. Soda: Million-scale dialogue distillation with social commonsense contextualization. *arXiv preprint arXiv:2212.10465*, 2022.

[58] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[59] Yekyung Kim, Seohyeong Jeong, and Kyunghyun Cho. Linda: Unsupervised learning to interpolate in natural language processing. *arXiv preprint arXiv:2112.13969*, 2021.

[60] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[61] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

[62] Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. Augpt: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, 2021.

[63] Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*, 2021.

[64] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. 2022.

[65] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. Pre-trained language models for interactive decision-making, 2022.

[66] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[67] Zekun Li, Wenhu Chen, Shiyang Li, Hong Wang, Jing Qian, and Xifeng Yan. Controllable dialogue simulation with in-context learning. *arXiv preprint arXiv:2210.04185*, 2022.

[68] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[69] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*, 2023.

[70] Yat Long Lo, Christian Schroeder de Witt, Samuel Sokota, Jakob Nicolaus Foerster, and Shimon Whiteson. Cheap talk discovery and utilization in multi-agent reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[71] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

[72] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[73] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *ICLR*, 2023.

[74] Michael J. Matthews, Samuel H. Matthews, and Thomas K. Kelemen. The alignment problem: Machine learning and human values. *Personnel Psychology*, 2022.

[75] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[76] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.

[77] Marvin Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.

[78] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, 2022.

[79] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[80] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2021.

[81] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021.

[82] OpenAI. Introducing chatgpt. *Open AI Blog*, 2022.

[83] OpenAI. Chatgpt plugins. *OpenAI blog*, 2023.

[84] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[85] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, 2005.

[86] Alexandros Papangelis, Karthik Gopalakrishnan, Aishwarya Padmakumar, Seokhwan Kim, Gokhan Tur, and Dilek Z. Hakkani-Tür. Generative conversational networks. In *SIGDIAL*, 2021.

[87] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[88] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, pages 13657–13665, 2021.

[89] Sundar Pichai. An important next step on our ai journey. *Google Blog*, 2023.

[90] Stefan Poslad. Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 2(4):15–es, 2007.

[91] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[92] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022.

[93] Laria Reynolds and Kyle McDonell. 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*, pages 1–7, 2021.

[94] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[95] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[96] Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. *arXiv preprint arXiv:2209.09900*, 2022.

[97] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[98] Gaurav Sahu, Pau Rodriguez, Issam H Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. *ACL*, 2022.

[99] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

[100] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[101] Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[102] Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(2):50, 2022.

[103] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. In *ICLR*, 2023.

[104] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[105] Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2023.

[106] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*, 2022.

[107] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[108] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[109] Abishek Sridhar, Robert Lo, Frank F. Xu, Hao Zhu, and Shuyan Zhou. Hierarchical prompting assists large language model on web navigation. In *ArXiv*, preprint.

[110] Jonathan Stray. Aligning ai optimization to community well-being. *International Journal of Community Well-Being*, 3:443 – 463, 2020.

[111] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

[112] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

[113] Ming Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *International Conference on Machine Learning*, 1997.

[114] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[115] Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[116] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[117] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[118] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

[119] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[120] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Supernaturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*, 2022.

[121] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[122] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. Survey Certification.

[123] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[124] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[125] Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[126] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[127] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[128] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022.

[129] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *ACL*, 2020.

[130] Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiao-Xi Mao, Yadong Xi, and Minlie Huang. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. *ArXiv*, abs/2009.09427, 2020.

[131] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[132] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *ICLR*, 2023.

[133] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[134] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[135] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[136] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023.

[137] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions, 2023.