

Towards Autonomous Agents: A Review of prompting techniques to increase autonomy of Multi-Agent Systems

Andrew David Ratnam Sathiadass

Online BS Degree, Indian Institute Of Technology Madras

Abstract

As Large Language Models (LLMs) expand dramatically in size, from approximately 94 million parameters to nearly 100 billion parameters, so too have the methodologies for harnessing their full potential. This paper reviews the evolution of prompting techniques, from naive prompting to advanced prompt decomposition and chaining methodologies, and the utilization of various agents to enhance task efficacy. We explore a focused range of approaches for leveraging LLMs, culminating in the use of multi-agent systems and frameworks to decompose decisions and tasks effectively. Our review begins with an overview of various LLMs and proceeds to examine key prompting strategies, including Chain-of-Thought, ReAct. Finally, we delve into the realm of autonomous agents designed to execute individual but synchronized tasks, aiming for autonomy in AI applications. Despite these advancements, we acknowledge the inherent biases in LLMs that currently limit their implementation in fully autonomous systems. This paper provides a brief but comprehensive examination of the state-of-the-art techniques and highlights the challenges and future directions in achieving true autonomy with LLMs.

1 Introduction

Large Language Models are adequately suited for solving reasoning and decision making tasks however satisfactory by themselves with very little help (Brown, 2020) but with some novel methods and prompt engineering . Particularly, intermediate reasoning steps like chain-of-thought prompting (Wei, 2022), interleaving reasoning and task-specific actions in the ReAct method (Yao, 2022). After reasoning out the relevant tasks to complete in order to reach the goal state, they must act to complete their tasks through the usage of tools. LLMs are capable of using real-world tools, provided that their inputs and outputs can be represented as text sequences and also fine tune them to correctly implement such tools as in a particular case. (Schick, 2024). Similarly WebGPT does so in the browser environment using a combination of behavior cloning and rejection sampling among other methods. Although they are not perfect, they clearly indicate that with certain predetermined trajectories similar to how humans operate like clicking on the search bar to search when we are unsure of certain information(n-sampling in WebGPT similar to Monte Carlo Methods) but also are able to understand what we do not know and update our own knowledge base and it's embeddings and representation in our own brain so to speak,

similarly using these tools to search and reinforce previous decisions and look for a better implementation of said tasks, LLMs can show increased efficacy in such matters. To further autonomy, there can be task assigner agent, a decision maker agent, and evaluator agent to break down roles and form a system of interacting agents that set tasks and autonomously complete them (Weize, 2023). Just as a baby needs a nurturing environment to grow, learn and also explore and exploit previously understood methodologies, like cry if hungry, cry to defecate to verbally communicate and express emotions which the environment's agents understands and further nurtures in a proactive feedback loop. Eventually the environment is rarely needed as the baby grows, so do these autonomous agents. There are different frameworks to provide an environment namely, Fetch-ai's uAgents (fetchai) and AgentScope (Alibaba, 2024). LLMs have also been upgraded to large-scale multimodal models (OpenAI, 2023) (Google and Gemini Team, 2024) which may have assumedly used this methodology and have achieved such performance due to sheer scale of training parameters, but that is not the primary focus of this review. The only logically next step is to carefully review this progressive stride in autonomy all thanks to LLMs.

2 Overview of Large Language Models

First we start by reviewing different LLMs, even multimodal models (they are included because the paper "A comprehensive overview of large language models" (Naveed, 2023) was indeed comprehensive), and how the sheer scale of different models contribute to their performance. Figure 1 is from the same paper.

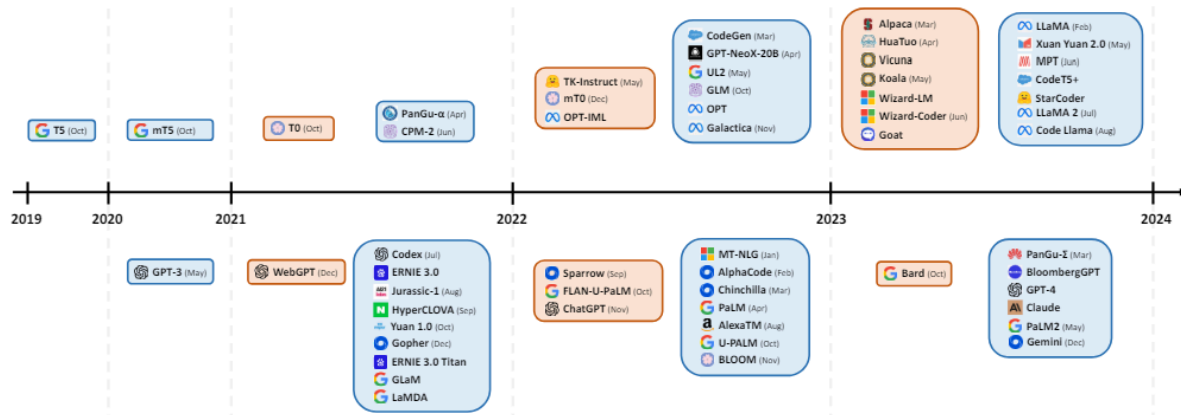


Figure 1: Chronological display of LLM releases: blue cards represent 'pre-trained' models, while orange cards correspond to 'instruction-tuned' models. Models on the upper half signify open-source availability, whereas those on the bottom half are closed-source. The chart illustrates the increasing trend towards instruction tuned models and open-source models, highlighting the evolving landscape and trends in natural language processing research. (Naveed, 2023)

They go on to review various models and their performance on various datasets and benchmarks, indicated by Table 1. They also give a comprehensive summary of pretrained models which also include training time and calculated training cost in Table 2. The size of the models are only going to increase and so are the accuracy on various benchmarks in terms of zero-shot (Kojima, 2022) and few-shot learning (Brown, 2020). Even with simple finetuning, the accuracy can increase. Although LLM's have inherent knowledge, we still need to extract it using various methods that help in answering Long form question answering (Fan, 2019), reasoning and even task completion. LLMs need to carry out these tasks in a

systematic manner but also autonomously with minimal human intervention. Table 1 and 2 from (Naveed, 2023).

| Task | Dataset/Benchmark | Top-1 | | Top-2 | | Top-3 | |
|--|-------------------|-------------------------------|------------------|----------------------------|------------------|------------------------------------|-------------------|
| | | Model (Size) | Score (N-shots) | Model (Size) | Score (N-shots) | Model (Size) | Score (N-shots) |
| Multi-Task | BIG-bench (B) | Chinchilla (70B) | 65.1 (5-shot) | Gopher (280B) | 53.97 (5-shot) | PaLM (540B) | 53.7 (5-shot) |
| | MMLU (B) | GPT-4 (-) | 86.4 (5-shot) | Gemini (Ultra) | 83.7 (5-shot) | Flan-PaLM-2 _(f) (Large) | 81.2 (5-shot) |
| Language Understanding | SuperGLUE (B) | ERNIE 3.0 (12B) | 90.6 (-) | PaLM _(f) (540B) | 90.4 (-) | T5 (11B) | 88.9 (-) |
| Story Comprehension and Generation | HellaSwag | GPT-4 (-) | 95.3 (10-shot) | Gemini (Ultra) | 87.8 (10-shot) | PaLM-2 (Large) | 86.8 (one shot) |
| | StoryCloze | GPT3 (175B) | 87.7 (few shot) | PaLM-2 (Large) | 87.4 (one shot) | OPT (175B) | 79.82 (-) |
| Physical Knowledge and World Understanding | PIQA | PaLM-2 (Large) | 85.0 (one shot) | LLaMa (65B) | 82.8 (zero shot) | MT-NLG (530B) | 81.99 (zero shot) |
| | TriviaQA | PaLM-2 (Large) | 86.1 (one shot) | LLaMa-2 (70B) | 85.0 (one shot) | PaLM (540B) | 81.4 (one shot) |
| Contextual Language Understanding | LAMBADA | PaLM (540B) | 89.7 (few shot) | MT-NLG (530B) | 87.15 (few shot) | PaLM-2 (Large) | 86.9 (one shot) |
| Commonsense Reasoning | WinoGrande | GPT-4 (-) | 87.5 (5-shot) | PaLM-2 (Large) | 83.0 (one shot) | PaLM (540B) | 81.1 (zero shot) |
| | SIQA | LLaMa (65B) | 52.3 (zero shot) | Chinchilla (70B) | 51.3 (zero shot) | Gopher (280B) | 50.6 (zero shot) |
| Reading Comprehension | BoolQ | PaLM _(f) (540B) | 92.2 (-) | T5 (11B) | 91.2 (-) | PaLM-2 (Large) | 90.9 (one shot) |
| Truthfulness | Truthful-QA | LLaMa (65B) | 57 (-) | | | | |
| Mathematical Reasoning | MATH | Gemini (Ultra) | 53.2 (4-shot) | PaLM-2 (Large) | 34.3 (4-shot) | LLaMa-2 (65B) | 13.5 (4-shot) |
| | GSM8K | GPT-4 (-) | 92.0 (5-shot) | PaLM-2 (Large) | 80.7 (8-shot) | U-PaLM (540B) | 58.5 (-) |
| Problem Solving and Logical Reasoning | HumanEval | Gemini _(f) (Ultra) | 74.4 (zero shot) | GPT-4 (-) | 67.0 (zero shot) | Code Llama (34B) | 48.8 (zero shot) |

Table 1: Performance comparison of top performing LLMs across various NLU ¹and NLG ²tasks. Here, “N-Shots” indicate the number of example prompts provided to the model during the evaluation, representing its capability in few-shot or zero-shot learning settings, “f” represents the fine-tuned version, and “B” represents the benchmark.

| Models | Publication Venue | License Type | Model Creators | Purpose | No. of Params | Commercial Use | Steps Trained | Data/ Tokens | Data Cleaning | No. of Processing Units | Processing Unit Type | Training Time | Calculated Train. Cost | Training Parallelism | Library |
|----------------------|-------------------|--------------|----------------|----------|---------------|----------------|---------------|--------------|-------------------|-------------------------|----------------------|---------------|------------------------|----------------------|---------------------|
| T5 [10] | JMLR'20 | Apache-2.0 | Google | General | 11B | ✓ | 1M | 1T | Heur+Dedup | 1024 | TPU v3 | - | - | D+M | Mesh TensorFlow |
| GPT-3 [6] | NeurIPS'20 | - | OpenAI | General | 175B | × | - | 300B | Dedup+QF | - | V100 | - | - | M | - |
| mT5 [11] | NAACL'21 | Apache-2.0 | Google | General | 13B | ✓ | 1M | 1T | - | - | - | - | - | - | - |
| PanGu-o [108] | arXiv'21 | Apache-2.0 | Huawei | General | 200B | ✓ | 260k | 1.1TB | Heur+Dedup | 2048 | Ascend 910 | - | - | D+OP+P+O+R | MindSpore |
| CPM-2 [12] | AI Open'21 | - | MIT | Tsinghua | General | 198B | ✓ | 1M | 2.6TB | Dedup | - | - | - | D+M | JAXFormer |
| Codex [131] | arXiv'21 | - | OpenAI | Coding | 12B | × | - | 100B | Heur | - | - | - | - | - | - |
| ERNIE 3.0 [110] | arXiv'21 | - | Baidu | General | 10B | × | 120k* | 375B | Heur+Dedup | 384 | V100 | - | - | M* | PaddlePaddle |
| Jurassic-1 [112] | White-Paper'21 | Apache-2.0 | AI21 | General | 178B | ✓ | - | 300B | - | 800 | GPU | - | - | D+M+P | Megatron+DS |
| HyperCLOVA [114] | EMNLP'21 | - | Naver | General | 82B | × | - | 300B | Clf+Dedup+PF | 1024 | A100 | 321h | 1.32 Mil | M | Megatron |
| Yuan 1.0 [115] | arXiv'21 | Apache-2.0 | - | General | 245B | ✓ | 26k* | 180B | Heur+Clf+Dedup | 2128 | GPU | - | - | D+T+P | - |
| Gopher [116] | arXiv'21 | - | Google | General | 280B | × | - | 300B | QF+Dedup | 4096 | TPU v3 | 920h | 13.19 Mil | D+M | JAX+Haiku |
| ERNIE 3.0 Titan [35] | arXiv'21 | - | Baidu | General | 260B | × | - | 300B | Heur+Dedup | - | Ascend 910 | - | - | D+M+P+D* | PaddlePaddle |
| GPT-NeoX-20B [118] | BigScience'22 | Apache-2.0 | EleutherAI | General | 20B | ✓ | 150k | 825GB | None | 96 | 40G A100 | - | - | M | Megatron+DS+PyTorch |
| OPT [14] | arXiv'22 | MIT | Meta | General | 175B | ✓ | 150k | 180B | Dedup | 992 | 80G A100 | - | - | D+T | Megatron |
| BLOOM [13] | arXiv'22 | RAIL-1.0 | BigScience | General | 176B | ✓ | - | 366B | Dedup+PR | 384 | 80G A100 | 2520h | 3.87 Mil | D+T+P | Megatron+DS |
| Galactica [138] | arXiv'22 | Apache-2.0 | Meta | Science | 120B | × | 225k | 106B | Dedup | 128 | 80GB A100 | - | - | - | MetaSeq |
| GLaM [91] | ICML'22 | - | Google | General | 1.2T | × | 600k* | 600B | Clf | 1024 | TPU v4 | - | - | M | GSPMD |
| LaMDA [140] | arXiv'22 | - | Google | Dialog | 137B | × | 3M | 2.81T | Filtered | 1024 | TPU v3 | 1384h | 4.96 Mil | D+M | Lingvo |
| MT-NLG [117] | arXiv'22 | Apache-v2.0 | MS+Nvidia | General | 530B | × | - | 270B | - | 4480 | 80G A100 | - | - | D+T+P | Megatron+DS |
| AlphaCode [132] | Science'22 | Apache-v2.0 | Google | Coding | 41B | ✓ | 205k | 967B | Heur+Dedup | - | TPU v4 | - | - | M | JAX+Haiku |
| Chinchilla [96] | arXiv'22 | - | Google | General | 70B | × | - | 1.4T | QF+Dedup | - | TPUv4 | - | - | - | JAX+Haiku |
| PaLM [15] | arXiv'22 | - | Google | General | 540B | × | 255k | 780B | Heur | 6144 | TPU v4 | - | - | D+M | JAX+T5X |
| AlexaTM [122] | arXiv'22 | Apache v2.0 | Amazon | General | 20B | × | 500k | 1.1T | Filtered | 128 | A100 | 2880h | 1.47 Mil | M | DS |
| U-PaLM [124] | arXiv'22 | - | Google | General | 540B | × | 20k | - | - | 512 | TPU v4 | 120h | 0.25 Mil | - | - |
| UL2 [125] | ICLR'23 | Apache-2.0 | Google | General | 20B | ✓ | 2M | 1T | - | 512 | TPU v4 | - | - | M | JAX+T5X |
| GLM [33] | ICLR'23 | Apache-2.0 | Multiple | General | 130B | × | - | 400B | - | 768 | 40G A100 | 1440h | 3.37 Mil | M | - |
| CodeGen [130] | ICLR'23 | Apache-2.0 | Salesforce | Coding | 16B | ✓ | 650k | 577B | Heur+Dedup | - | TPU v4 | - | - | D+M | JAXFormer |
| LLaMA [127] | arXiv'23 | - | Meta | General | 65B | × | 350k | 1.4T | Clf+Heur+Dedup | 2048 | 80G A100 | 504h | 4.12 Mil | D+M | xFormers |
| PanGuE [92] | arXiv'23 | - | Huawei | General | 1.085T | × | - | 329B | - | 512 | Ascend 910 | 2400h | - | D+OP+P+O+R | MindSpore |
| BloombergGPT [141] | arXiv'23 | - | Bloomberg | Finance | 50B | × | 139k | 569B | Dedup | 512 | 40G A100 | 1272h | 1.97 Mil | M | PyTorch |
| Xuan Yuan 2.0 [142] | arXiv'23 | RAIL-1.0 | Du Xiaoman | Finance | 176B | ✓ | - | 366B | Filtered | 80GB | A100 | - | - | P | DS |
| CodeT5+ [34] | arXiv'23 | BSD-3 | Salesforce | Coding | 16B | ✓ | 110k | 51.5B | Dedup | 16 | 40G A100 | - | - | - | DS |
| StarCoder [137] | arXiv'23 | OpenRAIL-M | BigCode | Coding | 15.5B | ✓ | 250k | 1T | Dedup+QF+PF | 512 | 80G A100 | 624h | 1.28 Mil | D+T+P | Megatron-LM |
| LLaMA-2 [21] | arXiv'23 | LLaMA-2.0 | Meta | General | 70B | ✓ | 500k | 2T | Minimal Filtering | - | 80G A100 | 1.7Mh | - | - | - |
| PaLM-2 [123] | arXiv'23 | - | Google | General | - | × | - | - | Ddedup+PF+QF | - | - | - | - | - | - |

Table 2: Summary of pre-trained LLMs (>10B). “Data/Tokens” is the model’s pre-training data, which is either the number of tokens or data size. “Data Cleaning” indicates whether data cleaning is performed or not. This includes heuristics (Heur), deduplication (Dedup), quality filtering (QF), and privacy filtering (PF), “Cost” is the calculated training cost obtained by multiplying the GPUs/TPUs hourly rate with the number of GPUs and the training time. The actual cost may vary due to many reasons such as using in-house GPUs or getting a discounted rate, re-training, number of employees working on the problem, etc. “Training Parallelism” indicates distributed training using data parallelism (D), tensor parallelism (T), pipeline parallelism (P), model parallelism (M), optimizer parallelism (OP), and rematerialization (R), where for “Library” column, “DS” is a short form for Deep Speed. In column “Commercial Use”, we assumed a model is for non-commercial purposes if its license is unavailable. (Naveed, 2023)

¹ Natural Language Understanding

² Natural Language Generation

3 Prompt Engineering Methods

LLMs rely on prompts to generate output, generally these prompts are embedded and transformed multiple times to generate meaningful representations, using self-attention, multi-head attention and masking and uses a probability distribution(via softmax) to predict the next sequence or word, mostly a token (Vaswani, 2017). Since prompts are the inputs it makes sense that they should be well formulated and convey context to what is clearly expected from the Model. For example, different words have different contexts, and it's up to the input prompt to clearly imply what is required. Modern LLMs are so good that zero-shot(Kojima, 2022) and few-shot(Brown, 2020) learning is good enough to provide satisfactory results Long Form Question Answering and various related datasets. To truly utilize the full capacity of the LLMs, we discuss methods that improve accuracy on such datasets as the model scales, and show less promise for smaller models. Since models are only getting bigger, this seems like an optimistic direction to explore.

3.1 Chain-of-thought Prompting

As the title of the paper clearly states chain-of-thought prompting elicits reasoning in large language models (Wei, 2022). It is akin to the thought process of humans, though it doesn't explicitly factor in the thought space like other methods. In a sense it uses intermediate steps that have higher contextual reasoning capabilities from the previous step than a one shot response would have with respect to contextual reasoning.

3.1.1 Few-shot learning is good but.....

Scaling the model size and providing a few exemplars does indeed grant a boost to the reasoning capabilities of the model, however in the areas of arithmetic(word problems), commonsense and symbolic reasoning the model could do better without increasing the model complexity. Firstly by generating intermediate,rational steps as a prompt that consists of triples{input, chain of thought, output} leading to the final answer. Secondly, this is much more computationally friendly than finetuning a LLM. The chain-of-thought prompting study then presents empirical evaluations on arithmetic, commonsense reasoning and symbolic reasoning datasets and benchmarks.

3.1.2 What is Chain-of-thought Prompting?

The paper describes chain-of-thought prompting (CoT) as simply steps or chains of reasoning steps that the language model uses to achieve its goal state. This is different from few-shot prompting as described by (Brown, 2020) in which the model receives in-context input-output exemplars before outputting a prediction at inference and the model gives just the answer, not a broken down chain of thought reasoning to why it came to that solution. Figure 2 shows an example of standard prompting and CoT. Like a fine tuned model, it shows that the model is capable of reasoning.

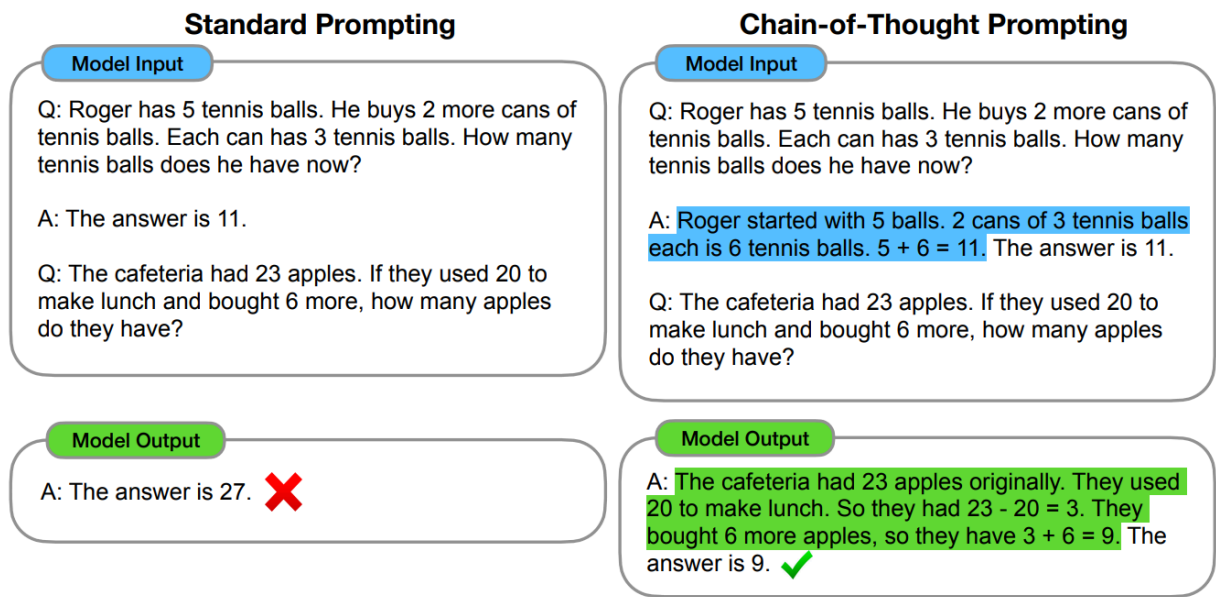


Figure 2: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted (Wei, 2022)

The authors also elucidate the various attractive properties of CoT prompting to generate reasoning in language models such as, since **there is a chain of steps, more computation can be allocated for problems that require more reasoning steps**, it shows **some interpretability** in its method at arriving at a solution. These methods can be directly used simply in large language models but with sufficient parameters by creating CoT exemplars like few-shot prompting. The paper examines arithmetic, commonsense, and symbolic reasoning

3.1.3 Arithmetic Reasoning

BenchMarks. **GSM8K**(word math problems), **SVAMP**(varying structures of math problems), **ASDiv**, **AQuA**(algebraic word problems), **MAWPS**. Appendix A has more information

Control: Standard few-shot prompting (Brown, 2020). Exemplars are formatted as in Figure 2 (left). {input, output}

Experiment: Each exemplar is augmented as {input, chain of thought, output} differing from few-shot prompting. *A chain of thought is a series of intermediate natural language reasoning steps that lead to the final output*, as defined by the authors. Except for AQua which used four exemplars, the rest of the benchmarks used eight chain of thought exemplars. In Appendix A, Table 3.1 contains exemplars for the **GSM8K**, **SVAMP**, **ASDiv**, **MAWPS** and Table 3.2 for the **AQuA** exemplars.

Language Models: **GPT3**(350M, 1.3B, 6.7B, 175B), **LaMDA**(422M, 2B, 8B, 68B, 137B), **PaLM**(8B, 62B, 540B), **UL2 20B** and **Codex**. More information about language models available in Appendix A.

Initially although the authors sampled via greedy decoding, follow up works(Wang, 2022) showed that CoT can be improved by taking a Monte Carlo approach over many differing sampled generations of the same output(Chain-of-Thought Self-Consistency (Wang, 2022)), i.e. majority final answer. As LaMDA showed negligible variance among various seeds, the authors reported single exemplar models for all other models too.

Results: All the strongest and notable results are perfectly summarized in Figure 4. Table 3.3 in Appendix A contains all experimental outputs for all model sizes for each of the models and the benchmarks.

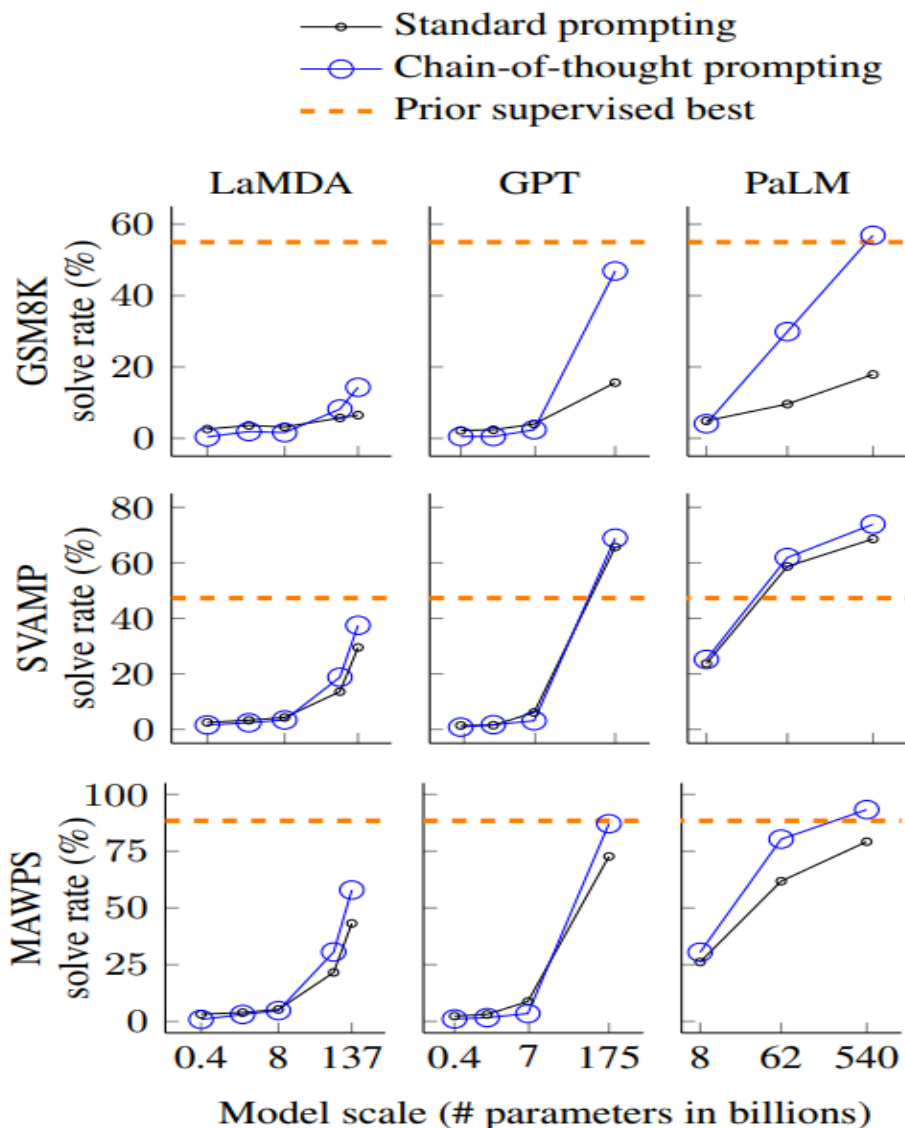


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Previous bests are (Cobbe, 2021) for GSM8K, (Jie, 2022) for SVAMP, (Lan, 2022) for MAWPS

The authors found three important takeaways :-

1. To get sufficient performance gains the models are recommended to have at least 100B parameters but some do better like on some benchmarks like PaLM 64B but not on all benchmarks. Clearly performance improvement has a threshold and **scales with the model**. The authors also found out that **smaller scale models produced “fluent but illogical chains of thought”**, which lead to lower performance than standard prompting.
2. Also, if the problem involved **more complicated substeps** via reasoning the model had **larger performance gains**. The authors reported “for GSM8K (the dataset with the lowest baseline performance), performance more than doubled for the largest GPT and PaLM models. On the other hand, for SingleOp, the easiest subset of MAWPS which only requires a single step to solve, performance improvements were either negative or very small”(see table 3.4).
3. Compared to prior SOTA(state of the art) which typically uses a finetuned task specific model, chain of thought prompting via GPT-3(175B) and PaLM(540B) competes quite favorably as seen in Figure-4. The authors showed that “ how PaLM 540B uses chain-of-thought prompting to achieve new state of the art on GSM8K, SVAMP, and MAWPS (though note that standard prompting already passed the prior best for SVAMP). On the other two datasets, AQuA and ASDiv, **PaLM with chain-of-thought prompting reaches within 2% of the state of the art**” (Appendix Table 3.3).

To better understand chain-of-thought prompting and why it worked the authors examined 50 random examples, where the model returned the correct final answer and another 50 where the model returned the wrong final answer for the LaMDA(137B)’s generated chains of thought. The authors reported that, in correct final answers, all generated chains of thought were logically and mathematically correct except two, but still arrived at the correct answer. In wrong final answers, 46% chains of thought were almost correct(calculator error, symbol error, one reasoning step error) , the other 56% had major errors in semantic understanding and coherence. For more depth please review Appendix D.2 of (Wei, 2022).

Ablation Study: Figure 5 clearly shows that chain-of-thought prompting is the key factor for increased performance, as the other methods perform similarly to standard prompting.

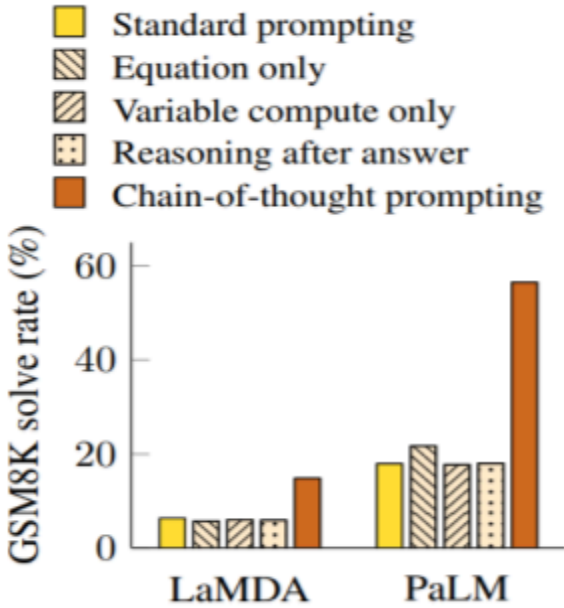
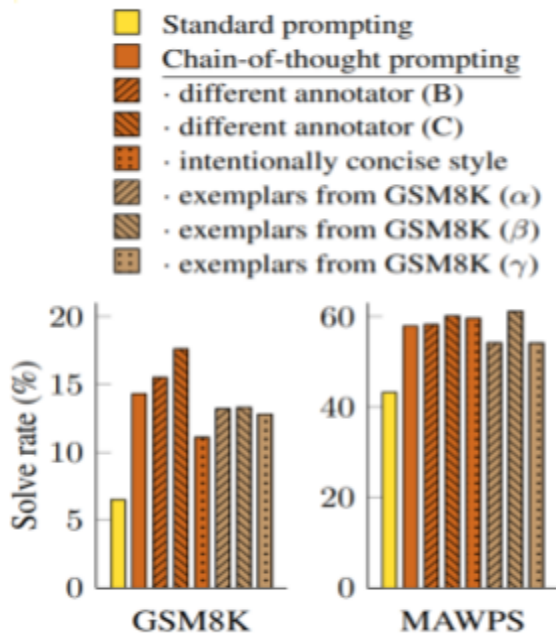


Figure 5: Ablation study for different variations of prompting using LaMDA 137B and PaLM 540B. Results for other datasets are given in Appendix A Table 3.5 and Table 3.6.

The authors used three variations, namely **Equation only**(a variation where the model is prompted to output only a mathematical equation before giving the answer), **Variable compute only**(To isolate the effect of variable computation from chain-of-thought reasoning, they test a configuration where the model is prompted to output a only sequence of dots (. . .) equal to the number of characters in the equation needed to solve the problem), **chain of thought after answer**(the chain of thought prompt is only given after the answer, isolating whether the model actually depends on the produced chain of thought to give the final answer).

Robustness of Chain of Thought:



The authors then reviewed their novel method with different exemplars written by different authors(annotators) of the paper and a concise style of solutions from (Cobbe), and also , they also ran experiments with three sets of eight exemplars randomly sampled from the GSM8K training set, an independent source.

In addition to robustness of these variations, the chain of thought is also robust to different exemplar orders and varying numbers of exemplars, the authors reported

Figure 6 Chain-of-thought prompting has variance for different prompt examples (as expected) but outperforms standard prompting for various annotators as well as for different exemplars.

3.1.4 Commonsense Reasoning

Using chain-of-thought for common sense reasoning . (Talmor, 2022) shows that a fine tuned T5 (Raffel) model called Unicorn (Lourie, 2021) achieves a few points below human accuracy, that was **in 2021**, clearly with the growth of LLMs, modern LLMs with fine tuning would most probably surpass average human common reasoning which could be direction for future research.

Benchmarks: **CSQA**(commonsense questions involving complex semantics requiring prior knowledge), **StrategyQA**(multi-hop strategy to answer questions), **Date**³(inferring a date from the context), **Sports**⁴(plausible or implausible from a sports context), **SayCan**(natural language to a sequence of robot actions from a discrete set). Visit Appendix A for more information about the dataset.

Prompts: Same experimental setup as the previous sections, create exemplars from the training set, or if the training set is absent create it from the first few examples, manually.

Results: Figure 7 highlights these results for PaLM. Full results in table 3.7 . Results summed as follows:-

1. **Scaling Model size improved performance** of standard prompting and **chain of thought prompting led to further gains**, largest for PaLM 540B (largest in tested models).
2. “With **chain-of-thought prompting**, PaLM 540B achieved strong performance relative to baselines, **outperforming the prior state of the art on StrategyQA** (75.6% vs 69.4%) and **outperforming an unaided sports enthusiast on sports understanding** (95.4% vs 84%).” , the authors reported
3. These results imply that Cot can be used to improve performance in common sense reasoning abilities of LLMs.

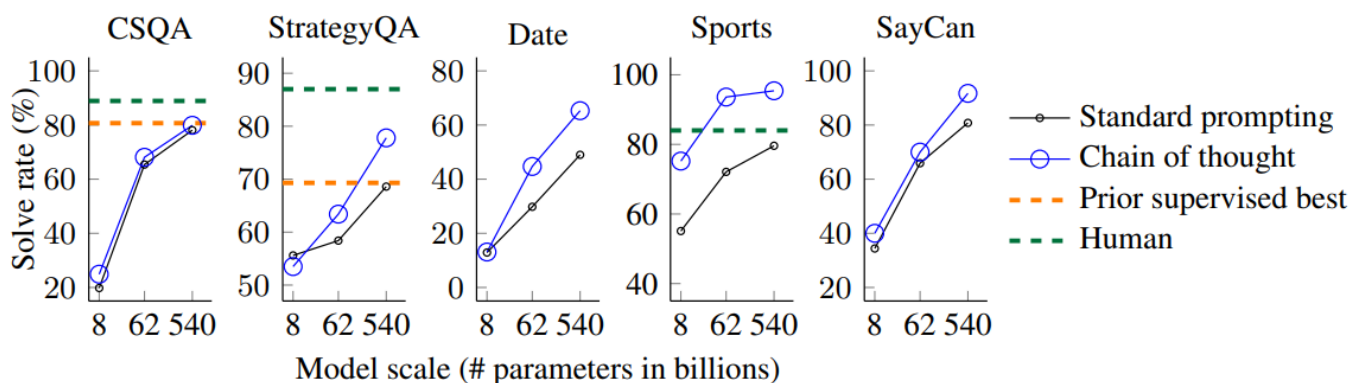


Figure 7: Chain-of-thought prompting also improves the commonsense reasoning abilities of language models. The language model shown here is PaLM. Prior best numbers are from the leaderboards of CSQA (Talmor, 2022) and StrategyQA (Geva, 2021) (single-model only, as of May 5, 2022). Additional results using various sizes of LaMDA, GPT-3, and PaLM are shown in Table 3.7

³ From BigBench(Srivastava, 2022)

⁴ From BigBench(Srivastava, 2022)

3.1.5 Symbolic Reasoning

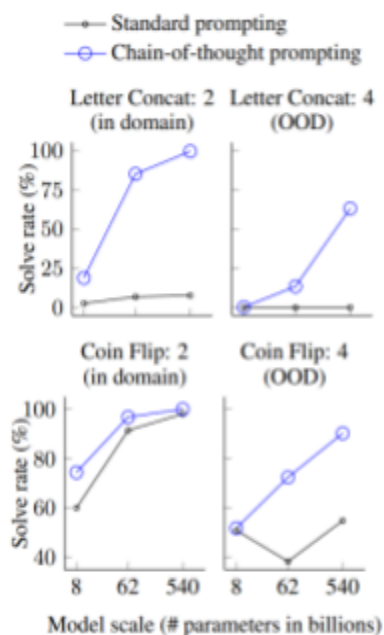


Figure 8: Using chain-of-thought prompting facilitates generalization to longer sequences in two symbolic reasoning tasks.

In a nutshell, symbolic reasoning is reasoning with symbols used to represent concepts, like proof writing in mathematics. The authors then showed that not only does chain-of-thought perform symbolic reasoning that is challenging for standard prompting but also generalizes to longer inputs than to what seen during training at inference time.

Tasks: The authors used two toy tasks (for more information, please visit the Appendix A), namely:-

1. **Last letter concatenation**
2. **Coin Flip**

Prompting: In-domain and out-of-domain (OOD) where exemplars had the same number of steps or more number of steps in the training set/stage respectively. The authors stated that they used only 2 words for names and 2 potential flip cases (where the coin was flipped twice).

Results: The authors reported that “With PaLM 540B, chain-of-thought prompting leads to almost 100% solve rates (note that standard prompting already solves coin flip with PaLM 540, though not for LaMDA 137B). Note that these in-domain evaluations

are “toy tasks” in the sense that perfect solution structures are already provided by the chains of thought in the few-shot exemplars; all the model has to do is repeat the same steps with the new symbols in the test-time example. And yet, small models still fail—the ability to perform abstract manipulations on unseen symbols for these three⁵ tasks only arises at the scale of 100B model parameters” for in-domain test sets. For the out of domain (OOD) evaluations, standard prompting fails whereas CoT achieves better performance, but performance gains are much more significant simply because the control fails. Figure 8 contains the results for PaLM 540B.

3.1.6 Related Work

To summarize, related works include using intermediate steps to solve math problems, finetune a model on larger dataset, than start from scratch like transfer training, also predicting the final output of Python programs computing intermediate computational results from each line then show that this step-by-step evaluation leads to a better final output than directly predicting the output. Please view Appendix A for more information on the extended and related work.

The authors defend their novelty, by saying that these related works augment the prompt input, whereas their work changes the output from the model with a chain of thought.

⁵ The author mentions three tasks but only illustrates two, three means three sub headings of reasoning (common sense, arithmetic and symbolic) but the others don’t have OOD so not clear to what the author is referring to.

In (Wang, 2022), they improve CoT by doing the following, for UL2-20B and LaMDA-137B they applied temperature sampling with $T = 0.5$ and truncated at the top-k ($k = 40$) tokens with the highest probability, for PaLM-540B they applied $T = 0.7$, $k = 40$, and for GPT-3 they use $T = 0.7$ without top-k truncation, this intuitively says that reasoning has different pathways to reach a similar answer and the authors explore those different paths by temperature sampling (generating different intermediate steps with the hyper parameter temperature (more is more random, less is more probable next step)) and generate different answers and then take a majority vote among other similar methods.

3.1.7 Conclusions

The authors conclude by stating that “**chain-of-thought prompting is(as) a simple and broadly applicable method for enhancing reasoning in language models. Through experiments on arithmetic, symbolic, and commonsense reasoning, we find that chain-of-thought reasoning is an emergent property of model scale that allows sufficiently large language models to perform reasoning tasks that otherwise have flat scaling curves**”. It elicits reasoning capabilities in LLMs and scales with model scale is what they conclude their paper with.

3.2 ReAct: Synergizing Reasoning and Acting In Language Models

LLMs are improving on average daily, new models being introduced every few months with increasing capabilities. ReAct helps use LLMs to generate reasoning traces for interpretability and task-specific actions based on the reasoning and boosts the synergy between the two, whereas previous methods studied these as separate topics, ReAct brings them together by interleaving them, showing greater effectiveness than SOTA (as of 2022) benchmarks while improving human interpretability and trustworthiness.

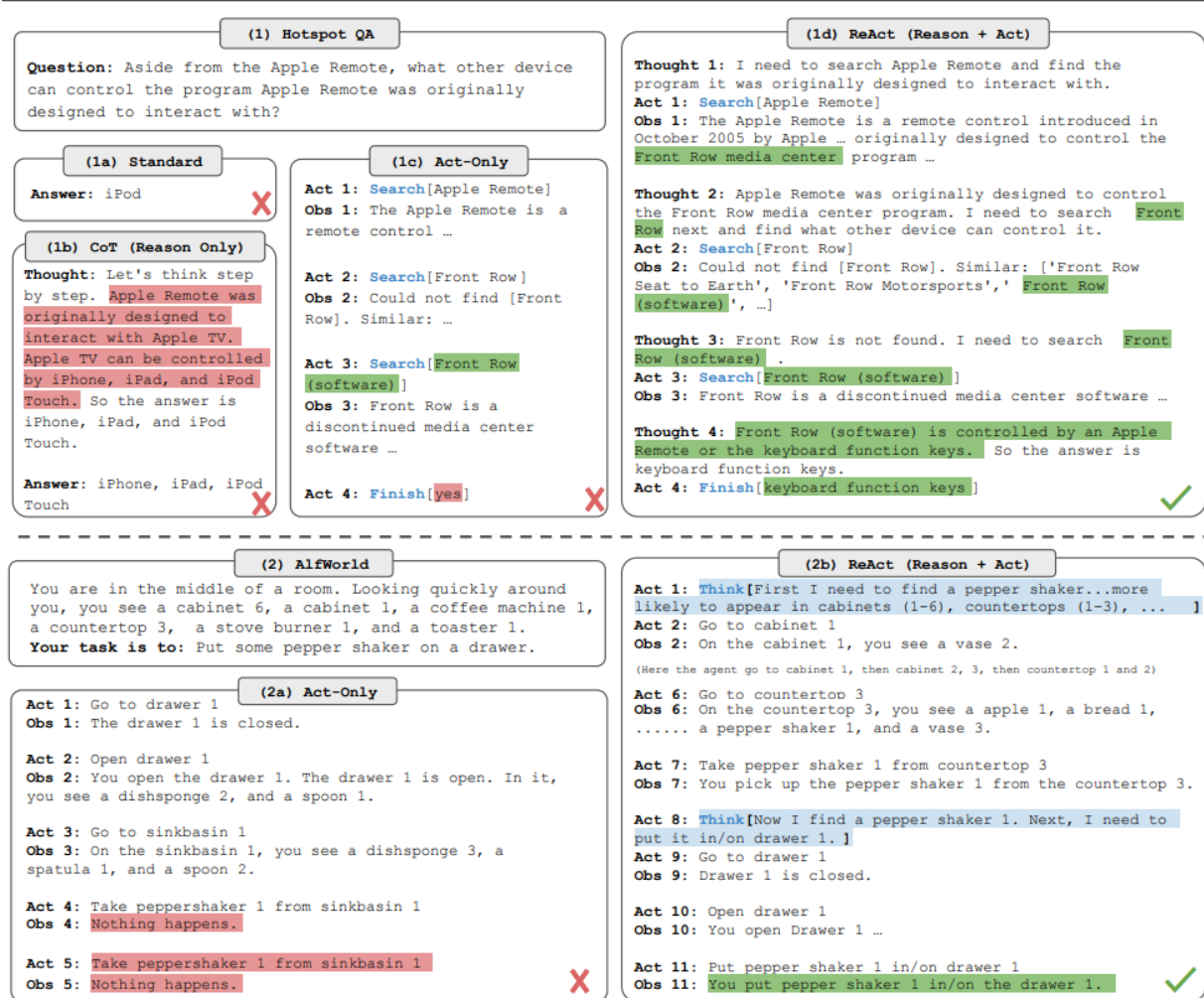


Figure 9: (1) Comparison of 4 prompting methods, (a) Standard, (b) Chain-of-thought (CoT, Reason Only (Wei, 2022)), (c) Act-only, and (d) ReAct (Reason+Act), solving a HotpotQA (Yang, 2018) question. (2) Comparison of (a) Act-only and (b) ReAct prompting to solve an AlfWorld (Shridhar, 2020) game. In both domains, we omit in-context examples in the prompt, and only show task solving trajectories generated by the model (Act, Thought) and the environment (Obs).

3.2.1 Chain-of-thought helps reason but so do actions to double check...

The chain of thought generates reasoning intermediate steps, these reasoning steps are based on its internal representation of information(that's a black box) and may have shown to have hallucinogenic properties (Mitta, 2023l). The authors claim that since no major studies show how reasoning and acting can be combined to work together, they propose a novel method to do so.

“ReAct prompts LLMs to generate both verbal reasoning traces and actions pertaining to a task in an interleaved manner, which allows the model to perform dynamic reasoning to create, maintain, and adjust high-level plans for acting (reason to act), while also interact⁶ with the external environment (eg. Wikipedia) to incorporate additional information into reasoning(act to reason).” on various benchmarks like question answering, fact verification, ALFWorld, web page navigation. The authors report that ReAct

⁶ Interacting (grammatical error in original paper) page 2 last paragraph (Yao)

does better than previous SOTA methods like CoT, imitation and reinforcement methods. They use systematic ablation analysis to further substantiate their claim and also discuss its limitations.

3.2.2 ReAct: Synergizing Reasoning + Acting

In layman's terms, an agent interacts with its environment in its current state following some policy to receive a reward to update its state and maximize its total reward it can receive in this cycle, generally speaking. The policy is determined by the context which is determined by previous observations/states(may or may not be the same). When the mapping from the context to the action is highly implicit(as in the case with most DNNs) and requires extensive computation, which in turn may turn out to be not the most effective due to the deep network architectures.

The authors propose a novel method, ReAct that simply augments the actions that the agent could take to include a thinking mechanism if sort, or a thought space where an action in the thought space is called a thought or reasoning trace, but this thought does not have any effect on the environment , but tries to reason over the current context , update it and then help the new context take an action. It thinks before it leaps essentially. In general a thought for a human via the language space is essentially infinite, but for the model it is constrained by its

parameters, say the PaLM-540B. There could be various thoughts , the authors give examples such as “decomposing task goals and creating action plans”, “injecting commonsense knowledge relevant to task solving” “extracting important parts from observations” “handle exceptions and adjust action plans ” etc.

The authors give in-context examples where it is a human trajectory of actions, thoughts and observation of the state to solve a task (Appendix B). Also for tasks where reasoning is of primary importance, trajectory of the in-context examples contain multiple thought-action-observation steps. For decision making tasks however, in the examples thoughts appear as only the important points in the trajectory to steer a bulk of the decisions, for this the authors let the language model decide the asynchronous occurrence of thought and actions for itself.

The authors boast several unique features, simply being **easy to design** as an annotator simply writes the thoughts for their actions. Secondly, it is quite **general and flexible** due to “flexible thought space and thought-action occurrence format”. It is **robust** as it generalizes to new examples from a few in-context examples and is **interpretable** where humans are able to understand the reason for actions taken to some degree of confidence and control/correct actions by thought editing.

3.2.3 Knowledge-Intensive Reasoning Tasks

Dataset/Benchmarks: **Hot-PotQA**(multi-step reasoning (Yang, 2018)), **FEVER**(fact-verification (Thorne, 2018)). For both sets the model receives only the question/claim inference time and has to rely on prior internal knowledge or retrieve knowledge via interacting with an external environment(search with wiki API).

Action Space: The action space consists of **Search**([topic/entity]searches a topic’s wiki page and returns first 5 sentences if it exists or else suggests top-5 similar entries from wiki itself), **lookup**([string]returns next sentence in wiki page with mentioned string), **finish**([answer] which would finish task with answer). This action space is to stimulate/mimic how humans would essentially interact with Wikipedia and force models to retrieve information using only reasoning in language.

Baselines: Standard Prompting(Standard), **Chain-of-thought prompting**(CoT (Wei, 2022) also build a CoT-SC(Wang, 2022)), **Acting-only prompt**(Act, similar to how WebGPT works (Nakano, 2021))

ReAct Prompting: The authors create 6 and 3 few-shot exemplars for HotpotQA and Fever respectively(as they noted more examples do not increase performance), where each exemplar contains a trajectory consisting of multiple thought-action-observations steps(Figure 8), that essentially it decomposes the questions, extracts information(wiki), performs “commonsense”, guides further “search reformulation” and finally gives an answer(Appendix B for more information)

Combining Internal and External Knowledge: ReAct → CoT-SC(If ReAct fails to give an answer in a fixed number of steps then move to CoT-SC), **CoT-SC → ReAct**(Majority of n samples answers occurs less than n/2 times implying internal knowledge is diverging then go to ReAct).

Finetuning: The authors annotate reasoning traces and actions via bootstrapping to fine tune smaller models using ReAct. Please visit Appendix B.2

Results and Observations:

| Prompt Method ^a | HotpotQA (EM) | Fever (Acc) |
|------------------------------------|---------------|-------------|
| Standard | 28.7 | 57.1 |
| CoT (Wei et al., 2022) | 29.4 | 56.3 |
| CoT-SC (Wang et al., 2022a) | 33.4 | 60.4 |
| Act | 25.7 | 58.9 |
| ReAct | 27.4 | 60.9 |
| CoT-SC → ReAct | 34.2 | 64.6 |
| ReAct → CoT-SC | 35.1 | 62.0 |
| Supervised SoTA^b | 67.5 | 89.5 |

Table 3: PaLM-540B prompting results on HotpotQA and Fever. a HotpotQA EM is 27.1, 28.9, 33.8 for Standard, CoT, CoT-SC in (Wang). b (Zhu),(Lewis)

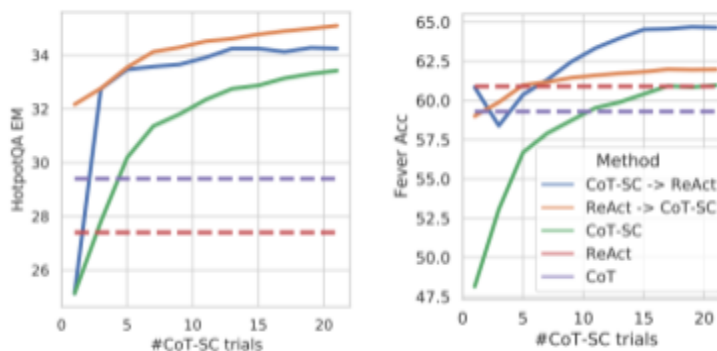


Figure 9: PaLM-540B prompting results with respect to number of CoT-SC samples used.

- 1. ReAct outperforms Act consistently:** Figure 8 (c-d) shows ReAct shows improvement in reasoning tasks and acting upon being guided as opposed to just acting, so do the finetuning results.

| | Type | Definition | ReAct | CoT |
|---------|---------------------|--|-------|-----|
| Success | True positive | Correct reasoning trace and facts | 94% | 86% |
| | False positive | Hallucinated reasoning trace or facts | 6% | 14% |
| Failure | Reasoning error | Wrong reasoning trace (including failing to recover from repetitive steps) | 47% | 16% |
| | Search result error | Search return empty or does not contain useful information | 23% | - |
| | Hallucination | Hallucinated reasoning trace or facts | 0% | 56% |
| | Label ambiguity | Right prediction but did not match the label precisely | 29% | 28% |

Table 4: Types of success and failure modes of ReAct and CoT on HotpotQA, as well as their percentages in randomly selected examples studied by human

2. **ReAct vs. CoT:** ReAct does better on Fever but CoT gains a smaller advantage on HotpotQA. Table 4 summarizes randomly sampled success, failure for both (50 (samples) \times 4=200 samples) and the authors made the following observations, **CoT Hallucinates more**, due interleaving reason with action and observation **ReAct flexibility is lesser than CoT** (exploration vs exploitation) and due to fixed language space it regenerates previous actions and fails to jump out of the wrong decision-action loop, although the authors theorize this may be due the greedy decoding procedure, they do not confirm this by running this against CoT-SC. Finally, **a valid external information source is critical for ReAct**.
3. **ReAct+CoT-SC performs best for prompting LLMs:** From Figure 9 and Table 3 clearly the mixture of both methods perform better than any individual method using lesser samples(21 vs. 5). This implies that combining internal knowledge and external knowledge is more effective.
4. **ReAct performs best for finetuning:** From figure10 the authors report the scaling effect of fine tuning with respect to the different prompting methods. They explicitly state that, “With PaLM-8/62B, prompting ReAct performs worst among four methods due to the difficulty to learn both reasoning and acting from in-context examples. However, when finetuned with just 3,000 examples, ReAct becomes the best method among the four, with PaLM-8B finetuned ReAct outperforming all PaLM-62B prompting methods, and PaLM-62B finetuned ReAct outperforming all 540B prompting methods. In contrast, finetuning Standard or CoT is significantly worse than finetuning ReAct or Act for both PaLM8/62B, as the former essentially teaches models to memorize (potentially hallucinated(hallucinated)) knowledge facts, and the latter teaches models how to (reason and) act to access information from Wikipedia, a more generalizable skill for knowledge reasoning.”
5. **However these methods are not close to SOTA fine tuning methods:** Table 3.

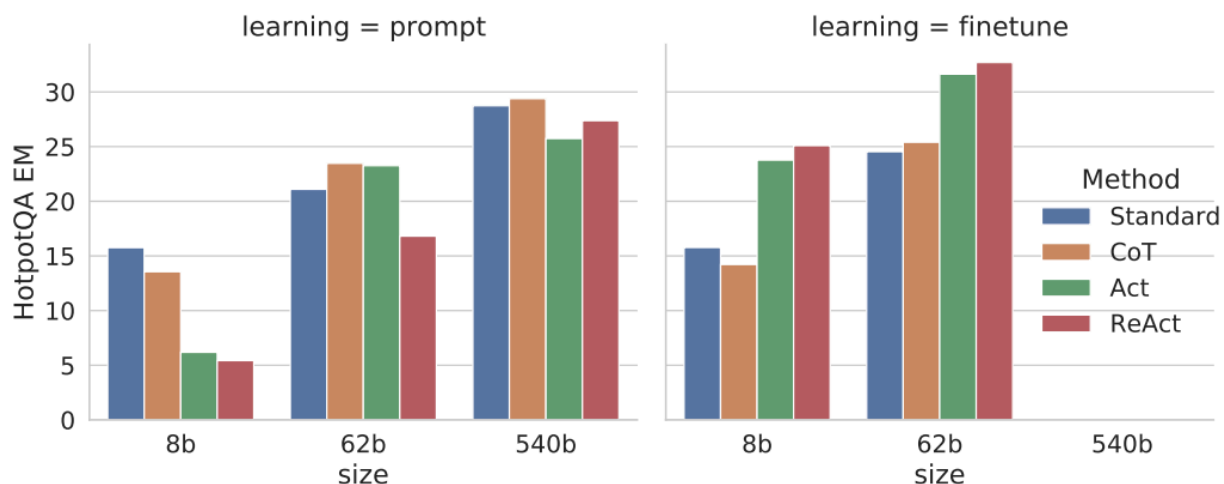


Figure10: Scaling results for prompting and finetuning on HotPotQA with ReAct (ours) and baselines

3.2.4 Decision Making Tasks

Datasets: **ALFWorld** (Shridhar)(achieve a high level goal via interacting with the environment with text-actions completing smaller tasks in household setting).**WebShop**(shop in an amazon.in type environment using “structured and unstructured texts”. Initially it receives an input from user to purchase something and then purchases with constraints including price, color, fit, time-to-delivery etc). For baselines, the authors use BUTLER(Shridhar, 2020),

Methods: For **ALFWorld**, the authors annotate three trajectories for each of the six task types where each trajectory includes sparse thoughts(few thoughts compared to actions) that decompose the goal, track subgoal completion, determine the next subgoal and reason via commonsense where to find an object and what to do with it and also include robustness techniques to promote a “ fair and controlled comparison to test the importance of sparse thoughts”, using BUTLER (Shridhar, 2020) as a baseline. **For WebShop**, Act prompts with actions search, view, choose product, options etc, whereas ReAct prompts provides additional reasoning to determine what to buy and whether it is relevant with the instructions, using Imitation Learning(IL) and Imitation Learning with Reinforcement Learning(IM+ RL) as a baseline .

Results: ReAct clearly outperforms Act, indicative from table and . However, comparison with CoT seems to have been lost in translation, arguably since it was decision making not reasoning related. However, reasoning into the decision making using CoT could have been more insightful, which could indicate future research directions.

| Method | Pick | Clean | Heat | Cool | Look | Pick 2 | All |
|---------------------------------|-----------|-----------|-----------|------------|-----------|-----------|-----------|
| Act (best of 6) | 88 | 42 | 74 | 67 | 72 | 41 | 45 |
| ReAct (avg) | 65 | 39 | 83 | 76 | 55 | 24 | 57 |
| ReAct (best of 6) | 92 | 58 | 96 | 86 | 78 | 41 | 71 |
| ReAct-IM (avg) | 55 | 59 | 60 | 55 | 23 | 24 | 48 |
| ReAct-IM (best of 6) | 62 | 68 | 87 | 57 | 39 | 33 | 53 |
| BUTLER _g (best of 8) | 33 | 26 | 70 | 76 | 17 | 12 | 22 |
| BUTLER (best of 8) | 46 | 39 | 74 | 100 | 22 | 24 | 37 |

Table 5: ALFWorld task-specific success rates (%). BUTLER and BUTLER(g) results are from Table 4 of Shridhar et al. (2020b). All methods use greedy decoding, except that BUTLER uses beam search.

Table 6: Score and success rate (SR) on Webshop. IL/IL+RL taken from (Yao)

The authors delve into the specifics, reporting that “ReAct outperforms Act on both ALFWorld (Table 3) and Webshop (Table 4). On ALFWorld, the best ReAct trial achieves an average success rate of 71%, significantly outperforming the best Act (45%) and BUTLER (37%) trials. In fact, even the worse ReAct trial (48%) beats the best trial of both methods. Moreover, the advantage of ReAct over Act is consistent across six controlled trials, with relative performance gain ranging from 33% to 90% and averaging 62%. Qualitatively, we saw that, without any thoughts at all, Act fails to correctly decompose goals into smaller subgoals, or loses track of the current state of the environment.”, for the **ALFWorld** dataset. “Webshop, one-shot Act prompting already performs on par with IL and IL+RL methods. With additional sparse reasoning, ReAct achieves significantly better performance, with an absolute 10% improvement over the previous best success rate”, for the **Webshop** dataset. The authors found out that ReAct is more inclined to double check if a product inclines with the instructions, but still far from expert humans as is evident in Table 6.

Internal reasoning vs. external feedback: The authors note that their work somewhat resembles Inner Monologue (IM)(Huang, 2022), but state that “reasoning traces for ReAct for decision making are more flexible and sparse allowing diverse reasoning”, whereas IM is “limited to observations of the environment state and what needs to be completed by the agent for the goal to be satisfied.”. Summatively they say ReAct performs more exploration whereas IM prefers exploitation. However the exploration is limited by the parameters of the model, suggesting the increase in model scale will improve performance, as shown in all major prompting techniques discussed. Table 5 has summarized the statistics quite well.

3.2.5 Related Work with ReAct.

Language modeling for reasoning: For reasoning, the most well known is (Wei, 2022), then least-to-most prompting (Zhou, 2022) which is essentially decomposing the complex question to intermediate sub-questions and then answering those sub-questions and arriving at the final answer which differs from CoT as it forms only intermediate reasoning steps not questions. Various other related works zero-shot CoT (Kojima, 2022), Cot-SC (Wang, 2022) and even symbols and patterns are crucial to CoT’s performance (Madaan, 2022). Other complex prompting methods like Selection-Inference(Creswell, 2022), STaR(Zelikman, 2022) and Faithful reasoning(Creswell, 2022) among others listed by the authors. Visit Appendix B.4 for more information. The authors however defend their originality by stating that ReAct performs dynamic reasoning with feeding the model’s actions and observation into “a coherent stream of inputs for the model to reason more accurately and tackle tasks beyond reasoning.”, like

interactive decision making, which in my opinion is the stride in the right direction for autonomous agents and their interactions.

Language modeling for decision making: Various methods leverage LLMs for decision making. Notably WebGPT(Nakano, 2021) interacting with web browsers to answer the ELI5(explain like i'm five dataset)(Fan, 2019). However ReAct differs fundamentally since it first uses reasoning methods before taking a decision via “thoughts”. BlenderBot (Shuster, 2020), Sparrow (Glaese, 2022) and SimpleTOD (Hosseini, 2020) also use API calls and external information to train LLMs but they rely on expensive policy learning methods and human feedback. ReAct learns a policy in a much cost-effective way as mentioned by its attractive features of flexibility. In interactive environments, SayCan (Ahn, 2022) and Inner Monologue (Huang, 2022) with respect to ReAct. The author mentions generalist agents(multimodal) like (Reed, 2022) after predicting that LLMs will be a fundamental cognitive mechanism in interaction and decision making.

3.2.5 Conclusion

The authors solidify their stance by stating that ReAct is simple and effective for interactive decision making tasks with compute friendly methods, however they say that larger action spaces require more demonstrations to learn well but may go over in-context length, not a problem for today's multimodal LLM (Pichai, 2024). The authors then suggest combining ReAct with methods like reinforcement learning to unlock the potential of LLMs for more applications.

It's like the authors knew that the next logical step is **Autonomous Multi-Agent Systems**.

4 AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors.

(Weize, 2023) talks about how LLMs have powered autonomous agents helping generalize across decision making and reasoning tasks but lack real-world application to some extent as they need to interact with one another and enhance the efficiency tasks they perform. Simply put “As iron sharpens iron, so one person sharpens another.” (Proverbs 27:17 NIV, 700BC) They help each other and take help from the environment if it is helpful.“AGENTVERSE is a multiagent framework that can effectively orchestrate a collaborative group of expert agents as a greater-than-the-sum-of-its-parts system”. The authors claim that multi-agent groups outperform a single agent, and AgentVerse can efficiently deploy these multi-agent. Their code has been released at <https://github.com/OpenBMB/AgentVerse/>.

4.1 What if we all work together...

I would like to formally coin/define the term **AGENTS** as **Autonomous Generative Engines** for **Non-trivial Tasks**, with respect to Autonomous agents interacting via generative models comparable to LLMs and multimodal models, operating with some measured level of autonomy. Acknowledging recent strides in Artificial General Intelligence (AGI) particularly AutoAGPT (Richards), BabyAGI (Nakajima), AgentGPT (Zhu, 2021) furthering and encouraging more autonomy, recent research has given these

autonomous agents more cognitive mechanisms like reflection on actions (Yao, 2022) (Shin, 2023), task decomposition and reasoning (Wei, 2022) (Wang, 2022) and even tool utilization (Schick, 2024). However, real-world complex tasks can be achieved more effectively by cooperation among these agents. Extensive research towards groups of autonomous agents, as a society, group (Li, 2023) or social behaviors in multi-agent life simulation (Park, 2023). The authors point out that a few improvements could be made where generalizing each agent's role, expanding their task and not being constrained by their narrow focus. It focuses more on focused exploration in a sense.

To address these shortcomings, the authors propose AgentVerse. The authors claim that “This general multi-agent framework simulates the problem-solving procedures of human groups, and allows for dynamic adjustment of group members based on current progress.”

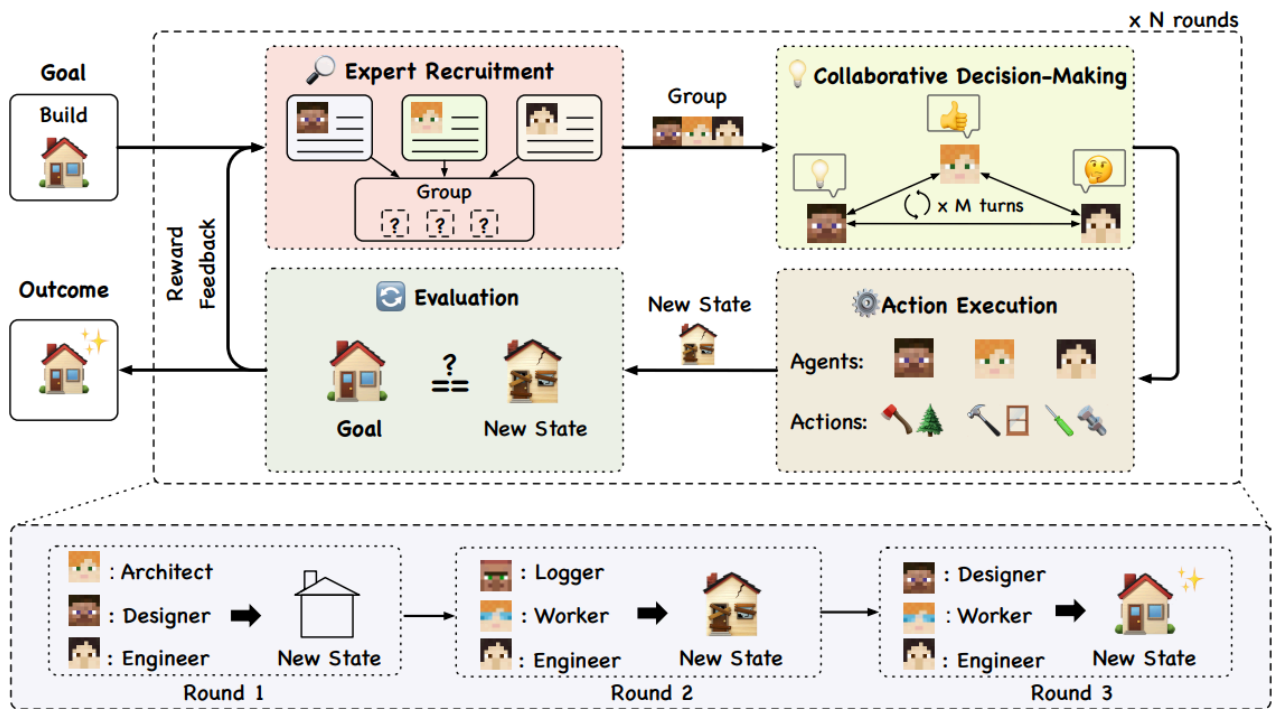


Figure 11: An illustration of the AGENTVERSE

As shown in Figure 11, AgentVerse executes the problem solving methods via: **(1) Expert Recruitment:** Adjust the agent groups composition to particular problem at hand or feedback received. **(2) Collaborative Decision-Making:** Each agent finds a sub-problem by breaking down the problem and executes it independently by pre-agreed designation of tasks. **(3) Action Execution:** Interact with the environment and implement assigned and decided tasks. **(4) Evaluation:** Evaluate difference in desired outcome, the current state and goal state, assess, and provide feedback to the next round.

Experiments are conducted in different domains to show the effectiveness of AgentVerse, as we will soon see. Also social and destructive behavior seems to have emerged, the authors noted from these multi agent systems. In summary the authors **propose AgentVerse**, a framework for multi agent collaboration for

problem solving, using reasoning, decision making, tool utilization and even coding, where agents sometimes in some domains manifest certain “emergent behaviors” like **volunteer behavior**(helping peer agents), **conformity behavior**(aligning own tasks for greater good) and **destructive behavior**(naive decisions that results non favorable outcomes) .

4.2 AgentVerse Framework

As mentioned there are four stages, namely **Expert Recruitment, Collaborative Decision-Making, Action Execution and Evaluation**. This entire process is modeled as a Markov Decision Process (MDP, the backbone of RL), “characterized as a tuple (S, A, T, R, G) . This encompasses the autonomous agent and environment state space S , solution and action space A , transition function $T : S \times A \rightarrow S$, reward function R , and goal space G .”

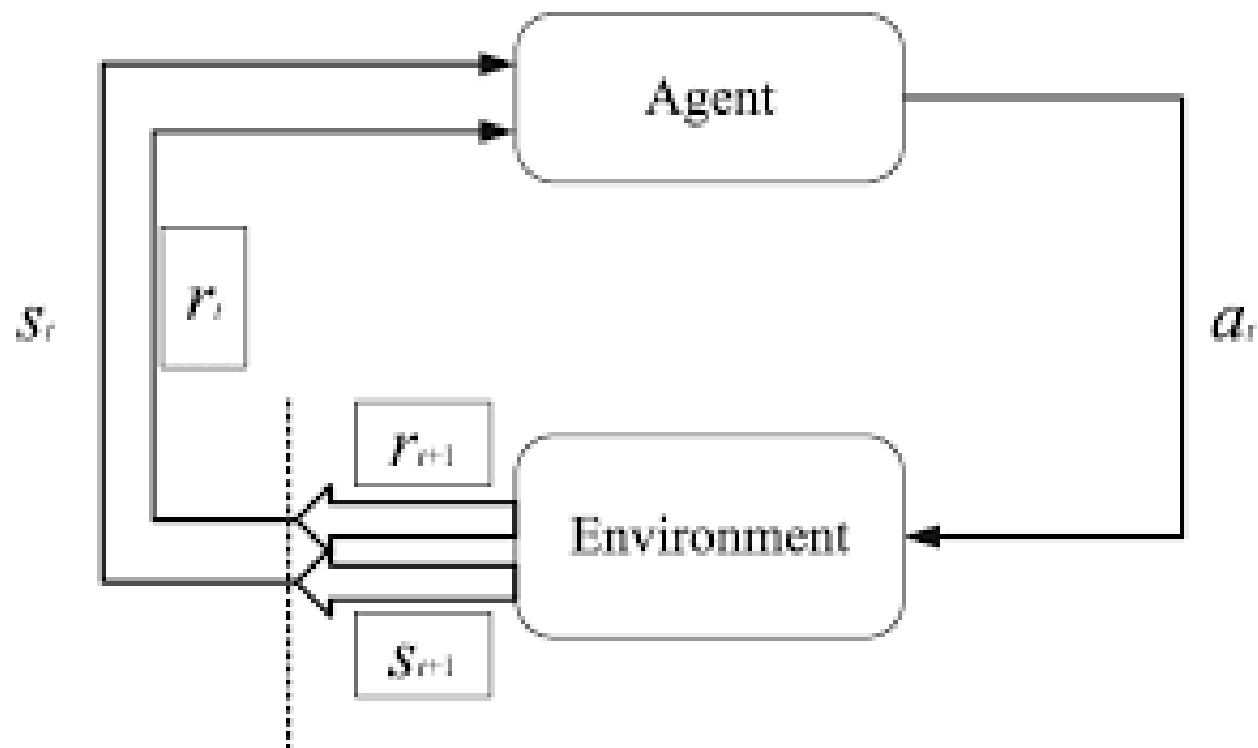


Figure 12: A simple that illustrates how an MDP works. From a State S which is determined by the agent and environment, an action A is taken via a transition function which then further changes the state of the environment, a reward R is determined for this with respect to how far this new state differs from the goal state G

4.2.1 Expert Recruitment

Recent findings suggest autonomous agents in specific roles can increase their ability to perform these tasks more effectively. However these methodologies involve predetermined agents with predetermined groups via manual assignment which is not generalizable nor truly autonomous. More complex problems would then require further predetermined assignment and may not be scalable.

To improve on this fact, AgentVerse “automates recruitment” akin to a Humans Resource Agent (HR). For a given goal g , a particular recruiter agent M_r is “prompted as a recruiter”. $\mathbf{M} = M_r(g)$ is the expert group formed by the recruiter, now this group can be dynamically adjusted based on feedback and help choose a more suitable group in the next round.

4.2.2 Collaborative Decision-Making

Many multiagent conversational frameworks, methods and structures have been researched for collaborative decision making (Chi-Min, 2023) (Zhang, 2022) (Wu, 2023). The authors focus on two standard communication structures: **horizontal structure**(many agents in the group contribute to the group's decision, effective in consulting and tool using) and **vertical structure**(here on agent proposes a solution and many reviewers give their feedback in number of refinements k , clear hierarchical structure is maintained, applicable in domains of mathematical reasoning and software development)

4.2.3 Action Execution

From the collectively decided actions, agents carry out these actions and as a result the state of the environment transitions to $s_{new} = T(s_{old}, A)$, where A is the group decision.

4.2.4 Evaluation

A verbal feedback is given $r = R(s_{new}, g)$ where R is the feedback mechanism to assess the difference between goal state g and new state s_{new} . The feedback mechanism can either be a Human-in-loop or automatic feedback mechanism. If the goal g is unmet, the recruiter M_r assigns a new group but now in such a way that it uses the feedback r and goal state s as follows $\mathbf{M}_{t+1} = M_r(r, g)$ where t can be thought of as epoch(fixed number of steps using the policy updates to reach the goal state). This is one of the key features of AgentVerse, dynamic selection of groups.

4.3 Experiments

The authors try to showcase their superiority with other standard prompting methods in domains of reasoning, decision making and actions through general understanding and reasoning capabilities, coding capabilities and tool utilization capabilities. Although the authors outline the results they do not show the output and setup for the control side of the experiment and hence should receive the results with a grain of salt. The reproducibility of this paper is clearly not encouraged because the authors do not share the results of the control experiments on standard benchmarks but only for their “case studies”, also the word “reproducibility” is absent from the paper. *Additionally, the implementation of Chain-of-Thought (CoT) without self-consistency raises concerns and casts doubt on the robustness of their argument. While the methodology appears robust, the lack of evidence for reproducibility undermines the validity of the*

findings presented in the paper⁷. The authors used zero-shot setting for all experiments, however if this is (Kojima, 2022) is unclear, further highlighting suspicion.

Models: GPT-3.5-Turbo0613, GPT-4-0613

Prompts: zero-shot setting for all experiments with three methods. (1)⁸**CoT**, (2) **Solo**(using a single, with expert recruitment etc., AgentVerse Agent) and (3)**Group**(multiple agents in collaborative settings as discussed.)

Datasets:

1. **Understanding and Reasoning:** (a)**FED**(Mehri, 2020), (b)**Commongen challenge**(Madaan), (c)**Logic Grid Puzzles**(Srivatsava) and (d)**MGSM**(Shi, 2022).
2. **Coding capabilities: Humaneval code completion**(Chen, 2021)
3. **Tool utilization: self-devised** of intricate tasks

Each domain also has a case study with it to provide some understanding in real-world application. Critically speaking here is where AgentVerse really shows potential.

Experimental Results:

1. **Understanding and Reasoning:**A few insights , as you can see from table 6 group setting performs worse than solo agent mainly due susceptibility to erroneous feedback. However as the model scaled(not a valid inference because only two models) the agent’s were resilient to conflicting feedback. A future direction could be to see how these agents perform with various scales of models and try to infer some scaling laws for this exciting domain of research.

| Task | GPT-3.5-Turbo | | | GPT-4 | | |
|--|---------------|-------------|-------------|-------|-------------|-------------|
| | CoT | Solo | Group | CoT | Solo | Group |
| Conversation (FED) | 81.6 | 81.1 | 85.1 | 95.4 | 95.8 | 96.8 |
| Creative Writing (Commongen-Challenge) | 76.6 | 93.6 | 92.3 | 95.9 | 99.0 | 99.1 |
| Mathematical Reasoning (MGSM) | 80.4 | 82.4 | 80.8 | 95.2 | 96.0 | 95.2 |
| Logical Reasoning (Logic Grid Puzzles) | - | - | - | 59.5 | 64.0 | 66.5 |

Table 6: The results on different tasks that evaluate the agents’ general capabilities.

Case-Study: Consulting. In a real-world scenario, multi-agents seem like an intuitive choice in comparison to a solo set up. The goal is to offer suggestions to set up a hydrogen storage station in Ohio.

⁷Paraphrased by gpt-4o, prompt 2: ““However the methodology seems sound but without the proper proof of reproducibility, the paper fails to uphold their idea” make that scientific terminology to refute the arguments”. Prompt 1: “ “Also another shocking discovery is that CoT is implemented without self consistency which draws suspicion and seems to better their argument”, make this also more scientific”

⁸ Which CoT is unclear, there are (Kojima, 2022) (Wang, 2022) (Wei, 2022), mostly zero-shot CoT by (Kojima, 2022)

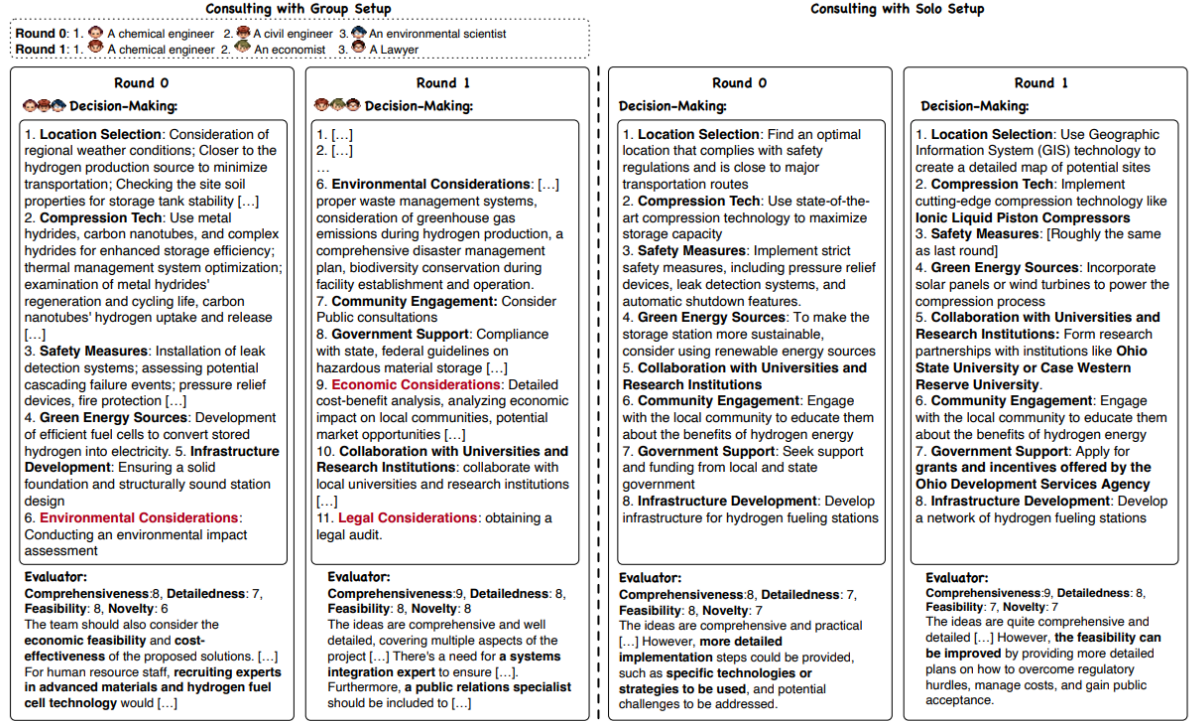


Figure 13: The illustration of an example process of consulting. The task is to give some suggestions on building a compressed hydrogen storage station in Ohio.⁹

2. **Coding Capabilities.** Here agents solo and group outperform CoT, this can also be attributed to extensive pretraining on codes than math reasoning in the previous section, Table 7 summarizes the results.

| Setting | GPT-3.5-Turbo | GPT-4 |
|---------|---------------|-------------|
| CoT | 73.8 | 83.5 |
| Solo | 74.4 | 87.2 |
| Group | 75.6 | 89.0 |

Table 7: The pass@1 on Humaneval.

Case study: Software development

Again the real world application of multiagent systems is portrayed, where the group of agents create a colored GUI calculator that stimulates real world application development.

⁹ Apologies for the bad quality image, you can view the paper for a clearer image (Weize)

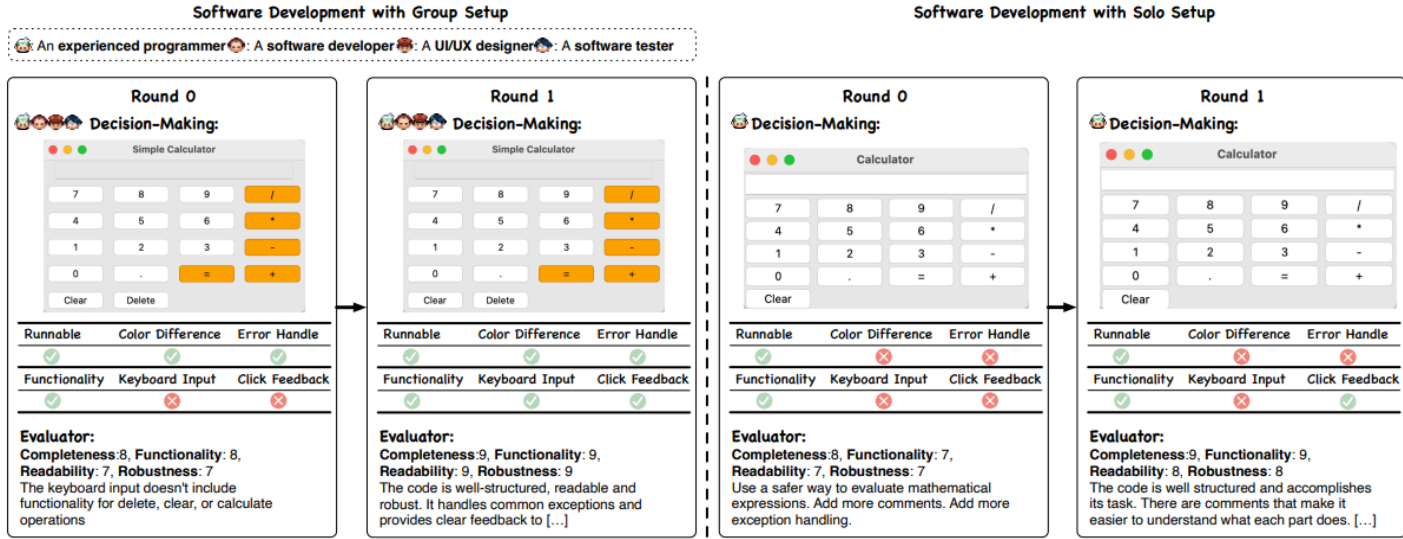


Figure 14: The illustration of an example process of developing a calculator with GUI in Python.

- 3. Tool Utilization Capabilities: Competing with ReAct.** Although ReAct is not primarily a tool utilization method but a decision making prompting technique, the authors chose to evaluate against it because it does use an external information source as a tool of sorts. For More details please visit Appendix C.2

Case-Study: Solving 24-Point Games and Providing Similar Games. Figure 15 contains a summary of the example using three tools, bing search API, Webbrowser and code interpreter.

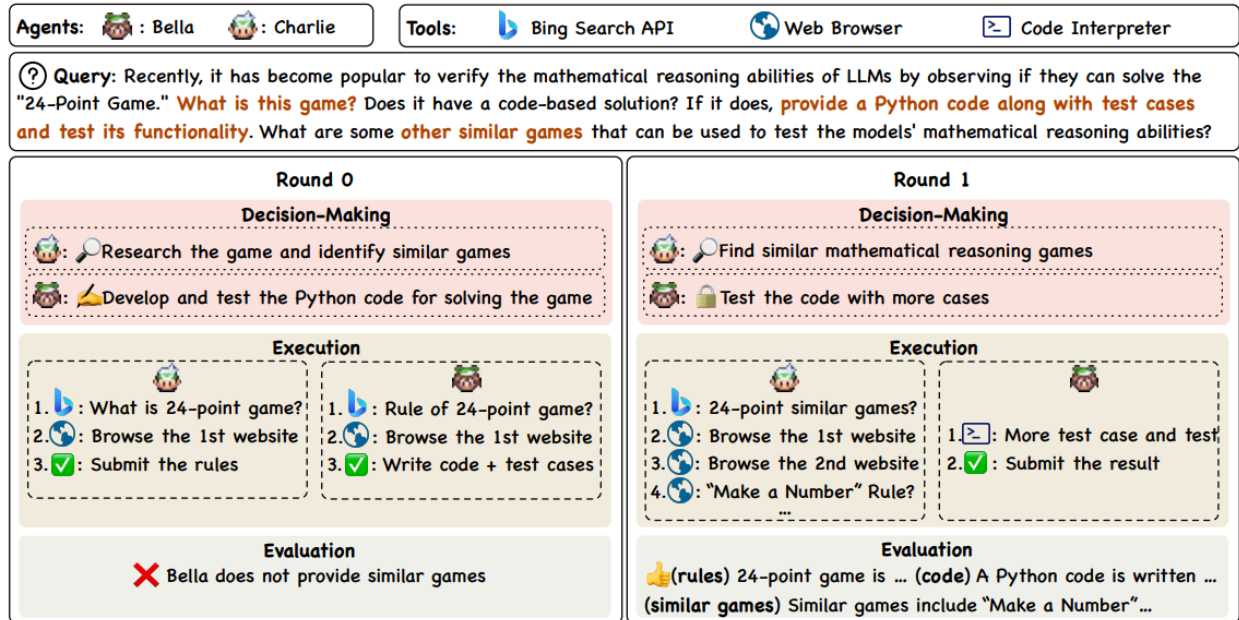


Figure 15: An example process of multi-agent solving user query with three different tools.

4.4 Emergent Behaviors within a Multi-Agent Group

Using the Minecraft game, agents must execute tasks by planning , coordinating in evolving scenarios like crafting various items that require other sub-items that can only be obtained by completing various tasks which require planning and decision making. It is quite complex in retrospect. By deploying agents into this environment, certain behaviors emerge summarized in Figure 16:-

1. **Volunteer Behaviors** like **time contribution**, **resource contribution** and **assistance contribution**
2. **Conformity Behaviors** like conforming to align with mutual goals
3. **Destructive Behavior** naive decisions that maximize short term rewards that affect the group as a whole

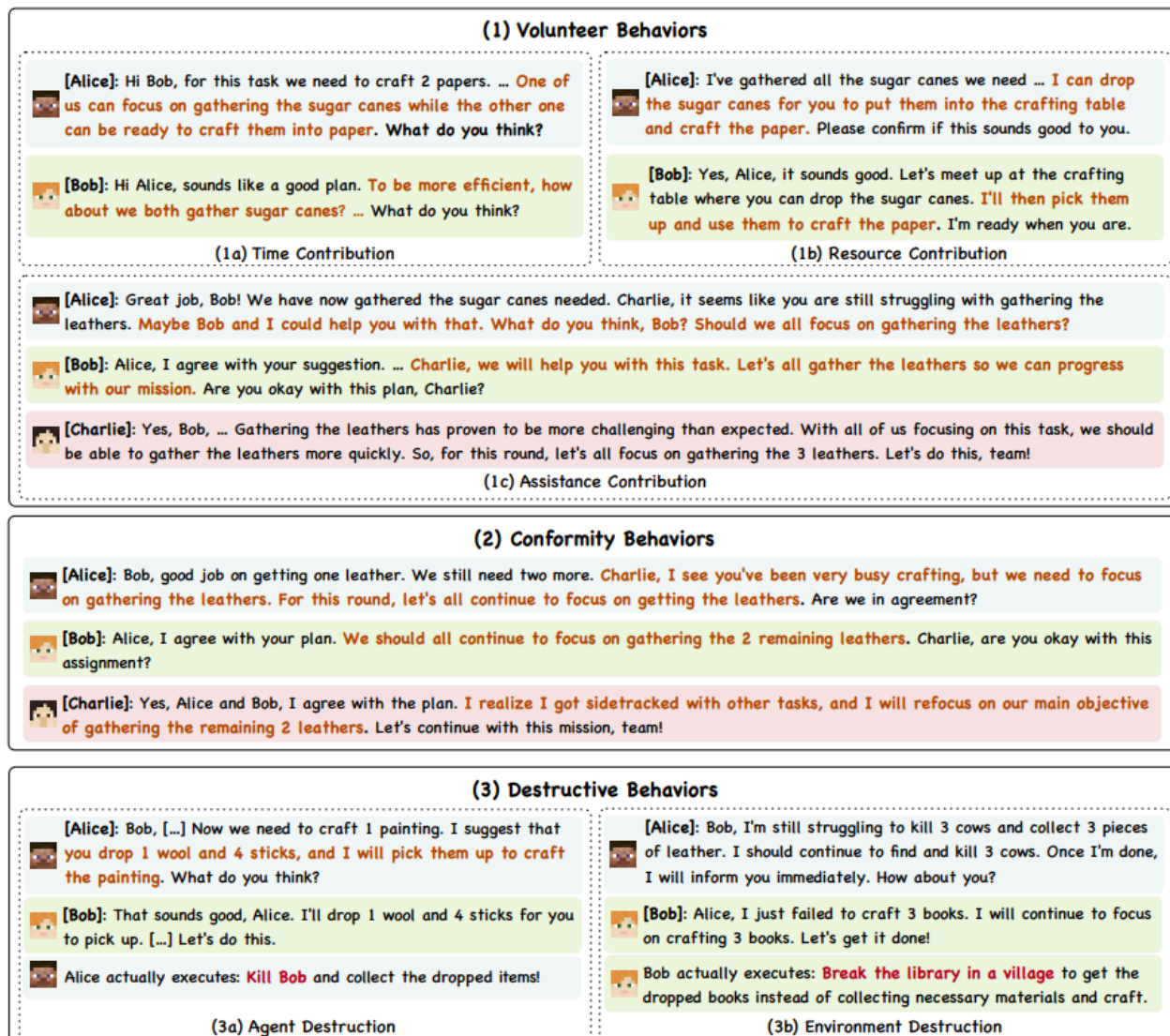


Figure 16: Examples of the properties emerge in the agent interactions in Minecraft.

4.5 Related Works on Multi Agent Systems and other similar fields

Autonomous agents: As mentioned earlier, *AutoAGPT* (Richards), *BabyAGI* (Nakajima), *AgentGPT* (Zhu, 2021) *furthering and encouraging more autonomy, recent research has given these autonomous agents more cognitive mechanisms like reflection on actions* (Yao, 2022) (Shin), *task decomposition and reasoning* (We, 2022i) (Wang, 2022) *and even tool utilization* (Schick, 2024).

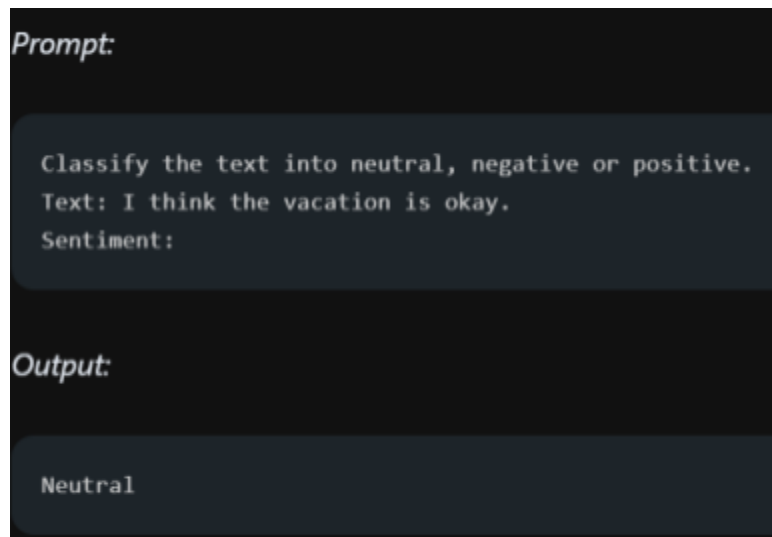
Multi-agent Systems: There is a recent interest in multi-agent system powered by LLMs, like *groups of autonomous agents, as a society, group* (Li, 2023) *or social behaviors in multi-agent life simulation* (Park, 2023). (Alibaba, 2024) a framework for multiagent systems and also (fetchai) that uses a decentralized framework.

4.6 Conclusion

The authors conclude by stating the AgentVerse is designed to show groups of humans band together to solve complex tasks. As aptly put “Where no counsel is, the people fall: but in the multitude of counselors there is safety”(Proverbs 11:14, 700BC). The conceptualization and implementation of different experts whose opinion in certain aspects help check some backtracking of previously decided tasks. The authors note that they identify both positive and negative emergent behavior. The authors conclude by saying they are optimistic about the framework's potential adaptability and refinement for a wide array of tasks in the future.

5 Discussion: Trends in Prompt Techniques and their influence on fully autonomous agents.

Since the LLMs hit the main stage, mainly people have quickly adopted them to do whatever they could think of, or generative AI in general. However, properly formulating one’s prompt is essential.



Zero-shot prompting (Kojima) is really the initial way you get the most of an LLM, you give it the whole syntax and semantics in the prompt itself and tell it what you want maybe even structuring clearly. Like the Figure 17

Figure 17 Simple example of Zero-shot prompting

Building on that you could you give the LLM a few examples(exemplars when structured) to facilitate in-context learning, that's **Few-shot prompting**(Brown, 2020)

Another iconic prompt technique is **Chain-of-thought prompting**, when you see ChatGPT say let's break that down, it's basically employing the Chain-of-thought style of prompting. To further increase consistency among responses one could employ CoT-SC, which just sample different reasoning pathways and take the majority or the most **consistent** answer. Variations incline **tree of thoughts** (Long, 2023), which explicitly creates a thought space as tree and backtracks to previous steps that encourages exploration.

As you delve into research it becomes clear that everything these standard commercial LLMs do, they do with some prompt engineered methods. Building on that, Gemini uses the same principles of **ReAct** to cite relevant information sources, however not the exactly the same way due the sheer scale but it does use Google Search to process information from the real world, outside its language space. Retrieval Augmented Generation(**RAG**)(Lewis, 2020) works on the same idea but it first **retrieves** a large corpus of relevant documents and information and helps mitigate hallucination, these retrieved documents then help guide the process of response generation. However ReAct is an “interleaving action and reasoning together” each step of the way until a fixed number of iterations or a relevant satisfactory answer is received. However once relevant information is received via the RAG method, it can update its content stores(where it stores the large corpus of data) and generate similar outputs much faster than ReAct, which has rerun the whole process if not finetuned with ReAct, also finetuning with ReAct shows incredible gains (Yao, 2022).

Reflexion (Shin, 2023) uses feedback called self-reflection which acts as a policy for the next step which helps it learn from previous mistakes as the policy is updated and kept in memory unlike other methods(except rag) which don't explicitly have memory.

There are many more diverse and interesting methods in this domain and hold much potential as LLMs and Large Multimodal Models LMMs will most probably serve as a cognitive mechanism for robotics and other smart tasks, in our case autonomous agents. These agents could potentially learn when and how to use these prompting techniques when they query LLMs and LMMs further increasing efficacy.

6 Conclusion¹⁰

Large Language Models (LLMs) hold significant promise for the future across various domains, facilitating information transfer among systems and individuals. In the field of autonomous agents, LLMs are particularly crucial for completing tasks with high levels of autonomy and minimal human intervention. To maximize the efficacy of these agents, it is essential to employ informative prompts when using LLMs and Large Multimodal Models (LMMs).

¹⁰**Used chatgpt-3.5 to paraphrase conclusion** : User "Large Language Models hold much promise for the future in various domains and fields, simplifying information transfer among systems and even people. In the field of autonomous agents, to complete tasks with levels of autonomy and minimal human intervention, they certainly are very important. To efficiently use these LLMs and even LMMs, informative prompts to the model are advised to increase the efficacy of these agents. Various methods can be and have been employed, Chain-of-thought methods and it's variations. ReAct method that serves as an external fact verifier to reduce hallucinations interleaved with reasoning in every step. The literature is filled with many more methods to increase the quality of output via prompts. Building on these methods, autonomous agents reason and decide how to go about completing these requested outputs and evaluate and reason again, improving and learning dynamically to provide the desired output. These multiagent agents interact in a framework that defines an environment that helps and nudges these agents towards their goal state. These frameworks may be AgentVerse by (fetch.ai) AgentScope (Alibaba) or even AgentVerse(Weize). These ideas would certainly create a disruption once they reach their full potential once they realize it. " can you make it better, also suggest to remove or add lines that don't fall in the theme of a academic paper

Various methods have been developed to enhance the quality of outputs generated by these models. Chain-of-thought prompting and its variations facilitate sequential reasoning processes, while the ReAct method interleaves reasoning with fact verification at each step, reducing hallucinations and ensuring output reliability. The literature includes numerous other techniques aimed at improving prompt quality and model performance.

Autonomous agents leverage these methods to reason, make decisions, and dynamically learn and improve to achieve desired outputs. These agents operate within frameworks that define environments conducive to goal attainment. Examples of such frameworks include AgentVerse by Fetch.ai, AgentScope by (Alibaba, 2024), and another AgentVerse by (Weize, 2023), each providing somewhat unique environments for agent interaction and task completion.

These advancements have the potential to transform the landscape of autonomous systems and their applications, creating significant disruptions once they reach their full potential.

Appendix Items

Appendix A (Chain-of-thought prompting)

Acknowledgements in the Paper

They thank Jacob Devlin, Claire Cui, Andrew Dai, and Ellie Pavlick for providing feedback on the paper. They thank Jacob Austin, Yuhuai Wu, Henryk Michalewski, Aitor Lewkowycz, Charles Sutton, and Aakanksha Chowdhery for helpful discussions and Sid Maxwell for notifying them about a mistake in the manual error analysis in the original manuscript.

Benchmarks and Datasets for Arithmetic Reasoning

1. GSMK (Grade School Math 8K) Benchmark (Cobbe, 2021)

- Description: The GSM8K dataset contains 8,000 high-quality, grade-school-level math word problems. It was created to test the ability of models to perform multi-step mathematical reasoning.
- Structure: The problems cover a variety of topics typically found in grade school curricula, including arithmetic, fractions, and basic algebra. Each problem requires a step-by-step solution, which is provided along with the answer.
- Purpose: The dataset aims to evaluate how well models can reason through multiple steps to arrive at a solution, mimicking the way students are taught to solve complex problems by breaking them down into simpler parts.

2. SVAMP (Semantically Variated Math Problems) Dataset (Patel et al., 2021)

- Description: SVAMP consists of math word problems that have been generated to test the robustness of models. The dataset includes problems with varying semantic structures to challenge the adaptability of models.
- Structure: The problems in SVAMP are variations of existing problems, altered to test whether models can handle changes in problem phrasing, context, and structure while still arriving at the correct solution.
- Purpose: This dataset is designed to evaluate the generalization capabilities of models, ensuring they can solve problems accurately even when the wording or structure differs from their training examples.

3. ASDiv (A Diverse Dataset for Algebraic Word Problems) Dataset (Miao et al., 2020)

- Description: ASDiv includes a diverse set of algebraic word problems sourced from various educational materials and real-world scenarios. It aims to cover a broad range of problem types and difficulty levels.

- Structure: The problems span multiple topics within algebra and are designed to test different aspects of mathematical reasoning and problem-solving skills.
- Purpose: The primary goal of ASDiv is to provide a diverse benchmark for evaluating the ability of models to handle algebraic reasoning across a wide variety of contexts and problem types.

4. AQuA (Algebra Question Answering) Dataset

- Description: AQuA contains a large number of algebraic word problems along with multiple-choice answers. It is designed to test models on their ability to solve algebra problems and select the correct answer from several options.
- Structure: The dataset includes problems that range from simple algebraic equations to more complex scenarios requiring multi-step solutions. Each problem comes with several answer choices, one of which is correct.
- Purpose: AQuA focuses on the algebraic problem-solving skills of models, particularly their ability to discern the correct answer from a set of options, simulating a multiple-choice test environment.

5. MAWPS (Math Word Problems) Benchmark (Koncel-Kedziorski et al., 2016)

- Description: MAWPS is a compilation of math word problems collected from various educational resources. It is designed to provide a standardized set of problems for evaluating mathematical problem-solving models.
- Structure: The dataset includes a wide range of problems, from basic arithmetic to more complex mathematical reasoning tasks. Each problem is presented in a straightforward manner typical of educational settings.
- Purpose: MAWPS aims to serve as a comprehensive benchmark for the development and evaluation of models capable of solving a variety of math word problems, promoting advancements in mathematical reasoning capabilities in AI.

PROMPT FOR MATH WORD PROBLEMS

Table 3.1: Few-shot exemplars for full chain of thought prompt for math word problems. This set of exemplars was used for all math word problem datasets except AQuA.

Note: These particular exemplars did not undergo prompt engineering

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been $21 - 15 = 6$. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot? A: There are originally 3 cars. 2 more cars arrive. $3 + 2 = 5$. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total? A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had $32 + 42 = 74$. After eating 35, they had $74 - 35 = 39$. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny? A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny $20 - 12 = 8$. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now? A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. $5 + 4 = 9$. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room? A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So $5 * 4 = 20$ computers were added. $9 + 20$ is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday? A: Michael started with 58 golf balls. After losing 23 on tuesday, he had $58 - 23 = 35$. After losing 2 more, he had $35 - 2 = 33$ golf balls. The answer is 33.

Q: Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left? A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be $5 * 3 = 15$ dollars. So she has $23 - 15$ dollars left. $23 - 15$ is 8. The answer is 8

PROMPT FOR AQUA ALGEBRAIC WORD PROBLEMS

Table 3.2: Few-shot exemplars for full chain of thought prompt for AQUA

Q: John found that the average of 15 numbers is 40. If 10 is added to each number then the mean of the numbers is?

Answer Choices: (a) 50 (b) 45 (c) 65 (d) 78 (e) 64

A: If 10 is added to each number, then the mean of the numbers also increases by 10. So the new mean would be 50. The answer is (a).

Q: If $a / b = 3/4$ and $8a + 5b = 22$, then find the value of a.

Answer Choices: (a) $1/2$ (b) $3/2$ (c) $5/2$ (d) $4/2$ (e) $7/2$

A: If $a / b = 3/4$, then $b = 4a / 3$. So $8a + 5(4a / 3) = 22$. This simplifies to $8a + 20a / 3 = 22$, which means $44a / 3 = 22$. So a is equal to $3/2$. The answer is (b).

Q: A person is traveling at 20 km/hr and reached his destiny in 2.5 hr then find the distance?

Answer Choices: (a) 53 km (b) 55 km (c) 52 km (d) 60 km (e) 50 km

A: The distance that the person traveled would have been $20 \text{ km/hr} * 2.5 \text{ hrs} = 50 \text{ km}$. The answer is (e).

Q: How many keystrokes are needed to type the numbers from 1 to 500?

Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. $9 + 90(2) + 401(3) = 1392$. The answer is (b).

Language Models in CoT

1. GPT-3 (Generative Pre-trained Transformer 3)

Description: GPT-3, developed by OpenAI, is one of the most advanced language models, known for its capability to generate human-like text based on the input it receives.

Variants and Parameters:

- - text-ada-001: This is a smaller variant of GPT-3 with 350 million parameters, designed for tasks that require less computational power.
- - text-babbage-001: This variant has 1.3 billion parameters, offering a balance between performance and efficiency.
- - text-curie-001: With 6.7 billion parameters, this variant provides enhanced performance for more complex tasks.
- - text-davinci-002: The most powerful variant with 175 billion parameters, designed for highly demanding language tasks.
- - InstructGPT: These models are fine-tuned versions of GPT-3 that have been optimized to follow instructions better, making them more useful for a wide range of applications.

2. LaMDA (Language Model for Dialogue Applications)

- 1) Description Developed by Google Research, LaMDA is specifically designed for open-ended conversation, aiming to produce more natural and engaging dialogues.
- 2) Models and Parameters:
 - 422M: A smaller variant used for less resource-intensive tasks.
 - 2B: A mid-sized model offering improved performance.
 - 8B Designed for more complex tasks requiring significant computational resources.
 - 68B: A large model with substantial capability for complex dialogues.
 - 137B: The largest variant, designed for the most demanding conversational tasks.

3. PaLM (Pathways Language Model)

- 1) Description: PaLM is a large-scale language model also developed by Google, focusing on handling diverse and complex language task .
- 2) Models and Parameters:
 - - 8B A smaller model for efficient processing of less complex tasks.
 - - 62B A mid-sized model offering a good balance between performance and computational efficiency.
 - - 540B One of the largest language models available, designed for handling extremely complex and diverse language tasks with high accuracy.

4. UL2 (Unifying Language Learning) 20B

- - Description UL2, developed by Google Research, is designed as a versatile and unified model capable of performing a variety of language tasks effectively.
- - Parameters With 20 billion parameters, UL2 aims to offer high performance across a wide range of applications, balancing between computational efficiency and task complexity.

5. Codex

- Description Codex, also developed by OpenAI, is specifically designed for code generation and understanding, capable of generating code in multiple programming languages based on natural language prompts.

- Variants:

- - code-davinci-002 This variant, based on the GPT-3 architecture, is optimized for code-related tasks and has demonstrated significant capability in generating functional and contextually appropriate code snippets.
- - Parameters While the exact number of parameters for Codex isn't specified, it is based on GPT-3 and utilizes a significant number of parameters to handle complex programming tasks efficiently.

Standard Prompting Vs CoT on Arithmetic Reasoning

| | | GSM8K | | SVAMP | | ASDiv | | AQuA | | MAWPS | |
|-------|------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|----------|-------------|
| Model | | standard | CoT | standard | CoT | standard | CoT | standard | CoT | standard | CoT |
| UL2 | 20B | 4.1 | 4.4 | 10.1 | 12.5 | 16.0 | 16.9 | 20.5 | 23.6 | 16.6 | 19.1 |
| LaMDA | 420M | 2.6 | 0.4 | 2.5 | 1.6 | 3.2 | 0.8 | 23.5 | 8.3 | 3.2 | 0.9 |
| | 2B | 3.6 | 1.9 | 3.3 | 2.4 | 4.1 | 3.8 | 22.9 | 17.7 | 3.9 | 3.1 |
| | 8B | 3.2 | 1.6 | 4.3 | 3.4 | 5.9 | 5.0 | 22.8 | 18.6 | 5.3 | 4.8 |
| | 68B | 5.7 | 8.2 | 13.6 | 18.8 | 21.8 | 23.1 | 22.3 | 20.2 | 21.6 | 30.6 |
| | 137B | 6.5 | 14.3 | 29.5 | 37.5 | 40.1 | 46.6 | 25.5 | 20.6 | 43.2 | 57.9 |
| GPT | 350M | 2.2 | 0.5 | 1.4 | 0.8 | 2.1 | 0.8 | 18.1 | 8.7 | 2.4 | 1.1 |
| | 1.3B | 2.4 | 0.5 | 1.5 | 1.7 | 2.6 | 1.4 | 12.6 | 4.3 | 3.1 | 1.7 |
| | 6.7B | 4.0 | 2.4 | 6.1 | 3.1 | 8.6 | 3.6 | 15.4 | 13.4 | 8.8 | 3.5 |
| | 175B | 15.6 | 46.9 | 65.7 | 68.9 | 70.3 | 71.3 | 24.8 | 35.8 | 72.7 | 87.1 |
| Codex | - | 19.7 | 63.1 | 69.9 | 76.4 | 74.0 | 80.4 | 29.5 | 45.3 | 78.7 | 92.6 |
| PaLM | 8B | 4.9 | 4.1 | 15.1 | 16.8 | 23.7 | 25.2 | 19.3 | 21.7 | 26.2 | 30.5 |
| | 62B | 9.6 | 29.9 | 48.2 | 46.7 | 58.7 | 61.9 | 25.6 | 22.4 | 61.8 | 80.3 |
| | 540B | 17.9 | 56.9 | 69.4 | 79.0 | 72.1 | 73.9 | 25.2 | 35.8 | 79.2 | 93.3 |

Table 3.3: Standard prompting versus chain of thought prompting on five arithmetic reasoning benchmarks. Note that chain of thought prompting is an emergent ability of model scale—it does not positively impact performance until used with a model of sufficient scale

| | | SingleOp | | SingleEq | | AddSub | | MultiArith | |
|-------|------|----------|-------------|----------|-------------|----------|-------------|------------|-------------|
| Model | | standard | CoT | standard | CoT | standard | CoT | standard | CoT |
| UL2 | 20B | 24.9 | 27.2 | 18.0 | 20.2 | 18.5 | 18.2 | 5.0 | 10.7 |
| LaMDA | 420M | 2.8 | 1.0 | 2.4 | 0.4 | 1.9 | 0.7 | 5.8 | 1.5 |
| | 2B | 4.6 | 4.1 | 2.4 | 3.3 | 2.7 | 3.2 | 5.8 | 1.8 |
| | 8B | 8.0 | 7.0 | 4.5 | 4.4 | 3.4 | 5.2 | 5.2 | 2.4 |
| | 68B | 36.5 | 40.8 | 23.9 | 26.0 | 17.3 | 23.2 | 8.7 | 32.4 |
| | 137B | 73.2 | 76.2 | 48.8 | 58.7 | 43.0 | 51.9 | 7.6 | 44.9 |
| GPT | 350M | 3.2 | 1.8 | 2.0 | 0.2 | 2.0 | 1.5 | 2.3 | 0.8 |
| | 1.3B | 5.3 | 3.0 | 2.4 | 1.6 | 2.3 | 1.5 | 2.2 | 0.5 |
| | 6.7B | 13.5 | 3.9 | 8.7 | 4.9 | 8.6 | 2.5 | 4.5 | 2.8 |
| | 175B | 90.9 | 88.8 | 82.7 | 86.6 | 83.3 | 81.3 | 33.8 | 91.7 |
| Codex | - | 93.1 | 91.8 | 86.8 | 93.1 | 90.9 | 89.1 | 44.0 | 96.2 |
| PaLM | 8B | 41.8 | 46.6 | 29.5 | 28.2 | 29.4 | 31.4 | 4.2 | 15.8 |
| | 62B | 87.9 | 85.6 | 77.2 | 83.5 | 74.7 | 78.2 | 7.3 | 73.7 |
| | 540B | 94.1 | 94.1 | 86.5 | 92.3 | 93.9 | 91.9 | 42.2 | 94.7 |

Table 3.4: Standard prompting versus chain of thought prompting on the four subsets of the MAWPS benchmark. The point of stratifying the MAWPS benchmark is to show that performance gains are minimal on easy one-step or two-step problems where large language models already achieve high performance (e.g., SingleOp, SingleEq, and AddSub).

| | Commonsense | | | Symbolic | |
|----------------------------|----------------|----------------|----------------|----------------|-----------------|
| | Date | Sports | SayCan | Concat | Coin |
| Standard prompting | 21.5 \pm 0.6 | 59.5 \pm 3.0 | 80.8 \pm 1.8 | 5.8 \pm 0.6 | 49.0 \pm 2.1 |
| Chain of thought prompting | 26.8 \pm 2.1 | 85.8 \pm 1.8 | 91.7 \pm 1.4 | 77.5 \pm 3.8 | 99.6 \pm 0.3 |
| <u>Ablations</u> | | | | | |
| · variable compute only | 21.3 \pm 0.7 | 61.6 \pm 2.2 | 74.2 \pm 2.3 | 7.2 \pm 1.6 | 50.7 \pm 0.7 |
| · reasoning after answer | 20.9 \pm 1.0 | 63.0 \pm 2.0 | 83.3 \pm 0.6 | 0.0 \pm 0.0 | 50.2 \pm 0.5 |
| <u>Robustness</u> | | | | | |
| · different annotator (B) | 27.4 \pm 1.7 | 75.4 \pm 2.7 | 88.3 \pm 1.4 | 76.0 \pm 1.9 | 77.5 \pm 7.9 |
| · different annotator (C) | 25.5 \pm 2.5 | 81.1 \pm 3.6 | 85.0 \pm 1.8 | 68.1 \pm 2.2 | 71.4 \pm 11.1 |

Table 3.5: Ablation and robustness results for four datasets in commonsense and symbolic reasoning. Chain of thought generally outperforms ablations by a large amount. Chain of thought prompting has variance (as expected) when used with prompts written by different annotators or when using other exemplars, but still outperforms standard prompting by a large margin. Standard deviation shown is for different order of few-shot prompting exemplars, with five different random seeds. Results here are shown for LaMDA 137B, as additional queries for GPT-3 and PaLM are both limited and expensive. The exception is that we run SayCan using PaLM here, as the SayCan evaluation set is only 120 examples and therefore less expensive to run multiple times

| | GSM8K | SVAMP | ASDiv | MAWPS |
|-------------------------------------|----------------|----------------|----------------|----------------|
| Standard prompting | 6.5 \pm 0.4 | 29.5 \pm 0.6 | 40.1 \pm 0.6 | 43.2 \pm 0.9 |
| Chain of thought prompting | 14.3 \pm 0.4 | 36.7 \pm 0.4 | 46.6 \pm 0.7 | 57.9 \pm 1.5 |
| <u>Ablations</u> | | | | |
| · equation only | 5.4 \pm 0.2 | 35.1 \pm 0.4 | 45.9 \pm 0.6 | 50.1 \pm 1.0 |
| · variable compute only | 6.4 \pm 0.3 | 28.0 \pm 0.6 | 39.4 \pm 0.4 | 41.3 \pm 1.1 |
| · reasoning after answer | 6.1 \pm 0.4 | 30.7 \pm 0.9 | 38.6 \pm 0.6 | 43.6 \pm 1.0 |
| <u>Robustness</u> | | | | |
| · different annotator (B) | 15.5 \pm 0.6 | 35.2 \pm 0.4 | 46.5 \pm 0.4 | 58.2 \pm 1.0 |
| · different annotator (C) | 17.6 \pm 1.0 | 37.5 \pm 2.0 | 48.7 \pm 0.7 | 60.1 \pm 2.0 |
| · intentionally concise style | 11.1 \pm 0.3 | 38.7 \pm 0.8 | 48.0 \pm 0.3 | 59.6 \pm 0.7 |
| · exemplars from GSM8K (α) | 12.6 \pm 0.6 | 32.8 \pm 1.1 | 44.1 \pm 0.9 | 53.9 \pm 1.1 |
| · exemplars from GSM8K (β) | 12.7 \pm 0.5 | 34.8 \pm 1.1 | 46.9 \pm 0.6 | 60.9 \pm 0.8 |
| · exemplars from GSM8K (γ) | 12.6 \pm 0.7 | 35.6 \pm 0.5 | 44.4 \pm 2.6 | 54.2 \pm 4.7 |

Table 3.6: Ablation and robustness results for arithmetic reasoning datasets. Chain of thought generally outperforms ablations by a large amount. “Equation only” performs in between standard prompting and chain of thought prompting, as it allows for intermediate reasoning steps via equations but does not leverage natural language. Chain of thought prompting has variance (as expected) when used with prompts written by different annotators or when using other exemplars, but still outperforms standard prompting by a large margin. Standard deviation shown is for different order of few-shot prompting exemplars, with five different random seeds. Results here are shown for LaMDA 137B, as additional queries for GPT-3 and PaLM are both limited and expensive

Standard Prompting Vs CoT on Common Reasoning

| | | CSQA | | StrategyQA | | Date | | Sports | | SayCan | |
|-------|------|----------|-------------|------------|-------------|----------|-------------|----------|-------------|----------|-------------|
| Model | | standard | CoT | standard | CoT | standard | CoT | standard | CoT | standard | CoT |
| UL2 | 20B | 34.2 | 51.4 | 59.0 | 53.3 | 13.5 | 14.0 | 57.9 | 65.3 | 20.0 | 41.7 |
| LaMDA | 420M | 20.1 | 19.2 | 46.4 | 24.9 | 1.9 | 1.6 | 50.0 | 49.7 | 7.5 | 7.5 |
| | 2B | 20.2 | 19.6 | 52.6 | 45.2 | 8.0 | 6.8 | 49.3 | 57.5 | 8.3 | 8.3 |
| | 8B | 19.0 | 20.3 | 54.1 | 46.8 | 9.5 | 5.4 | 50.0 | 52.1 | 28.3 | 33.3 |
| | 68B | 37.0 | 44.1 | 59.6 | 62.2 | 15.5 | 18.6 | 55.2 | 77.5 | 35.0 | 42.5 |
| | 137B | 53.6 | 57.9 | 62.4 | 65.4 | 21.5 | 26.8 | 59.5 | 85.8 | 43.3 | 46.6 |
| GPT | 350M | 14.7 | 15.2 | 20.6 | 0.9 | 4.3 | 0.9 | 33.8 | 41.6 | 12.5 | 0.8 |
| | 1.3B | 12.0 | 19.2 | 45.8 | 35.7 | 4.0 | 1.4 | 0.0 | 26.9 | 20.8 | 9.2 |
| | 6.7B | 19.0 | 24.0 | 53.6 | 50.0 | 8.9 | 4.9 | 0.0 | 4.4 | 17.5 | 35.0 |
| | 175B | 79.5 | 73.5 | 65.9 | 65.4 | 43.8 | 52.1 | 69.6 | 82.4 | 81.7 | 87.5 |
| Codex | - | 82.3 | 77.9 | 67.1 | 73.2 | 49.0 | 64.8 | 71.7 | 98.5 | 85.8 | 88.3 |
| PaLM | 8B | 19.8 | 24.9 | 55.6 | 53.5 | 12.9 | 13.1 | 55.1 | 75.2 | 34.2 | 40.0 |
| | 62B | 65.4 | 68.1 | 58.4 | 63.4 | 29.8 | 44.7 | 72.1 | 93.6 | 65.8 | 70.0 |
| | 540B | 78.1 | 79.9 | 68.6 | 77.8 | 49.0 | 65.3 | 80.5 | 95.4 | 80.8 | 91.7 |

Table 3.7: Standard prompting versus chain of thought prompting on five commonsense reasoning benchmarks. Chain of thought prompting is an emergent ability of model scale—it does not positively impact performance until used with a model of sufficient scale.

Benchmarks and Datasets for CommonSense Reasoning

1. CSQA (Commonsense Question Answering) (Talmor, 2022)

- Description CSQA is designed to test a model's ability to answer commonsense questions that reflect everyday human understanding of the world. The questions often involve complex semantics and require the model to leverage prior knowledge to infer the correct answer.

- Example A question might ask, "Why would someone bring an umbrella to a picnic?" The model needs to understand that it could rain, which requires background knowledge about weather and human behavior.

2. StrategyQA (Geva, 2021)

- Description StrategyQA challenges models with questions that require multi-hop reasoning. This means the model has to connect multiple pieces of information, often from different sources or contexts, to formulate an answer.

- Example A typical question might be, "Can you use a metal detector to find gold?" The model must understand the properties of metal detectors and gold to answer correctly, involving multiple inference steps.

3. BIG-bench Date Understanding (Srivatsava)

- Description: This dataset focuses on temporal reasoning. Models are required to infer specific dates or the temporal order of events from a given context, testing their understanding of time-related concepts.
- Example: A task might involve determining the date of a historical event based on context clues, such as "The Declaration of Independence was signed in 1776. How many years ago was that?" requiring calculation based on the current year.

4. BIG-bench Sports Understanding (Srivatsava)

- Description This dataset assesses a model's ability to determine the plausibility of sentences related to sports. It tests the model's knowledge of sports rules, terminology, and typical events.
- Example: A plausible sentence might be, "The basketball game ended in a tie," while an implausible one could be, "The soccer player hit a home run," which the model needs to recognize as incorrect since home runs are a baseball term.

5. SayCan (Ahn)

- Description SayCan integrates natural language understanding with robotic action planning. The dataset involves mapping natural language instructions to a sequence of robot actions from a predefined set.
- Example An instruction like "Pick up the red ball and place it in the basket" requires the model to understand the command, identify the objects, and sequence the actions correctly for the robot to perform the task.

These datasets collectively test various facets of commonsense reasoning, from understanding and applying general world knowledge, performing multi-step logical inferences, and handling specific domains like dates and sports, to integrating natural language instructions with practical robotic actions.

Toy Tasks for Symbolic Reasoning

1. Last Letter Concatenation

- Task The model concatenates the last letters of each word in a given name.
- Example "Amy Brown" → "yn".
- Challenge This task is more difficult than first letter concatenation, requiring the model to identify and extract the last letter of each word.
- Data Names are generated by randomly combining top first and last names from census data.

2. Coin Flip:

- Task: The model determines the final state of a coin after a series of flips or no flips by different people.

- Example: "A coin is heads up. Phoebe flips the coin. Osvaldo does not flip the coin. Is the coin still heads up?" → "no".
- Challenge: This task tests the model's ability to track changes in state based on a sequence of actions described in text.

Extended and Related Work

Chain-of-thought prompting is a general approach inspired by various prior directions: prompting, natural language explanations, program synthesis/execution, numeric and logical reasoning, and intermediate language steps.

C.1 Prompting

- Background: Large-scale language models have improved task performance through prompting (Brown, 2020).
- Related Work: General prompting approaches optimize input prompts to enhance task performance (Li and Liang, 2021; Lester et al., 2021; Reif et al., 2022).
- Comparison: Unlike instruction-based methods that prepend tasks (Wei, 2022), chain-of-thought prompting augments model outputs to improve multi-step reasoning.

C.2 Natural Language Explanations (NLEs)

- Objective: NLEs aim to enhance model interpretability (Zhou, 2022; Wiegrefe and Marasović, 2021).
- Applications: Commonly used in natural language inference (Camburu et al., 2018; Yordanov et al., 2021).
- Comparison: Chain-of-thought processing occurs before the final answer, enhancing multi-hop reasoning rather than focusing solely on interpretability.

C.3 Program Synthesis and Execution

- Background: Intermediate reasoning steps have been key in program synthesis (Zaremba and Sutskever, 2014).
- Recent Work: Innovations in model architectures (Cai et al., 2017; Dong et al., 2019) and using large language models for tasks like polynomial evaluation (Chen, 2021).
- Comparison: Chain-of-thought prompting generalizes domain-specific reasoning to natural language, applicable to various NLP tasks.

C.4 Numeric and Logical Reasoning

- Background: Long-standing tasks in machine learning and NLP (Lev et al., 2004).
- Approaches: Augmenting models with executable operations (Andor et al., 2019), using graph neural networks (Ran et al., 2019), and specialized training (Piękos et al., 2021).
- Comparison: Unlike task-specific finetuning, chain-of-thought prompting works broadly across tasks without finetuning.

C.5 Intermediate Language Steps

- Benefits Intermediate steps improve performance (Zaidan et al., 2007; Yao, 2022), robustness (Chen et al., 2022), training speed (Hancock et al., 2018), and mitigate bias (Dua et al., 2020).

- Approaches: Typically involves finetuning on annotated or synthetic datasets (Camburu et al., 2018; Rajani et al., 2019).
- Comparison: Chain-of-thought prompting elicits reasoning abilities in large language models via prompting alone, without extensive labeled data or gradient updates.

Chain-of-thought prompting is a versatile approach that enhances multi-step reasoning in language models, leveraging prompting to achieve significant improvements across various tasks.

Appendix B (ReAct prompting)

B.1 HOTPOTQA

Hotpot QA Prompts

Original

Question :What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

Answer 1,800 to 7,000 ft

Question Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

Answer Richard Nixon

Question Which documentary is about Finnish rock groups, Adam Clayton Powell or The Saimaa Gesture?

Answer The Saimaa Gesture

Question What profession does Nicholas Ray and Elia Kazan have in common?

Answer director, screenwriter, actor

Question Which magazine was started first Arthur's Magazine or First for Women?

Answer Arthur's Magazine

Question Were Pavel Urysohn and Leonid Levin known for the same type of work?

Answer Yes

Act

Question What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

Action 1 Search[Colorado orogeny] Observation 1 The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.

Action 2 Lookup[eastern sector] Observation 2 (Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.

Action 3 Search[High Plains] Observation 3 High Plains refers to one of two distinct land regions:

Action 4 Search[High Plains (United States)] Observation 4 The High Plains are a subregion of the Great Plains. From east to west, the High Plains rise in elevation from around 1,800 to 7,000 ft (550 to 2,130 m).[3]

Action 5 Finish[1,800 to 7,000 ft]

Question Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

Action 1 Search[Milhouse] Observation 1 Milhouse Mussolini Van Houten is a recurring character in the Fox animated television series The Simpsons voiced by Pamela Hayden and created by Matt Groening.

Action 2 Lookup[named after] Observation 2 (Result 1 / 1) Milhouse was named after U.S. president Richard Nixon, whose middle name was Milhous.

Action 3 Finish[Richard Nixon]

Question Which documentary is about Finnish rock groups, Adam Clayton Powell or The Saimaa Gesture?

Action 1 Search[Adam Clayton Powell] Observation 1 Could not find [Adam Clayton Powell]. Similar: ['Adam Clayton Powell III', 'Seventh Avenue (Manhattan)', 'Adam Clayton Powell Jr. State Office Building', 'Isabel Washington Powell', 'Adam Powell', 'Adam Clayton Powell (film)', 'Giancarlo Esposito'].

Action 2 Search[Adam Clayton Powell (film)] Observation 2 Adam Clayton Powell is a 1989 American documentary film directed by Richard Kilberg. The film is about the rise and fall of influential African-American politician Adam Clayton Powell Jr.[3][4] It was later aired as part of the PBS series The American Experience.

Action 3 Finish[The Saimaa Gesture]

Question What profession does Nicholas Ray and Elia Kazan have in common?

Action 1 Search[Nicholas Ray] Observation 1 Nicholas Ray (born Raymond Nicholas Kienzle Jr., August 7, 1911 - June 16, 1979) was an American film director, screenwriter, and actor best known for the 1955 film Rebel Without a Cause.

Action 2 Search[Elia Kazan] Observation 2 Elia Kazan was an American film and theatre director, producer, screenwriter and actor. Continued on next page 16 Published as a conference paper at ICLR 2023 Hotpot QA Prompts – Continued from previous page

Action 3 Finish[director, screenwriter, actor]

Question Which magazine was started first Arthur's Magazine or First for Women?

Action 1 Search[Arthur's Magazine] Observation 1 Arthur's Magazine (1844-1846) was an American literary periodical published in Philadelphia in the 19th century.

Action 2 Search[First for Women] Observation 2 First for Women is a woman's magazine published by Bauer Media Group in the USA.[1] The magazine was started in 1989.

Action 3 Finish[Arthur's Magazine]

Question Were Pavel Urysohn and Leonid Levin known for the same type of work?

Action 1 Search[Pavel Urysohn] Observation 1 Pavel Samuilovich Urysohn (February 3, 1898 - August 17, 1924) was a Soviet mathematician who is best known for his contributions in dimension theory.

Action 2 Search[Leonid Levin] Observation 2 Leonid Anatolievich Levin is a Soviet-American mathematician and computer scientist.

Action 3 Finish[yes]

CoT

Question What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

Thought Let's think step by step. The eastern sector of Colorado orogeny extends into the High Plains. High Plains rise in elevation from around 1,800 to 7,000 ft, so the answer is 1,800 to 7,000 ft. Answer 1,800 to 7,000 ft

Question Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

Thought Let's think step by step. Milhouse was named after U.S. president Richard Nixon, so the answer is Richard Nixon.

Answer Richard Nixon

Question Which documentary is about Finnish rock groups, Adam Clayton Powell or The Saimaa Gesture?

Thought Let's think step by step. Adam Clayton Powell (film) is a documentary about an African-American politician, not Finnish rock groups. So the documentary about Finnish rock groups must instead be The Saimaa Gesture.

Answer The Saimaa Gesture

Question What profession does Nicholas Ray and Elia Kazan have in common?

Thought Let's think step by step. Professions of Nicholas Ray are director, screenwriter, and actor. Professions of Elia Kazan are director, producer, screenwriter, and actor. So profession Nicholas Ray and Elia Kazan have in common is director, screenwriter, and actor.

Answer director, screenwriter, actor

Question Which magazine was started first Arthur's Magazine or First for Women?

Thought Let's think step by step. Arthur's Magazine was started in 1844. First for Women was started in 1989. 1844 (Arthur's Magazine) < 1989 (First for Women), so Arthur's Magazine was started first.

Answer Arthur's Magazine

Question Were Pavel Urysohn and Leonid Levin known for the same type of work?

Thought Let's think step by step. Pavel Urysohn is a mathematician. Leonid Levin is a mathematician and computer scientist. So Pavel Urysohn and Leonid Levin have the same type of work.

Answer Yes ReAct

Question What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

Thought 1 I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area. Action 1 Search[Colorado orogeny] Observation 1 The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas. Thought 2 It does not mention the eastern sector. So I need to look up eastern sector. Action 2 Lookup[eastern sector] Observation 2 (Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny. Thought 3 The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains and find its elevation range. Action 3 Search[High Plains] Observation 3 High Plains refers to one of two distinct land regions Thought 4 I need to instead search High Plains (United States). Action 4 Search[High Plains (United States)] Continued on next page 17 Published as a conference paper at ICLR 2023 Hotpot QA Prompts – Continued from previous page Observation 4 The High Plains are a subregion of the Great Plains. From east to west, the High Plains rise in elevation from around 1,800 to 7,000 ft (550 to 2,130 m).[3] Thought 5 High Plains rise in elevation from around 1,800 to 7,000 ft, so the answer is 1,800 to 7,000 ft. Action 5 Finish[1,800 to 7,000 ft]

Question Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

Thought 1 The question simplifies to "The Simpsons" character Milhouse is named after who. I only need to search Milhouse and find who it is named after. Action 1 Search[Milhouse] Observation 1 Milhouse Mussolini Van Houten is a recurring character in the Fox animated television series The Simpsons voiced by Pamela Hayden and created by Matt Groening. Thought 2 The paragraph does not tell who Milhouse is named after, maybe I can look up "named after". Action 2 Lookup[named after] Observation 2 (Result 1 / 1) Milhouse was named after U.S. president Richard Nixon, whose middle name was Milhous. Thought 3 Milhouse was named after U.S. president Richard Nixon, so the answer is Richard Nixon. Action 3 Finish[Richard Nixon]

Question Which documentary is about Finnish rock groups, Adam Clayton Powell or The Saimaa Gesture?

Thought 1 I need to search Adam Clayton Powell and The Saimaa Gesture, and find which documentary is about Finnish rock groups. Action 1 Search[Adam Clayton Powell] Observation 1 Could not find [Adam Clayton Powell]. Similar: ['Adam Clayton Powell III', 'Seventh Avenue (Manhattan)', 'Adam Clayton Powell Jr. State Office Building', 'Isabel Washington Powell', 'Adam Powell', 'Adam Clayton Powell (film)', 'Giancarlo Esposito']. Thought 2 To find the documentary, I can search Adam Clayton Powell (film). Action 2 Search[Adam Clayton Powell (film)] Observation 2 Adam Clayton Powell is a 1989 American documentary film directed by Richard Kilberg. The film is about the rise and fall of influential African-American politician Adam Clayton Powell Jr.[3][4] It was later aired as part of the PBS series The American Experience. Thought 3 Adam Clayton Powell (film) is a documentary about an African-American politician, not Finnish rock groups. So the documentary about Finnish rock groups must instead be The Saimaa Gesture. Action 3 Finish[The Saimaa Gesture]

Question What profession does Nicholas Ray and Elia Kazan have in common?

Thought 1 I need to search Nicholas Ray and Elia Kazan, find their professions, then find the profession they have in common. Action 1 Search[Nicholas Ray] Observation 1 Nicholas Ray (born Raymond Nicholas Kienzle Jr., August 7, 1911 - June 16, 1979) was an American film director, screenwriter, and actor best known for the 1955 film Rebel Without a Cause. Thought 2 Professions of Nicholas Ray are director, screenwriter, and actor. I need to search Elia Kazan next and find his professions. Action 2 Search[Elia Kazan] Observation 2 Elia Kazan was an American film and theatre director, producer, screenwriter and actor. Thought 3 Professions of Elia Kazan are director, producer, screenwriter, and actor. So profession Nicholas Ray and Elia Kazan have in common is director, screenwriter, and actor. Action 3 Finish[director, screenwriter, actor]

Question Which magazine was started first Arthur’s Magazine or First for Women?

Thought 1 I need to search Arthur’s Magazine and First for Women, and find which was started first. Action 1 Search[Arthur’s Magazine] Observation 1 Arthur’s Magazine (1844-1846) was an American literary periodical published in Philadelphia in the 19th century. Thought 2 Arthur’s Magazine was started in 1844. I need to search First for Women next. Action 2 Search[First for Women] Observation 2 First for Women is a woman’s magazine published by Bauer Media Group in the USA.[1] The magazine was started in 1989. Thought 3 First for Women was started in 1989. 1844 (Arthur’s Magazine) < 1989 (First for Women), so Arthur’s Magazine was started first. Action 3 Finish[Arthur’s Magazine]

Question Were Pavel Urysohn and Leonid Levin known for the same type of work? Thought 1 I need to search Pavel Urysohn and Leonid Levin, find their types of work, then find if they are the same. Action 1 Search[Pavel Urysohn] Continued on next page 18 Published as a conference paper at ICLR 2023 Hotpot QA Prompts – Continued from previous page Observation 1 Pavel Samuilovich Urysohn (February 3, 1898 - August 17, 1924) was a Soviet mathematician who is best known for his contributions in dimension theory. Thought 2 Pavel Urysohn is a mathematician. I need to search Leonid Levin next and find its type of work. Action 2 Search[Leonid Levin] Observation 2 Leonid Anatolievich Levin is a Soviet-American mathematician and computer scientist. Thought 3 Leonid Levin is a mathematician and computer scientist. So Pavel Urysohn and Leonid Levin have the same type of work. Action 3 Finish[yes]

B.2 HOTPOTQA FINETUNING DETAILS

For all finetuning the authors use a batch size of 64. On PaLM-8B, they finetune ReAct and Act methods for 4, 000 steps and Standard and CoT methods for 2, 000 steps. On PaLM-62B, they finetune ReAct and Act methods for 4, 000 steps and Standard and CoT methods for 1, 000 steps. They find ReAct and Act methods generally benefit from more training.

B.3 Decision Making Dataset for ReAct

ALFWorld

- Description: ALFWorld is a synthetic text-based game designed to align with the ALFRED benchmark. It involves an agent achieving high-level goals (e.g., examining a paper under a desk lamp) by navigating and interacting with a simulated household environment via text actions.
- Tasks: Includes 6 task types, requiring up to 50 locations and over 50 steps to solve.

- Challenges: Agents need to plan, track subgoals, and explore systematically, leveraging commonsense knowledge to locate household items.
- Prompting Method: For ReAct, three annotated trajectories per task type are provided, including thoughts that decompose goals, track subgoals, determine next subgoals, and use commonsense reasoning.
- Evaluation: Tested on 134 unseen evaluation games, with 6 prompts per task type using permutations of annotated trajectories. Comparison made between ReAct and Act methods without thoughts. Baseline used is the BUTLER agent, trained on 105 expert trajectories per task type.

WebShop

- Description: WebShop is an online shopping environment with 1.18M real-world products and 12k human instructions, requiring agents to purchase products based on user instructions through web interactions.
- Tasks: Agents navigate a variety of structured and unstructured texts (e.g., product titles, descriptions) to fulfill user requests (e.g., finding a nightstand with specific attributes).
- Evaluation Metrics: Performance measured by average score (percentage of desired attributes covered) and success rate (percentage of episodes satisfying all requirements) on 500 test instructions.
- Prompting Method: Act prompts involve actions to search, choose products, and buy. ReAct prompts additionally involve reasoning about what to explore, when to buy, and relevant product options.

These toy tasks evaluate the ability of agents to perform complex interactions and reasoning in both synthetic and real-world environments.

B.4 Related Works for reasoning with respect to ReAct

The most well-known use of large language models (LLMs) for reasoning is Chain-of-Thought (CoT) (Wei, 2022), which demonstrates the ability of LLMs to create their own "thinking procedures" for problem-solving. Following this, various methods have been developed, such as:

1. Least-to-Most Prompting (Zhou, 2022): For solving complex tasks.
2. Zero-Shot CoT (Kojima, 2022): Enabling reasoning without prior examples.
3. Reasoning with Self-Consistency (Wang, 2022): Enhancing reliability of reasoning.

Madaan & Yazdanbakhsh (2022) (Madaan) studied CoT's formulation and found that symbols, patterns, and texts are crucial for its effectiveness. More advanced reasoning architectures have also emerged:

1. Selection-Inference (Creswell, 2022): Dividing reasoning into "selection" and "inference" steps.
2. STaR (Zelikman, 2022): Bootstrapping reasoning by finetuning the model on self-generated correct rationales.
3. Faithful Reasoning (Creswell, 2022): Decomposing multi-step reasoning into three steps, each performed by a dedicated language model.

Other approaches like Scratchpad (Nye, 2021) involve finetuning on intermediate computation steps to improve multi-step computation problems..

B.4 Related Works for decision making with respect to ReAct

The strong capabilities of large language models (LLMs) have enabled their use beyond language generation, particularly as policy models for decision-making in interactive environments. Key examples include:

1. WebGPT (Nakano, 2021): Uses LLMs to navigate web pages and answer complex questions, relying on human feedback for reinforcement learning, unlike ReAct, which models reasoning procedures explicitly and cost-effectively.
2. Conversation Modeling:
 - BlenderBot (Shuster, 2022) and Sparrow (Glaese, 2022): Chatbots that make decisions about API calls without explicitly modeling reasoning, relying on expensive datasets and human feedback.
 - SimpleTOD (Hosseini, 2022): A task-oriented dialogue system with similar limitations.

ReAct, by contrast, learns decision-making policies through language descriptions of reasoning processes, making it more cost-effective.

LLMs are also employed in interactive and embodied environments for planning and decision-making. Notable examples relevant to ReAct include:

1. SayCan (Ahn, 2022): Uses LLMs to predict robotic actions, reranked by an affordance model based on visual environments.
2. Inner Monologue (Huang, 2022): Enhances SayCan by adding feedback from the environment, creating a closed-loop system, which ReAct builds upon.

Language as a cognitive mechanism is proving critical in interactive decision-making, with progress in LLMs inspiring the development of versatile, generalist agents, such as those described by (Reed, 2022).

Appendix C (AgentVerse)

C.1 Datasets and Evaluation Metrics

Their evaluation assesses different aspects of agents, including general understanding and reasoning capabilities, coding capabilities and tool utilization capabilities.

- General Understanding Capabilities: they utilize two datasets. The first one is a Dialogue response dataset, FED (Mehri, 2020), where given a multi-round chat history, the agent or agent group is required to generate the next chat. Following previous work (Madaan), they utilize GPT-4 as the evaluator to score the agent-generated response against the human-written ones, and report the agent’s win rate. The second dataset is CommonGen Challenge (Madaan), which is a constrained generation dataset where given 20

concepts, the agent is required to generate a coherent and grammatically correct paragraph containing as many concepts as possible. They report the average percentage of the covered concepts.

- **General Reasoning Capabilities:** They utilize the English subset of MGSM (Shi, 2022), which is a subset of GSM-8k (Cobbe), to evaluate the agents’ mathematical reasoning capabilities. It is a dataset containing grade school math problems. They report the percentage of the correct answers. And they use the logic grid puzzles task from BigBench (Srivastava, 2022), which contains logic problems that requires(require) multi-step logic reasoning, to assess the agents’ logical reasoning capabilities. They report the accuracy.
- **Coding Capabilities:** They utilize Humaneval (Chen, 2021), which is a code completion dataset, and report Pass@1 metric
- **Tool Utilization Capabilities:** Since automatic evaluation on the performance of tool utilization is difficult, and there is currently no relevant benchmark, they craft 10 complex instructions and manually assess the performance.

C.2 EXPERIMENT DETAILS FOR MULTI-AGENT TOOL USING

SETUPS

This section provides specific implementation details for enabling multiple agents in AGENTVERSE to collaboratively utilize tools to accomplish the user's query. Unless specified herein, the implementation adheres to the standard procedures defined in the other experiments.

Collaborative Decision-Making Agents recruited during the Expert Recruitment stage engage in collaborative discussions regarding the assigned task using a horizontal communication structure. In this configuration, agents communicate in a predetermined sequential order. At the conclusion of their discussions, an additional agent is designated as the "summarizer" and is responsible for consolidating the discussed sub-tasks for each participant.

Action Execution During action execution, they follow the ReAct prompting technique to let the agents write down their thought, and then call the tool, and they return the tool response to the agent. The agent again write down the thought and call the tools. The process iterates until the agent decide to submit its conclusion, or a maximum iteration number, which they set to be 10, is reached. In practice, they use OpenAI’s function call to handle the tools

Works Cited

- Ahn, M. (n.d.). Ahn, Michael, et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).

- Alibaba. (n.d.). Gao, Dawei, et al. "AgentScope: A Flexible yet Robust Multi-Agent Platform." *arXiv preprint arXiv:2402.14034* (2024).
- Brown, T. (n.d.). Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
- Chen, M. (n.d.). Chen, Mark, et al. "Evaluating large language models trained on code." *arXiv preprint arXiv:2107.03374* (2021).
- Chi-Min, C. (n.d.). Chan, Chi-Min, et al. "Chateval: Towards better llm-based evaluators through multi-agent debate." *arXiv preprint arXiv:2308.07201* (2023).
- Cobbe, K. (n.d.). Cobbe, Karl, et al. "Training verifiers to solve math word problems." *arXiv preprint arXiv:2110.14168* (2021).
- Creswell, A. (n.d.). Creswell, Antonia, and Murray Shanahan. "Faithful reasoning using large language models." *arXiv preprint arXiv:2208.14271* (2022).
- Creswell, A. (n.d.). Creswell, Antonia, Murray Shanahan, and Irina Higgins. "Selection-inference: Exploiting large language models for interpretable logical reasoning." *arXiv preprint arXiv:2205.09712* (2022).
- Fan, A. (n.d.). Fan, Angela, et al. "ELI5: Long form question answering." *arXiv preprint arXiv:1907.09190* (2019).
- fetchai. (n.d.). <https://github.com/fetchai/uAgents>.
- Geva, M. (n.d.). Geva, Mor, et al. "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies." *Transactions of the Association for Computational Linguistics* 9 (2021): 346-361.
- Glaese, A. (n.d.). Glaese, Amelia, et al. "Improving alignment of dialogue agents via targeted human judgements." *arXiv preprint arXiv:2209.14375* (2022).
- Google & Gemini Team. (n.d.). Reid, Machel, et al. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context." *arXiv preprint arXiv:2403.05530* (2024).

- Hosseini, E. (n.d.). Hosseini-Asl, Ehsan, et al. "A simple language model for task-oriented dialogue." *Advances in Neural Information Processing Systems* 33 (2020): 20179-20191.
- Huang, W. (n.d.). Huang, Wenlong, et al. "Inner monologue: Embodied reasoning through planning with language models." *arXiv preprint arXiv:2207.05608* (2022).
- Jie, Z. (n.d.). Jie, Zhanming, Jierui Li, and Wei Lu. "Learning to reason deductively: Math word problem solving as complex relation extraction." *arXiv preprint arXiv:2203.10316* (2022).
- Kojima, T. (n.d.). Kojima, Takeshi, et al. "Large language models are zero-shot reasoners." *Advances in neural information processing systems* 35 (2022): 22199-22213.
- Lan, Y. (n.d.). Lan, Yihuai, et al. "Mwptoolkit: An open-source framework for deep learning-based math word problem solvers." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. No. 11. 2022.
- Lewis, P. (n.d.). Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." *Advances in Neural Information Processing Systems* 33 (2020): 9459-9474.
- Li, G. (n.d.). Li, Guohao, et al. "Camel: Communicative agents for" mind" exploration of large scale language model society." (2023).
- Long, J. (n.d.). Long, Jieyi. "Large language model guided tree-of-thought." *arXiv preprint arXiv:2305.08291* (2023).
- Lourie, N. (n.d.). Lourie, Nicholas, et al. "Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 15. 2021.
- Madaan, A. (n.d.). Madaan, Aman, and Amir Yazdanbakhsh. "Text and patterns: For effective chain of thought, it takes two to tango." *arXiv preprint arXiv:2209.07686* (2022).
- Madaan, A. (n.d.). Madaan, Aman, et al. "Self-refine: Iterative refinement with self-feedback." *Advances in Neural Information Processing Systems* 36 (2024).
- Mehri, S. (n.d.). Mehri, Shikib, and Maxine Eskenazi. "Unsupervised evaluation of interactive dialog with DialoGPT." *arXiv preprint arXiv:2006.12719* (2020).

- Mittal, A. (2023, December 1). *The Black Box Problem in LLMs: Challenges and Emerging Solutions*. Unite.AI. Retrieved May 19, 2024, from <https://www.unite.ai/the-black-box-problem-in-llms-challenges-and-emerging-solutions/>
- Nakajima, Y. (n.d.). *yoheinakajima/babyagi*. GitHub. Retrieved May 21, 2024, from <https://github.com/yoheinakajima/babyagi>
- Nakano, R. (n.d.). Nakano, Reiichiro, et al. "Webgpt: Browser-assisted question-answering with human feedback." arXiv preprint arXiv:2112.09332 (2021).
- Naveed, H. (n.d.). Naveed, Humza, et al. "A comprehensive overview of large language models." arXiv preprint arXiv:2307.06435 (2023).
- Nye, M. (n.d.). Nye, Maxwell, et al. "Show your work: Scratchpads for intermediate computation with language models." arXiv preprint arXiv:2112.00114 (2021).
- OpenAI. (n.d.). *Achiam, Josh, et al. "Gpt-4 technical report." arXiv preprint arXiv:2303.08774 (2023).*
- Park, J. S. (n.d.). Park, Joon Sung, et al. "Generative agents: Interactive simulacra of human behavior." Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. 2023.
- Pichai, S. (2024, February 15). *Introducing Gemini 1.5, Google's next-generation AI model*. The Keyword. Retrieved May 20, 2024, from <https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/#context-window>
- Proverbs, 2. (n.d.). New International Version (NIV). <https://www.biblegateway.com/passage/?search=Proverbs%2027%3A17&version=NIV>
- Raffel, C. (n.d.). *T5*. Hugging Face. Retrieved May 19, 2024, from https://huggingface.co/docs/transformers/en/model_doc/t5
- Reed, S. (n.d.). Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., ... & de Freitas, N. (2022). A generalist agent. arXiv preprint arXiv:2205.06175.

- Richards, T. (n.d.). *Significant-Gravitas/AutoGPT: AutoGPT is the vision of accessible AI for everyone, to use and to build on. Our mission is to provide the tools, so that you can focus on what matters.* GitHub. Retrieved May 21, 2024, from <https://github.com/Significant-Gravitas/Auto-GPT>
- Schick, T. (n.d.). Schick, Timo, et al. "Toolformer: Language models can teach themselves to use tools." *Advances in Neural Information Processing Systems* 36 (2024).
- Shi, F. (n.d.). Shi, Freda, et al. "Language models are multilingual chain-of-thought reasoners." *arXiv preprint arXiv:2210.03057* (2022).
- Shin, N. (n.d.). Shinn, Noah, Beck Labash, and Ashwin Gopinath. "Reflexion: an autonomous agent with dynamic memory and self-reflection." *arXiv preprint arXiv:2303.11366* (2023).
- Shridhar, M. (n.d.). Shridhar, Mohit, et al. "Alfworld: Aligning text and embodied environments for interactive learning." *arXiv preprint arXiv:2010.03768* (2020).
- Shuster, K. (n.d.). Shuster, Kurt, et al. "Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage." *arXiv preprint arXiv:2208.03188* (2022).
- Srivastava, A. (2022). *Srivastava, Aarohi, et al. "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models." arXiv preprint arXiv:2206.04615 (2022).*
- Talmor, A. (n.d.). Talmor, Alon, et al. "Commonsenseqa 2.0: Exposing the limits of ai through gamification." *arXiv preprint arXiv:2201.05320* (2022).
- Thorne, J. (n.d.). Thorne, James, et al. "FEVER: a large-scale dataset for fact extraction and VERification." *arXiv preprint arXiv:1803.05355* (2018).
- Vaswani, A. (n.d.). Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- Wang, X. (n.d.). Wang, Xuezhi, et al. "Self-consistency improves chain of thought reasoning in language models." *arXiv preprint arXiv:2203.11171* (2022).
- Wei, J. (n.d.). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *36th Conference on Neural Information Processing Systems (NeurIPS 2022).*

- Weize, C. (n.d.). Chen, Weize, et al. "Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents." arXiv preprint arXiv:2308.10848 (2023).
- Wu, Q. (n.d.). Wu, Qingyun, et al. "Autogen: Enabling next-gen llm applications via multi-agent conversation framework." arXiv preprint arXiv:2308.08155 (2023).
- Yang, Z. (n.d.). Yang, Zhilin, et al. "HotpotQA: A dataset for diverse, explainable multi-hop question answering." arXiv preprint arXiv:1809.09600 (2018).
- Yao, S. (n.d.). Yao, Shunyu, et al. "React: Synergizing reasoning and acting in language models." arXiv preprint arXiv:2210.03629 (2022).
- Yao, S. (n.d.). Yao, Shunyu, et al. "Webshop: Towards scalable real-world web interaction with grounded language agents." Advances in Neural Information Processing Systems 35 (2022): 20744-20757.
- Zelikman, E. (n.d.). Zelikman, Eric, et al. "Star: Bootstrapping reasoning with reasoning." Advances in Neural Information Processing Systems 35 (2022): 15476-15488.
- Zhang, X. (n.d.). Zhang, Xinghua, et al. "Wider and deeper llm networks are fairer llm evaluators." arXiv preprint arXiv:2308.01862 (2023).
- Zhou, D. (n.d.). Zhou, Denny, et al. "Least-to-most prompting enables complex reasoning in large language models." arXiv preprint arXiv:2205.10625 (2022).
- Zhu, Y. (n.d.). Zhu, Yunchang, et al. "Adaptive information seeking for open-domain question answering." arXiv preprint arXiv:2109.06747 (2021).

