

International Neural Network Society Workshop on Deep Learning Innovations and Applications

Exploring Language Model Self-Improvement through Domain-Independent Gameplay in Text Environments

Matteo Forasassi^{a,b}, Francesco Maberino^a, Nicolò Brunello^a,
Vincenzo Scotti^{c,*}, Mark James Carman^a

^aDEIB, Politecnico di Milano, Via Giuseppe Ponzio 34, Milano (MI) 20133, Italy

^bInstitute of Logic and Computation, Vienna University of Technology, Favoritenstraße 9-11, Vienna 1040, Austria

^cKASTEL, Karlsruhe Institute of Technology, Am Fasanengarten 5, Karlsruhe 76131, Germany

Abstract

In this paper we investigate the effects of self-training Large Language Models (LLMs) to enhance their reasoning capabilities. We introduce SAL, an iterative training framework where the LLMs engage in textual games with domain-independent prompts. Through self-appraisal, the models autonomously select game instances identified as beneficial to their learning trajectory. The chosen games are then used to fine-tune the LLMs, iteratively improving the model's performance. In our work, we explore different textual games and training approaches. Moreover, we take care to evaluate the language understanding capabilities of the LLMs both before and after the training, to keep track of the effect of the self-training process on the reasoning capabilities also outside of the games environment. Our results demonstrate how the LLM's ability in self-appraisal can allow to improve, even significantly in some games, it's own performance with little impact on its linguistic capabilities. Our findings underscore the potential of leveraging self-training techniques with LLMs to improve their capabilities to solve problems, in particular we show how LLMs self-appraisal capabilities can be exploited to identify relevant experiences useful for fine-tuning.

© 2025 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the scientific committee of the IJCNN 2025.

Keywords: LLM; NLP; Self-training; Self-appraisal; TextWorld

1. Introduction

Large Language Models (LLMs) are the core of the recent paradigm shift in *Natural Language Processing* (NLP) and *Artificial Intelligence* (AI) [34, 26], impacting the research landscape with their unprecedented capacity to understand language, extract information and generate human-like text [23, 7, 25]. The *Transformer architecture* [30] underling these LLMs has demonstrated empirically to be suitable to compress and memorise a vast amount of knowledge (despite it being a lossy compression algorithm) and to be suitable for reasoning (to a certain extent) to solve a given problem [14, 6]. However, we still lack a complete understanding of the possibilities and limits of these features characterising LLMs.

* Corresponding author. Tel.: +49 72160843476.

E-mail address: vincenzo.scotti@kit.edu

Many benchmarks evaluating LLMs assess linguistic capabilities, probing information memorisation and logical entailment [32, 31, 24, 28, 15]. In these evaluations the LLMs are used to score alternative completions of a given context to see how often the correct completion has the higher probability of being observed under the model. More recently, there has been a rising attention to evaluate the reasoning capabilities as well. For example, some experiments approached the usual benchmarks using a *Chain-of-Thoughts* (CoT) [33] approach, where the LLM is prompted to explain its “reasoning” process while producing the answer, often yielding improved results. Other approaches, are currently proposing to evaluate the LLMs on *textual-games* [22], where the model should understand the final goal of the game in order to reason over the next action and plan a sequence of actions to win the game. We focus our work on this last topic.

During our investigations, we developed our framework for *Self-Appraisal Learning* (SAL)¹. The idea behind SAL is to have an LLM play some games to collect experiences and learn from these experiences. Learning is performed on those experiences that the LLM itself judges as useful (i.e., experiences where the LLM detects an improvement towards the final goal of the game). This self-appraisal step where the LLM itself judges the effectiveness of the performed actions is the core of the proposed framework: the LLM likely wouldn’t improve without a reasonable appraisal function and, thus, the LLMs should understand the game to some extent in order to perform well as an appraisal function. In our experiments we show (i) how an LLM can learn from this kind of self-appraisal by improving its performances, (ii) how the SAL framework can be adopted in completely *game-agnostic* settings and (iii) how the SAL framework impacts the linguistic capabilities of the LLM.

We divide the rest of this paper into the following sections. In Section 2, we detail the research questions motivating our work. In Section 3, we describe the SAL training framework. In Section 4, we outline our experimental approach and the benchmarking environment we used to evaluate SAL. In Section 5, we present results of our experiments. In Section 6, we comment on the results based on our research questions. In Section 7, we present the related research lines in using NLP for textual-games. Finally, in Section 8, we summarise our contribution and we propose possible future extensions.

2. Research questions

In this paper, we tackle the following research questions:

RQ1 Can LLMs play textual games without prompts (too) specific to the game?

- How can the rules of the game be provided to the model?
- How can the state of the game be provided to the model?
- How can the model be instructed to choose an action to perform?

RQ2 Can LLMs play the game well through self-appraisal of game performance?

- Can the model be prompted to determine whether the game state has improved as a result of the action performed?
- To what extent is the model able to determine whether the game state has improved?
- Given sufficient ability to determine a state improvement, can this self-appraisal of gameplay be used to choose a more appropriate action to perform?

RQ3 Can LLMs learn to improve their ability to play a game through the use of self-appraisal of gameplay and model fine-tuning?

- Can self-appraisal of gameplay be used to choose appropriate training instances for fine-tuning the model?
- Is fine-tuning the LLM effective for improving the game-playing ability of the model on the game?

RQ4 Does learning to play one or more games impact the overall gaming performance?

- Does learning to play a single game improve the performance in any game?

¹ Source code available at: <https://github.com/MatteoForasassi/Self-Appraisal-Learning/tree/v1.0>

- Does learning to play multiple games improve the performance in any game (better than learning a single game)?

RQ5 What are the effects of learning to reason to solve games on the LLM behaviour?

- Does the LLM become disfluent?
- Are the LLM language understanding capabilities preserved?
- Is LLM factual knowledge affected?

We broke down every question in smaller sub-questions for better understanding and motivating of the research goals.

3. SAL framework

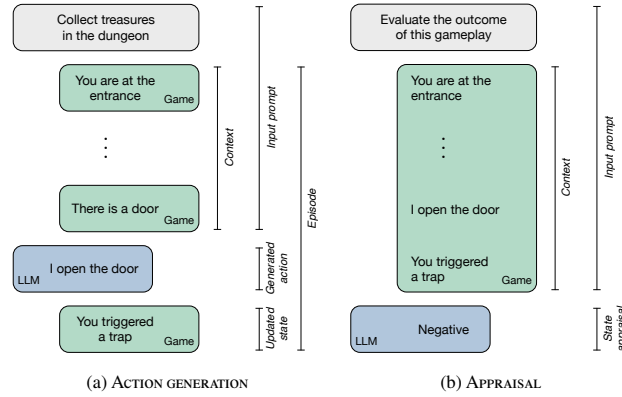


Fig. 1: LLM interacting with the game environment in SAL

The SAL framework is designed to enable LLMs to interact with textual environments, specifically games, in a way that promotes reasoning and self-improvement. The framework involves the LLM acting in the textual environment and evaluating its own behavior in a self-appraisal step to select interaction episodes that provide valuable learning experiences for training, as depicted in Figure 1. The main goal of SAL is to enhance LLM's reasoning capabilities through self-appraisal, allowing it to iteratively refine its strategies and decision-making processes. Moreover, our approach aims at showing how it is possible to improve exploiting the prior knowledge in the LLM and the in-context information, even despite noisy behaviour of the LLM in the appraisal step.

We detailed the training framework of SAL in Algorithm 1. The algorithm consists of three key steps: (i) interaction, (ii) appraisal and (iii) training. During interaction, the LLM engages with the game environment, generating actions based on the game's current state and accumulated experiences. An episode E is represented by every sequence $(s_0, a_0), \dots, (s_{|E|}, a_{|E|})$ composed by pairs of state and action, coming from an interaction with the text environment, that leads to a final state. During appraisal the LLM evaluates its performance by analysing the outcomes of its actions in all episodes. Finally, the training phase uses these appraisals to select the episodes with a positive outcome towards the game end goal and updates the model's parameters, promoting behaviours and strategies leading to improvements. This iterative process allows the LLM to continuously learn and adapt, improving its own ability to navigate and succeed in these textual game environments.

In the algorithm, we divide the trainable parameters of the LLM in two parts: θ , the parameters of the pre-trained model, and $\Delta\theta$, the adapter parameters. The adapter parameters, which are updated during the training process, work as a patch to apply on the original parameters, which stay frozen during the training process. To the end of this work we use the *Low Rank Adaptation* (LoRA) [16], which allow for efficient and regularised fine-tuning. While this is a technical implementation detail rather than a conceptual one, we wanted to stress out the importance of using these adapters during the fine tuning steps. We provide further details on the motivations for these adapters in Section 4.

Algorithm 1 LLM self-training algorithm**Require:**

<ul style="list-style-type: none"> • $\vartheta, \Delta\vartheta$ • $\tau \in \mathbb{R}$ • $n_{epochs}, n_{episodes}$ • $f_{init}(): \emptyset \rightarrow S_0$ • $f_{transition}(\cdot): S \times A \rightarrow S$ • $f_{end}(\cdot): S \rightarrow \mathbb{1}$ • $f_{eval}(\cdot \vartheta): S \rightarrow \mathbb{1}$ 	<ul style="list-style-type: none"> ▷ Model and adapter weights ▷ Score tolerance
<p>for $i \in [0, n_{epochs}) \subseteq \mathbb{N}$ do</p> <p style="padding-left: 20px;">$D \leftarrow \{\}$</p> <p style="padding-left: 20px;">$D_g \leftarrow \{\}$</p> <p style="padding-left: 20px;">for $j \in [0, n_{episodes}) \subseteq \mathbb{N}$ do</p> <p style="padding-left: 40px;">$E \leftarrow \{\}$</p> <p style="padding-left: 40px;">$s \leftarrow f_{init}()$</p> <p style="padding-left: 40px;">$g \leftarrow 0$</p> <p style="padding-left: 40px;">while $\neg f_{end}(s)$ do</p> <p style="padding-left: 60px;">$a \leftarrow a \sim P_{LM}(a E, s; \vartheta, \Delta\vartheta)$</p> <p style="padding-left: 60px;">$s' \leftarrow f_{transition}(s, a)$</p> <p style="padding-left: 60px;">$g \leftarrow g + f_{eval}(s' \vartheta)$</p> <p style="padding-left: 60px;">$E \leftarrow E \cup \{(s, a)\}$</p> <p style="padding-left: 60px;">$s \leftarrow s'$</p> <p style="padding-left: 40px;">end while</p> <p style="padding-left: 40px;">$D_g \leftarrow D_g \cup \{(E, g/ E)\}$</p> <p style="padding-left: 20px;">end for</p> <p style="padding-left: 20px;">$\mu_{g \max} \leftarrow \max_{(E, \mu_g) \in D_g} \mu_g$</p> <p style="padding-left: 20px;">for $(E, \mu_g) \in D_g$ do</p> <p style="padding-left: 40px;">if $\mu_g \geq \mu_{g \max} - \tau$ then</p> <p style="padding-left: 60px;">$D \leftarrow D \cup \{E\}$</p> <p style="padding-left: 40px;">end if</p> <p style="padding-left: 20px;">end for</p> <p style="padding-left: 20px;">for $E \equiv \{(s_0, a_0), \dots, (s_N, a_N)\} \in D$ do</p> <p style="padding-left: 40px;">for $i \in [0, N] \subseteq \mathbb{N}$ do</p> <p style="padding-left: 60px;">$E' \leftarrow \{(s_{i'}, a_{i'}) \mid \forall i' < i\}$</p> <p style="padding-left: 60px;">$\mathcal{L}(\Delta\vartheta) \leftarrow -\ln P_{LM}(a_i E', s_i; \vartheta, \Delta\vartheta)$</p> <p style="padding-left: 60px;">$\Delta\vartheta \leftarrow \Delta\vartheta - \eta \cdot \nabla_{\Delta\vartheta} \cdot \mathcal{L}(\Delta\vartheta)$</p> <p style="padding-left: 40px;">end for</p> <p style="padding-left: 20px;">end for</p> <p>end for</p> <p>return $\Delta\vartheta$</p>	<ul style="list-style-type: none"> ▷ Initialisation function ▷ Transition function ▷ End of game function ▷ Self-evaluation function ▷ Iterate over epochs ▷ Initialise data set of episodes ▷ Iterate over episodes ▷ Initialise episode ▷ Get episode initial state ▷ Initialise cumulative evaluation ▷ Generate action ▷ Update episode state ▷ Update episode ▷ Update current state ▷ Save ▷ Get best result ▷ Iterate over collected episodes ▷ Check average score ▷ Save episode ▷ Iter. episodes ▷ Loss ▷ Update weights

During the interaction phase, the LLM is provided with the initial game state s_0 . For every given state, the model is required to generate an action and receive in response a new updated game state. The i -th action a_i is generated prompting the model with the sequence of states and actions $(s_0, a_0), \dots, (s_{i-1}, a_{i-1})$ down to the current point $i - 1$. This approach allows the LLM to interact iteratively with the environment until an end condition is met (the level of the game is finished or a maximum number of actions is reached). Throughout this phase, the model accumulates episodes of state-action pairs that form the basis for subsequent evaluation and training.

The self-appraisal and experience selection phase is critical for fine-tuning the model and improve its performances. After interacting with the environment, the model evaluates the outcomes of its actions using a self-evaluation function. This function assesses whether a state in a game episode s_i reached through action a_i represent an improvement by prompting the model with the action-state pairs up to the i -th interaction: $(s_0, a_0), \dots, (s_i, a_i)$. The outputs of the appraisal steps are accumulated in a cumulative score g for each episode and then normalised on the length of the episode in $\mu_g = g/|E|$. All episodes whose normalised cumulative score μ_g does not fall further than a fixed threshold τ from the maximum normalised cumulative score $\mu_{g \max}$ are selected for the training step.

During the training step, the LLM's parameters are updated through gradient-based optimisation, promoting the actions of successful episodes and, thus, implicitly promoting successful strategies and discarding less effective ones. The LLM is fine-tuned to minimise the negative log-likelihood of the actions the model predicted during the experience collection, we reported the loss in Equation (1).

This process ensures that the model's training is focused on the most informative and beneficial experiences, promoting continuous improvement in reasoning and decision-making.

$$\mathcal{L}(\Delta\theta) = -\ln P(a_i | s_0, a_0, \dots, s_{i-1}, a_{i-1}, s_i; \theta, \Delta\theta) \quad (1)$$

Alternatively to this *supervised* approach, we considered also a *contrastive* approach where we retain also episodes with a negative outcome for training. We compute the negated loss on these episodes to discourage the behaviour leading to less effective strategies. The resulting negated loss is multiplied by a scaling factor α , inversely proportional to the number of actions extracted from negative episodes in the dataset, as described by Equation (2).

$$\mathcal{L}_{negated}(\Delta\theta) = -\alpha(-\ln P(a_i | s_0, a_0, \dots, s_{i-1}, a_{i-1}, s_i; \theta, \Delta\theta)) \quad (2)$$

4. Experiments

Table 1: LLM gaming approach exploration on COIN COLLECTOR

Training configs			Scores							
Verbs	Similarity	Method	Epoch	Level					Average	
				1	2	3	4	5		6
✓	✓	Baseline	0	0.95	0.77	0.50	0.30	0.25	0.12	0.48
		Supervised	1	1.00	0.77	0.85	0.57	0.42	0.40	0.67
		Contrastive	1	1.00	0.85	0.7	0.35	0.33	0.20	0.57
		Baseline	0	0.85	1.00	0.72	0.30	0.27	0.2	0.56
		Supervised	1	1.00	1.00	0.97	0.95	0.87	0.75	0.92
		Contrastive	1	1.00	1.00	1.00	1.00	0.93	1.00	0.98
		Baseline	0	0.97	1.00	0.70	0.50	0.32	0.22	0.62
		Supervised	1	0.80	0.95	0.87	0.77	0.70	0.50	0.77
		Contrastive	1	1.00	1.00	0.95	0.83	0.60	0.47	0.80
✓	✓	Baseline	0	1.00	1.00	0.83	0.45	0.25	0.20	0.62
		Supervised	1	1.00	1.00	1.00	0.95	0.80	0.83	0.93
		Contrastive	1	1.00	1.00	0.97	0.87	0.72	0.65	0.87

Before conducting the main experiments, we engaged in initial explorations to identify and address preliminary challenges. One notable issue was the LLM’s difficulty in generating actions in the correct format due to its instruction tuning. Additionally, we observed that the model often produced semantically correct but lexically incorrect actions. For example, instead of generating the action “pick up coin” the model might produce “grab the coin”, which, while semantically accurate, is not recognised by the game. Furthermore, we discovered that fine-tuning on game episodes negatively affected the appraisal system, leading to corrupted evaluations.

We ran all evaluations using the same approach. We evaluated each model on different level, running the same amount of games (40) for each level. In this way, we could collect comparable results level-wise and method-wise. In fact, we approached each game using different methods to collect insights on strong and weak points of our framework.

To address these action-related issues, we conducted preliminary experiments using the Coin Collector game to evaluate alternative solutions. We compared the performance of the vanilla model with three modified approaches: (i) including a comprehensive list of all possible actions in the game’s initial instructions (we avoided listing only currently available actions given a state to maintain a game-agnostic approach), (ii) employing semantic similarity to match generated actions with the closest valid game actions and (iii) combining both strategies. We compared the results on some evaluation episodes using the base model, the model after one iteration of supervised fine-tuning with the SAL framework and the models after one iteration of contrastive fine-tuning with the SAL framework. As shown in Table 1, simply informing the model of the entire action space by listing the accepted verbs significantly improved performance. To mitigate the corruption of the appraisal system during fine-tuning, we used LoRA adapters [16] that were disabled during the appraisal phase.

For the main experiments, we evaluated the base model and several variants fine-tuned using the SAL framework. Specifically, we tested the base model, the model fine-tuned in the supervised approach with SAL after one and two iterations, and the model fine-tuned using the contrastive approach after one iteration of SAL. These evaluations were conducted across the four selected games. In addition to measuring in-game performance on a separate set of game instances, we assessed the cross-game reasoning improvements by evaluating each model trained on a single game across all other games. Furthermore, we performed a Leave-One-Out (LOO) experiment, where the LLM was fine-tuned on all games except one and then evaluated on the remaining game to investigate any potential scaling effects related to the number and diversity of games.

Alongside the evaluation on the gaming performance, we assessed also the quality of model’s appraisal. Being the appraisal the core of the SAL framework, since it enables selecting the correct behaviour and strategies to promote, it’s crucial to ensure the quality of LLM’s predictions in this step. To do so, we analyse the *Spearman correlation* between the average predicted values during the appraisal step (we have the model produce a score in the range $[0,5) \in \mathbb{N}$) and the cumulative collected reward weighted on the ratio between the length of the optimal trajectory (i.e., the length of the episode obtained under optimal policy) and the length of the generated episode.

This metric will give us an indication on how similar model prediction on the value at a given state are if compared with the value of a model following the optimal policy.

Besides the gaming and appraisal evaluations, we further investigated the effects of the training under the SAL framework on the LLMs after each training step using the H6 benchmark for LLM evaluation [20] (we used the benchmarks implementations available with the *Language Model Evaluation Harness* tool [13]). The objective of this additional evaluation was twofold: on one side, we were interested in determining whether and to what extent the SAL framework impacted the linguistic capabilities of the models and, on the other side, we wanted to assess if self-training to reason on specific tasks (the games) improved the LLMs’ reasoning abilities on other tasks requiring reasoning (the benchmarks). The comprehensive evaluation ensured a thorough understanding of both the benefits and potential drawbacks of the SAL framework in enhancing LLM reasoning and gameplay performance.

In all experiments we used the Mistral 7B LLM with instruction tuning performed on the OpenOrca dataset [18] [19]. We used `paged adamw 32bit` as optimizer, setting a learning rate scheduler with an initial value of $2e-4$ and a batch size of 1 for training.

5. Results

We report the main evaluation results on the considered games in Table 2, and the level-wise breakdown in Figure 2. The LLM achieves the best results on COIN COLLECTOR and TREASURE HUNTER, which are the simplest of the considered games. In more complex games like COOKING, we see an improvement of the results; however, the training proceeded

Table 2: Experiments results

Training configs		Epoch	Score	H6
Game	Method			
COIN COLLECTOR	Baseline	0	0.56	0.54
	Supervised	1	0.92	0.53
		2	0.97	0.53
	Contrastive	1	0.99	0.53
	Leave-one-out	1	0.90	0.54
COMMONSENSE	Baseline	0	0.06	0.54
	Supervised	1	0.11	0.53
		2	0.05	0.53
	Contrastive	1	0.06	0.53
	Leave-one-out	1	0.02	0.53
COOKING	Baseline	0	0.09	0.54
	Supervised	1	0.17	0.54
		2	0.26	0.53
	Contrastive	1	0.24	0.53
	Leave-one-out	1	0.18	0.53
TREASURE HUNTER	Baseline	0	0.59	0.54
	Supervised	1	0.67	0.54
		2	0.86	0.54
	Contrastive	1	0.89	0.53
	Leave-one-out	1	0.79	0.53

much slower than the other two games. Finally, in COMMONSENSE we can see how the SAL framework struggles in improving LLM’s performance and quickly overfits.

Looking at the different approaches we can make two important considerations. The former consideration is that using contrastive examples speeds up training and allow to get good results on higher (more difficult) level (at least in COIN COLLECTOR and TREASURE HUNTER), especially if compared with a single round of supervised training. The latter consideration comes from the LOO results, here we can see that playing other games is helpful to generalise and learn how to play at least better than the baseline model on unseen games (with COMMONSENSE being the exception). This second consideration is particularly helpful if we consider how different some of the games are.

As premised we evaluated the models using 40 game episodes for each level of the considered games. The number of training episodes changes depending on the appraisal and the distance threshold from the best episode collected during the correct epoch. We report in Figure 3 the number of game episodes used to train the reported models. Going on with the training allows the LLM to better explore higher levels, meaning that the LLM is improving its capabilities. The lower results achieved on COOKING and COMONSENSE can be explained looking at these graphs. We can see that the LLM is having an hard time collecting game episodes to learn from, as a results the scores on these two games are lower.

Our LLM agent learns from the game episodes collected through self-appraisal. that is the core of the SAL framework. As we can see from Figure 4, the LLM achieves a better correlation score on COIN COLLECTOR and TREASURE HUNTER, where results are also better that those of the other games. Conversely, on COMMONSENSE the correlation is significantly

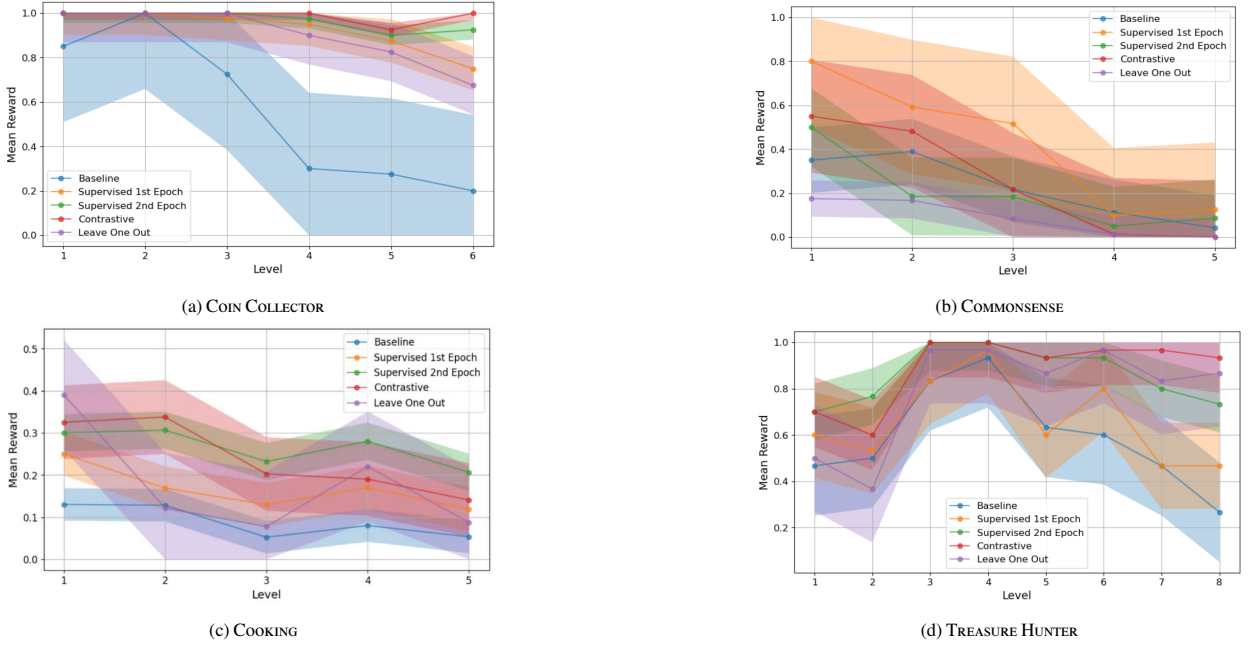


Fig. 2: Results level-wise and approach-wise on the different games

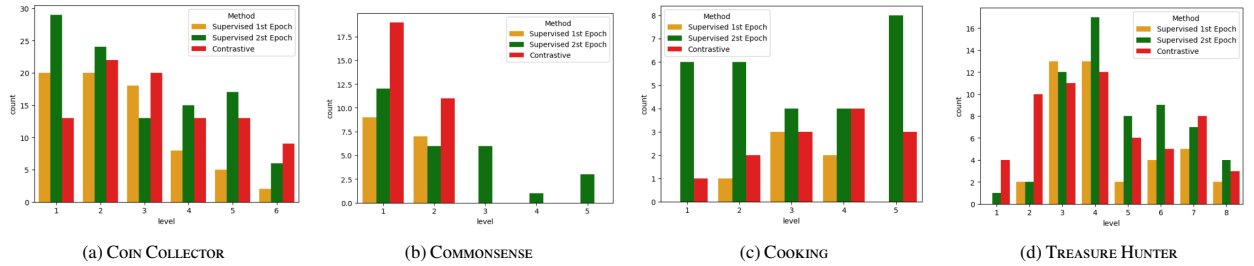


Fig. 3: Collected episodes level-wise and approach-wise on the different games, the legend refers to the models obtained with the episodes reported in the figures

lower, explaining the lower performances. This results underlines how appraisal capabilities are crucial to a successful training under the SAL framework. Moreover, we can hypothesise that model complexity directly affects the appraisal capabilities and thus the results of learning.

Finally, we would like to point out the results on language understanding using the H6 benchmark. Table 2 shows us that after training under the SAL framework, there have been, if any, only slight changes on the results. This results hints that the LLM is learning how to play the games without disrupting its internal knowledge.

6. Discussion

Our findings suggest that LLMs can indeed play textual games without needing game-specific prompts, though the approach we employed, which involved simple text-based interactions, raises important considerations. While the models relied on domain-independent textual prompts, some might argue that the games are not fully “game agnostic” due to their reliance on verbs in the prompts and semantic-similarity in the action selection. Despite this, we only provided the model with game-related instructions via natural language text, without introducing hardcoded actions or game-specific

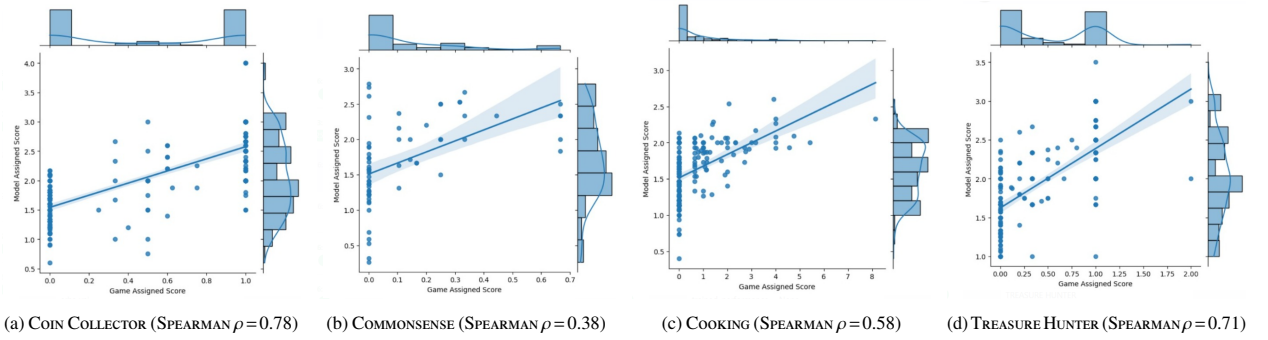


Fig. 4: Spearman correlation between the average appraisal predicted by the LLM and the weighted cumulative reward collected by the LLM

mechanics. This approach allowed us to focus primarily on developing the SAL framework to explore how the model could autonomously select instances to improve its performance. This setup emphasized the self-appraisal process and game-state evaluation as key mechanisms for learning, without artificially constraining the gameplay mechanics to domain-specific rules.

Addressing the broader applicability of our framework, we acknowledge that the fine-tuning proved to be computationally onerous. This limitation means that while we observed positive results in terms of performance improvement, particularly with simpler games, scaling the approach to a wider range of games or further refining the self-appraisal mechanism is an area for future work. Furthermore, while the LLM’s understanding capabilities remained largely intact throughout the training process, we recognize that the performance on games like Commonsense highlights the potential limits of the model’s ability to generalize complex reasoning tasks, especially when training data is sparse or the game is particularly difficult.

7. Related works

NLP techniques have been widely adopted to tackle the development of agents acting in textual gaming [22]. These environments are characterised by the need for language understanding to assess the current game state and by the ability to deal with reward signals that may be noisy or dynamic. Research have widely explored this area to tackle the challenges posed by textual games, ranging from classic text adventures like *Zork* to tailored environments such as *TextWorld* [10] or *BabyAI-Text* [8]. The common approach is to interpret the game as a *Partially Observable Markov Decision Process* (POMDP) and adopt *Reinforcement Learning* (RL) techniques to train an agent towards the optimal strategy to beat the game.

Traditional methodologies based on *Knowledge Engineering* (KE) and *symbolic reasoning* have provided foundational frameworks for game-solving strategies. These approaches often involve expert-provided *prior knowledge* and rule-based systems to play. In contrast, recent advancements leverage LLMs to provide agents with semantic understanding, exploiting all the prior knowledge of the LLM coming from the pre-training [12]. These diverse approaches highlight the variety of possible strategies that may be employed to address textual game challenges.

While KE-based and game-specific approaches offer robust solutions tailored to individual games, their domain-specific nature poses challenges in scalability and adaptability across different environments. For instance, there has been a lot of work to provide structured solution to *Zork* using RL and KE to build a state representation, even combining these technique with LMs like BERT [11] to help improve information extraction from the ongoing game [2, 3, 4]. Similarly, the EXPLORER framework [5] incorporates *neuro-symbolic reasoning* to enhance model flexibility and adaptability in *TextWorld* scenarios. However, such approaches often require substantial modifications to accommodate new game elements or domains, limiting their reusability and generalisation. Another example comes from the GATA framework [1], which uses a *dynamic belief graph* representation in conjunction with a Recurrent Neural Network (RNN) to learn how to play *TextWorld* games. This particular approach requires a separate self-supervised training to have the underlying RNN effectively use the belief graph during the games.

In more recent solutions, researchers have explored leveraging pre-trained LLMs, such as ChatGPT [21], to tackle complex textual games like Zork [29]. The key-points from this research were particularly insightful, even a model as complex as ChatGPT has difficulties in generating actions in the correct format when used in zero-shots or few-shot learning and requires a human in the loop to interact correctly with the game. Despite these difficulties, ChatGPT showed how its prior knowledge on Zork, helped face the game. Other frameworks like GLAM [8] use LLMs like Flan T5 [9] to align internal symbols with external dynamics, facilitating task-solving in environments like BabyAI-Text. Despite these advancements, many existing frameworks rely on RL-inspired solutions, particularly policy learning, and game-specific knowledge to drive model behaviour and decision-making.

To our knowledge, among the LLM-based frameworks for RL in textual environments, only *Reflexion* [27] represents the closest alternative to SAL. Reflexion uses a *self-reflexion* mechanism to exploit the environment feedback to evaluate agents actions, but it does so with explicit external feedback. For example, with Reflexion the model can write code and run tests; however, the feedback asking to fix the code given the results of test seems to be managed via external code, rather than asking the model to inspect the test results. Notably, this method does not require updating the model and works only, thus introducing other limitations due to context size.

Finally, we report about *Motif* [17], an approach to extract prior knowledge from LLMs to train RL agents. The idea is to use the LLM to annotate transitions generated playing TextWorld games using an LLM, as we do in our appraisal step. The annotations are then used to train a reward function and from that an agent playing the game. Differently from SAL, the knowledge from LLM is used to train more traditional RL models rather than fine tune the LLM to play better.

In contrast to prevalent RL approaches for text environments the SAL framework detaches from RL as it augments the exploration process with self-appraisal mechanisms obtained by the word knowledge coming from the underlying LLM and removing the need for a domain expert providing the reward function. Moreover, while existing frameworks often rely on game-specific information and heuristics to play, SAL uses the LLM in a game-agnostic way, emphasising the adaptability and the generalisability of a model reasoning across diverse textual gaming environments.

8. Conclusion

In this paper we presented the SAL framework, a framework for autonomous model improvement within textual environments. By integrating a self-appraisal mechanisms into the learning process, SAL enables LLMs to iteratively enhance their performance without explicit human supervision. SAL's game-agnostic nature makes it applicable across various environments, as we show in the evaluation on different games, detaching it from previous expert-driven task-specific approaches. This versatility makes SAL a promising solution towards advancing the self improvement of LLMs.

Starting from our SAL framework, we envision different future developments for our research aimed at better understanding LLMs interaction and reasoning capabilities within textual (gaming) environments. One promising direction is the integration of the Chain-of-Thoughts (CoT) reasoning mechanisms in the pipeline. This mechanism allows LLMs elaborate their responses providing an explanation of their reasoning during the process of generating an action or while working on the appraisal steps. Additionally, we are interested in examining the scaling behaviour, both on model complexity and task complexity. We are interested in understanding the effect of these two aspects in SAL.

Acknowledgments

This work was partially supported by the FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence), by funding from the pilot program Core Informatics at KIT (KiKIT) of the Helmholtz Association (HGF) and supported by the German Research Foundation (DFG) - SFB 1608 - 501798263 and KASTEL Security Research Labs, Karlsruhe.

References

- [1] Adhikari, A., Yuan, X., Côté, M., Zelinka, M., Rondeau, M., Laroche, R., Poupart, P., Tang, J., Trischler, A., Hamilton, W.L., 2020. Learning dynamic belief graphs to generalize on text-based games, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. URL: <https://proceedings.neurips.cc/paper/2020/hash/1fc30b9d4319760b04fab735fbfed9a9-Abstract.html>.

- [2] Ammanabrolu, P., Riedl, M.O., 2019. Playing text-adventure games with graph-based deep reinforcement learning, in: Burstein, J., Doran, C., Solorio, T. (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics. pp. 3557–3565. URL: <https://doi.org/10.18653/v1/n19-1358>, doi:10.18653/V1/N19-1358.
- [3] Ammanabrolu, P., Tien, E., Hausknecht, M.J., Riedl, M.O., 2020a. How to avoid being eaten by a grue: Structured exploration strategies for textual worlds. CoRR abs/2006.07409. URL: <https://arxiv.org/abs/2006.07409>, arXiv:2006.07409.
- [4] Ammanabrolu, P., Tien, E., Luo, Z., Riedl, M.O., 2020b. How to avoid being eaten by a grue: Exploration strategies for text-adventure agents. CoRR abs/2002.08795. URL: <https://arxiv.org/abs/2002.08795>, arXiv:2002.08795.
- [5] Basu, K., Murugesan, K., Chaudhury, S., Campbell, M., Talamadupula, K., Klinger, T., 2024. EXPLORER: exploration-guided reasoning for textual reinforcement learning, in: Graham, Y., Purver, M. (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, Association for Computational Linguistics. pp. 394–405. URL: <https://aclanthology.org/2024.eacl-long.24>.
- [6] Biderman, S., Prashanth, U.S., Sutawika, L., Schoelkopf, H., Anthony, Q., Purohit, S., Raff, E., 2023. Emergent and predictable memorization in large language models, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. URL: http://papers.nips.cc/paper/_files/paper/2023/hash/59404fb89d6194641c69ae99ecdf8f6d-Abstract-Conference.html%7D.
- [7] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. URL: <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>.
- [8] Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., Oudeyer, P., 2023. Grounding large language models in interactive environments with online reinforcement learning, in: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, PMLR*. pp. 3676–3713. URL: <https://proceedings.mlr.press/v202/carta23a.html>.
- [9] Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Narang, S., Mishra, G., Yu, A., Zhao, V.Y., Huang, Y., Dai, A.M., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V., Wei, J., 2022. Scaling instruction-finetuned language models. CoRR abs/2210.11416. URL: <https://doi.org/10.48550/arXiv.2210.11416>, doi:10.48550/ARXIV.2210.11416, arXiv:2210.11416.
- [10] Côté, M., Kádár, Á., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M.J., Asri, L.E., Adada, M., Tay, W., Trischler, A., 2018. Textworld: A learning environment for text-based games, in: Cazenave, T., Saffidine, A., Sturtevant, N.R. (Eds.), *Computer Games - 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers, Springer*. pp. 41–75. URL: https://doi.org/10.1007/978-3-030-24337-1_3, doi:10.1007/978-3-030-24337-1_3.
- [11] Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding, in: Burstein, J., Doran, C., Solorio, T. (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics. pp. 4171–4186. URL: <https://doi.org/10.18653/v1/n19-1423>, doi:10.18653/V1/N19-1423.
- [12] Du, Y., Watkins, O., Wang, Z., Colas, C., Darrell, T., Abbeel, P., Gupta, A., Andreas, J., 2023. Guiding pretraining in reinforcement learning with large language models, in: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, PMLR*. pp. 8657–8677. URL: <https://proceedings.mlr.press/v202/du23f.html>.
- [13] Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonnell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., Zou, A., 2023. A framework for few-shot language model evaluation. URL: <https://zenodo.org/records/10256836%7D>, doi:10.5281/zenodo.10256836.
- [14] Geva, M., Schuster, R., Berant, J., Levy, O., 2021. Transformer feed-forward layers are key-value memories, in: Moens, M., Huang, X., Specia, L., Yih, S.W. (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Association for Computational Linguistics. pp. 5484–5495. URL: <https://doi.org/10.18653/v1/2021.emnlp-main.446%7D>, doi:10.18653/V1/2021.EMNLP-MAIN.446.
- [15] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J., 2021. Measuring massive multitask language understanding, in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net. URL: <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- [16] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2022. Lora: Low-rank adaptation of large language models, in: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net. URL: <https://openreview.net/forum?id=nZevKeeFYf9>.
- [17] Klissarov, M., D'Oro, P., Sodhani, S., Raileanu, R., Bacon, P., Vincent, P., Zhang, A., Henaff, M., 2024. Motif: Intrinsic motivation from artificial intelligence feedback, in: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, OpenReview.net. URL: <https://openreview.net/forum?id=tmBKlIecDE9>.
- [18] Lian, W., Goodson, B., Wang, G., Pentland, E., Cook, A., Vong, C., "Teknum", 2023. Mistralorca: Mistral-7b model instruct-tuned on filtered openorca1 gpt-4 dataset. <https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca>.

- [19] Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H.W., Tay, Y., Zhou, D., Le, Q.V., Zoph, B., Wei, J., Roberts, A., 2023. The flan collection: Designing data and methods for effective instruction tuning, in: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.), International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA, PMLR. pp. 22631–22648. URL: <https://proceedings.mlr.press/v202/longpre23a.html>.
- [20] Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M.S., Love, J., Tafti, P., Hussenot, L., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., Tacchetti, A., Bulanov, A., Paterson, A., Tsai, B., Shahriari, B., Lan, C.L., Choquette-Choo, C.A., Crepy, C., Cer, D., Ippolito, D., Reid, D., Buchatskaya, E., Ni, E., Noland, E., Yan, G., Tucker, G., Muraru, G., Rozhdestvenskiy, G., Michalewski, H., Tenney, I., Grishchenko, I., Austin, J., Keeling, J., Labanowski, J., Lespiau, J., Stanway, J., Brennan, J., Chen, J., Ferret, J., Chiu, J., et al., 2024. Gemma: Open models based on gemini research and technology. CoRR abs/2403.08295. URL: <https://doi.org/10.48550/arXiv.2403.08295>, doi:10.48550/ARXIV.2403.08295, arXiv:2403.08295.
- [21] OpenAI, 2023. GPT-4 technical report. CoRR abs/2303.08774. URL: <https://doi.org/10.48550/arXiv.2303.08774>, doi:10.48550/ARXIV.2303.08774, arXiv:2303.08774.
- [22] Osborne, P., Nömm, H., Freitas, A., 2022. A survey of text games for reinforcement learning informed by natural language. Trans. Assoc. Comput. Linguistics 10, 873–887. URL: <https://transacl.org/ojs/index.php/tacl/article/view/3595>.
- [23] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21, 140:1–140:67. URL: <http://jmlr.org/papers/v21/20-074.html>.
- [24] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. Squad: 100, 000+ questions for machine comprehension of text, in: Su, J., Carreras, X., Duh, K. (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, The Association for Computational Linguistics. pp. 2383–2392. URL: <https://doi.org/10.18653/v1/d16-1264>, doi:10.18653/V1/D16-1264.
- [25] Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M.S., Xu, C., Thakker, U., Sharma, S.S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N.V., Datta, D., Chang, J., Jiang, M.T., Wang, H., Manica, M., Shen, S., Yong, Z.X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Févry, T., Fries, J.A., Teehan, R., Scao, T.L., Biderman, S., Gao, L., Wolf, T., Rush, A.M., 2022. Multitask prompted training enables zero-shot task generalization, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022, OpenReview.net. URL: <https://openreview.net/forum?id=9Vrb9DOWI4>.
- [26] Scotti, V., Sbattella, L., Tedesco, R., 2024. A primer on seq2seq models for generative chatbots. ACM Comput. Surv. 56, 75:1–75:58. URL: <https://doi.org/10.1145/3604281>, doi:10.1145/3604281.
- [27] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S., 2023. Reflexion: language agents with verbal reinforcement learning, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10–16, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.
- [28] Srivastava, A., Rastogi, A., Rao, A., Shoeb, A.A.M., Abid, A., Fisch, A., Brown, A.R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A.W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Rahane, A., Iyer, A.S., Andreassen, A., Santilli, A., Stuhlmüller, A., Dai, A.M., La, A., Lampinen, A.K., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubakaran, A., Mullokkandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakas, A., et al., 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. CoRR abs/2206.04615. URL: <https://doi.org/10.48550/arXiv.2206.04615>, doi:10.48550/ARXIV.2206.04615, arXiv:2206.04615.
- [29] Tsai, C., Zhou, X., Liu, S.S., Li, J., Yu, M., Mei, H., 2023. Can large language models play text games well? current state-of-the-art and open questions. CoRR abs/2304.02868. URL: <https://doi.org/10.48550/arXiv.2304.02868>, doi:10.48550/ARXIV.2304.02868, arXiv:2304.02868.
- [30] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp. 5998–6008. URL: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [31] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R., 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems, in: Wallach, H.M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp. 3261–3275. URL: <https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html>.
- [32] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R., 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019, OpenReview.net. URL: <https://openreview.net/forum?id=rJ4km2R5t7>.
- [33] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28–December 9, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.
- [34] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., Wen, J., 2023. A survey of large language models. CoRR abs/2303.18223. URL: <https://doi.org/10.48550/arXiv.2303.18223>, doi:10.48550/ARXIV.2303.18223, arXiv:2303.18223.