Probing Reasoning of Language Models with Inductive In-Context Learning

Iordanis Fostiropoulos , **Laurent Itti**
University of Southern California
{fostirop, itti}@usc.edu

Abstract

Despite their recent success, Language Models (LMs) have brought to question whether they statistically repeat data ('stochastic parrots')[Bender *et al.*, 2021] or can learn the underlying *generative process* of the data. Current benchmarks used to probe the reasoning ability of LMs can be ambiguous as it is unclear whether the model has learned the benchmark or the generative process of the data. In this work we introduce a novel evaluation setting that we use with *Inductive In-Context Learning* (IIL) and a dataset, `ReAnalogy`, to probe the reasoning ability of LMs. `ReAnalogy` consists of sequences with positive examples, generated from regular expressions (regex), and contains quasi-natural language. We use regex to evaluate *implicitly* whether a LM can infer 'Rules' (regex) given limited sets of examples ('Facts'). We use the LM to generate additional Facts to evaluate whether the generated Facts abide by the Rules. We evaluate a `GPT` model in our setting and compare with the same model where a Rule is injected during training to replace a Fact. We use IIL during evaluation to probe the model to infer the Rule given Facts. We then use the inferred Rule to synthesize an additional Fact. IIL improves 'reasoning' performance by as much as 33%. Our results suggest that LMs can learn more than statistical patterns in the data and we support our findings with ablation studies. We evaluate our dataset with existing benchmarks and baselines in inductive programming and find that current state-of-the-art symbolic or neuro-symbolic approaches fail to the complexity of our dataset; while the existing dataset and benchmark in the domain are inapplicable for LMs. Our probing method and dataset are complex enough for LMs and applicable for evaluating the inductive reasoning abilities of LMs, while IIL can improve 'reasoning' of LMs. We make our dataset available at https://github.com/fostiropoulos/reanalogy

1 Introduction

Reasoning capabilities of Language Models (LMs) has been of recent interest in the Machine Learning (ML) community and beyond. Evaluating the ability of LMs to reason is an open problem where current work focus on psychometric tests designed based on human priors that are inapplicable to LMs [Yu *et al.*, 2023; Bubeck *et al.*, 2023]. Principled benchmarks on evaluating reasoning can contain annotator bias due to the ambiguity inherent in natural language. Although the goal is for LMs to perform inductive reasoning comparably to humans, a more systematic approach is required to evaluate and compare between models.

Inductive reasoning tasks require a model to infer general rules given observations, where the rule must match the observations such that $\{\texttt{Facts}_i, \forall i\} \rightarrow \texttt{Rule}$. Consider a toy example of an inductive reasoning task in natural language {`apples, oranges, pears, ...` } → `Plural Nouns`; this can be ambiguous as these same facts could also define `Fruits`. Then, `cars` could be part of {`Plural Nouns`} but not `Fruits`.Under such ambiguity, abductive inference attempts to find the most probable Rule that describes the data (e.g., here `Fruits` might be preferred as it is a smaller set than `Plural Nouns`). In ML literature, inductive reasoning is an overloaded term and as such we also use the term to refer to the ambiguous setting. In contrast, deductive reasoning would seek to deduce Facts from Rules.

Much of the criticism of language models has been on whether they are 'stochastic parrots' [Bender *et al.*, 2021] as they learn the statistical patterns of the data but not the underlying generative process, i.e., `Fruits`. However under ambiguity, there can be many explanations for the same Facts, optimizing the model for learning a single Rule from a list of Facts can be biased to the labels of the annotator; e.g., if the annotator marked all of those as Plural Nouns, how can we evaluate a model on its ability to infer `Fruits` and penalize if it cannot?

Unambiguous evaluation approaches include, Inductive Logic Programming (ILP) where the goal is to deduce the logical rules from a set of examples (positive and negative). For example: $\{\texttt{abcd}, \texttt{abzz}, \texttt{abeee}, \dots\} \rightarrow \texttt{ab[a-z]}\star$ or 2, 8, 16 $\cdots \rightarrow 2^i$. ILP has been motivated by recent work [Cropper *et al.*, 2022] as a way of evaluating reasoning of LMs. While the field has mainly considered benchmarks limited in vocabulary, size or that require manual effort in curating and processing examples. Further, baseline methods are outdated when compared to current LMs. As such, there is a gap between natural language datasets in which there is ambiguity

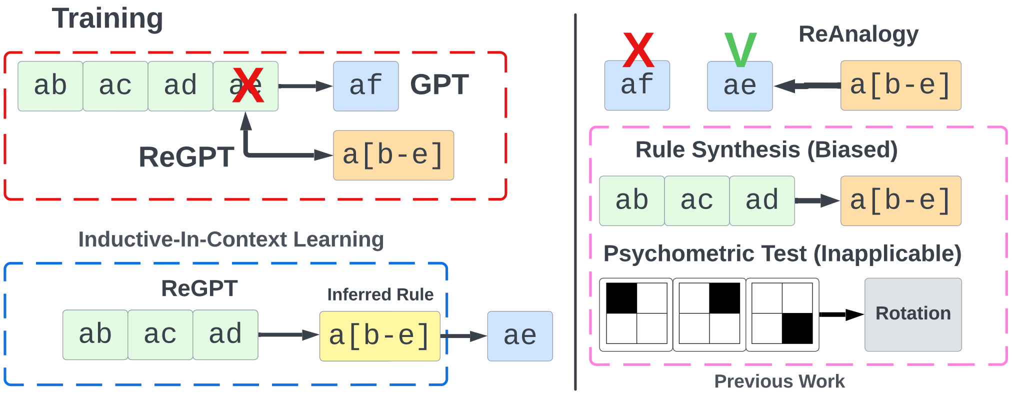

Figure 1: Illustration on the differences between our approach in Training (**left**) and Evaluating (**right**) as compared to previous methods. For ReGPT we inject a **Rule** ('R') during training (bottom row of the red outline) as compared to GPT that is trained with additional **Facts** ('F'). Previous approaches directly optimize for a Rule synthesis task (biased) or use tests that can be inapplicable for LMs (pink outline). During evaluation, we probe ReGPT with IIL (blue outline) to **infer a Rule** ($\hat{R}$) from Facts that we use to generate additional Facts (blue tokens). ReAnalogy evaluate whether the **generated Facts** abide by the ground-truth Rule. Our benchmark can implicitly evaluate on whether a model has learned to infer the underlying Rules as opposed to explicitly evaluate a model optimized for 'Rule Synthesis'. For IIL the soundness of the Rule is unambiguous and as such it improves performance when compared to a baseline model.

on the ground-truth and synthetic datasets for which we have access to their generative process but are not applicable for modern LMs.

To this end we motivate that to evaluate the inductive reasoning abilities of a LM model we need a dataset for which we have access to Rules which we can evaluate unambiguously, *i.e.* the membership of a Fact to a Rule. Similarly, such dataset should be challenging enough to be used with current state-of-the-art models. Additionally, the evaluation protocol should probe the generalization abilities of the model without being subject to the same bias as the question it is trying to answer, for example $P(\texttt{Rule}|\texttt{Facts})$ is an insufficient test when Facts are ambiguous. The same protocol is biased when the model is trained such that to maximize $P(\texttt{R}|\texttt{F})$, which can be referred to as *Rule Synthesis*. As such a holistic approach to evaluation can be more effective at probing the reasoning ability of a LM. For reasons of brevity we use R and F to denote Rules and Facts respectively and use similar notation throughout the text.

Regular Expressions (regex) are Finite Automata and can represent a program (R) that can be used to both synthesize and evaluate membership ($\in$) of an example (F) in an unambiguous way. Current dataset composed of regex are inapplicable for LMs where, they are too small to train on, or contain random patterns. For this reason we propose ReAnalogy benchmark composed of 60,368 regex mined from open-source repositories that contain quasi-natural language. For each expression we generate examples (F) and evaluate the model on the ability to generate additional facts from a *limited* set of **only** positive implication examples, Figure 1. Our benchmark is challenging enough for existing LMs, while it is able to evaluate the analogy making ability of a system in an unambiguous manner.

We evaluate LMs and find they perform at 83% accuracy on this benchmark where previous state-of-the-art on ILP synthesis catastrophically fails. As such, we introduce ReGPT, where we inject Rules in the training sequence to evaluate whether LMs can learn to infer rules. During evaluation we use IIL to prompt a LM to infer Rules that we use to generate Facts and improve performance by as much as 33% compared to a GPT. Our approach can be summarized as induction followed by deduction. We evaluate ReGPT to find that LMs can improve on inductive reasoning and learn to infer Rules implicitly when probed via IIL. We will make our code and dataset publicly available after the review process concludes. Our contributions can be summarized:

- We introduce ReAnalogy, a reasoning benchmark composed of complex quasi-natural language generated from complex regular expressions.
- We introduce ReGPT a GPT model that 'reason' by performing induction to infer the Rules given a limited set of true Facts via *Inductive In-Context Learning* followed by deduction where it generates novel Facts.
- We ablate and analyze the performance of ReGPT and ReAnalogy with results that can help to probe and improve the reasoning abilities of LMs.

2 Related Work

We identify three categories of related works most similar to our work. Work that attempt to evaluate and understand the reasoning abilities of LMs; work that introduce evaluation settings of reasoning for LMs; and work that introduce methods for improving reasoning.

2.1 LM probing

Several works evaluate the reasoning abilities of LMs [Yu *et al.*, 2023] by probing to specific tasks, such as inductive, deductive and abductive reasoning.

[Zhang *et al.*, 2022] use a Language Model as a Logic Programmer to reason over knowledge bases. Their dataset contains query, facts, rules where the goal is, given the query, to find a proof path. Contrary to their method, we evaluate implicitly on how well a model can infer Rules by evaluating the generated Facts. Additionally, IIL deduce rules implicitly, by first performing inductive reasoning and using the inferred rule to generate facts.

Similarly, [Yang *et al.*, 2022] observe that current inductive reasoning datasets and tasks are superficial in probing the inductive reasoning abilities of LMs. They propose DEER, a dataset which contains a list of short natural language facts accompanied by a rule. The dataset is curated by human evaluators that find facts on the internet given a rule. Contrary to our work, their dataset is not publicly available. We evaluate membership of Facts in Rule in an unambiguous manner where DEER is inapplicable in this setting. Additionally, their benchmark, and similar to [Teru *et al.*, 2020], evaluates the probability of inferring the correct Rule given Facts *i.e.* $P(\mathtt{R}|\mathtt{F})$ and as such can be used auxiliary to our setting.

Other work [Misra *et al.*, 2022] evaluate the ability of LMs to perform property induction, *i.e.* if $\mathtt{F}_i \rightarrow \mathtt{R}_1$ and $\{\mathtt{F}_j, \mathtt{F}_i\} \rightarrow \mathtt{R}_2$ then $\mathtt{F}_i \rightarrow \mathtt{R}_2$. Their evaluation setting is orthogonal to our work, while their dataset is artificially generated from natural language Facts with True / False pairs that are inapplicable to be used for evaluating inductive reasoning.

[Telle *et al.*, 2019] evaluate the minimum number of examples (Facts) required for a model to learn a concept (Rule) using a regex dataset. They find that the total size (in length) of the Facts as opposed to their number is a better indicator of complexity to learn a Rule. Similarly, we evaluate the relationship in learning Rules from Facts to arrive at similar conclusions as we increase the number of facts we use to infer Rules, Table 2. Contrary to their setting we evaluate the inferred Rules by using them to generate additional Facts. Additionally, `ReAnalogy` contain quasi-natural language as opposed terminal 0 and 1 that are constructed from a restrict set of regex.

[Yang *et al.*, 2023; Liu *et al.*, 2022] supplement our analysis where they identify that natural language descriptions alone can be inapplicable to evaluate the reasoning ability of LMs. They combine descriptions (Facts) in natural language with automaton-based representations (Rules) that can be formally verified. Contrary to our work, we evaluate the Facts generated by the Rules inferred via IIL.

Similar to our work, [Min *et al.*, 2022] probe the mechanism behind in-context learning and the aspects of the demonstration that contribute to end-task performance. They find that ground-truth demonstrations (Rules) are not required. Similarly, we find that explicitly learning (Rules) is not required for a LM to learn to generate Facts where a `GPT` model performs with 83% accuracy. Additionally, our method, IIL, improves performance by as much as 33% where Rules are injected during training and the model is probed via IIL.

2.2 Benchmarks

CLUTRR [Sinha *et al.*, 2019] evaluate the ability of a model to provide an answer (Fact) to a question in an in-context learning setting. Contrary to evaluating $P(A|Q)$ we evaluate the ability of a model to generate additional facts. Similar to 'Rule Synthesis' we find that a benchmark that evaluates $P(A|Q)$, cannot provide a holistic view. Additionally, their dataset is inapplicable in our setting and the two benchmark are orthogonal and as such can be used auxiliary to each other.

[Bhagavatula *et al.*, 2019] introduce an 'Abductive Reasoning' benchmark where given observations (Facts) they evaluate how well a model can generate (Rules) with natural language pairs. This benchmark can be seen as similar to a Rule Synthesis evaluation protocol such as, $maxP(R|F)$. Work by [Cornelio and Thost, 2021] is similar in that regard as well.

[Yang and Deng, 2021] identify the gap between natural language datasets and first-order logical rules. They propose a dataset that expresses both natural language as well as formal logic, quasi-natural language. In contrast, `ReAnalogy` use regex to generate the examples where a DFA can generate more complex expressions than first-order logic of limited terminals.

[Mitchell, 2021] propose to evaluate the reasoning ability of AI on psychometric tests designed for humans. Most similar to our evaluation setting, Raven's Progress Matrix evaluates whether a model can learn to complete a sequence of complex visual patterns. Such setting can be inapplicable to LM as they are not agnostic to prior knowledge *i.e.* identification of shapes, which can be a source of noise during evaluation. For example, it would be hard to diagnose whether the model performs poorly at recognizing rotations as opposed to reasoning. Additionally, such priors can be natural for humans but are known failure points for ML models [Geirhos *et al.*, 2018; Li *et al.*, 2018].

2.3 Reasoning Improvements

[Chen *et al.*, 2020] propose a method to synthesize regular expressions from positive and negative examples and natural language description. The main component of their method is the multimodal approach that combined natural language with AlphaRegex [Lee *et al.*, 2016]; a symbolic search based approach on the regular expressions. Our work are orthogonal, as their method does not work under ambiguity.

Similarly, [Wei *et al.*, 2022] introduce a method for eliciting reasoning via in-context learning of LMs. The reasoning ability of the model increases when the model is demonstrated additional examples. Their analysis supplements our work and to work by [Jaimovitch-Lopez *et al.*, 2021]. Contrary, to [Jaimovitch-Lopez *et al.*, 2021], where they use input output pairs to learn the concepts, we only use input pairs (Facts). Additionally their dataset are artificial sequences of 0s and 1s.

[Rytting and Wingate, 2021] explore a neuro-symbolic approach of using the learned representations of a LM to train symbolic engine. Similarly our approach can also be seen as neuro-symbolic with the use of IIL to infer unambiguous Rules that are used to synthesize Facts. Their analysis is auxiliary to our work, while their method uses a reasoning engine on templates of natural language meant to Synthesize Rules.

3 Preliminaries

3.1 Inductive Reasoning

For Language Models, reasoning is the process of drawing conclusions (Rules) from a set of observations (Facts). In general, it consists of inductive reasoning, deductive reasoning or abductive reasoning. **Inductive reasoning** is to reach generalized conclusions (Rules) based on observed specific instances (Facts). When there is insufficient evidence to reach a conclusion, abductive inference involves making educated hypotheses from the incomplete observations. Mathematically, inductive reasoning can be presented in Facts $\rightarrow$ Rules. On the contrary, **deductive reasoning** uses general principles or rules to make specific conclusions, *i.e.* Rules $\rightarrow$ Facts.

For example, during inductive reasoning, given the Facts: abcd, abzz, and abeee; we can find a Rule$_1$ that they all 'start with ab'. Formally, {abcd, abzz, abeee} $\rightarrow$ Rule$_1$. While for deductive reasoning, given the Rule 'strings that start with ab', we can present Facts {abcd, abzz, abeee}. ReGPT performs inductive reasoning where we infer the Rules from a limited set of Facts followed by deductive reasoning where we generate Facts from the inferred Rules *i.e.* $P(\mathtt{R}|\mathtt{F}) \rightarrow P(\mathtt{F}_{i+1}|\mathtt{R})$. Contrary, GPT generates new Facts from the previous Facts, *i.e.* $P(\mathtt{F}_{i+1}|\mathtt{F}_i)$.

3.2 Regular Expressions

Regular expressions (regex) are automata and provide a formal representation of a language. Although not equivalent, Formal Languages have been applied in Natural Language in the past. More recent work [Hahn *et al.*, 2022] can find equivalent approximations between the two. While it is no mathematical equivalence between Natural Languages and Formal Languages, we refer to their intersection as *quasi-natural* language.

Regex are algebraic approaches to search for and manipulate text based on a set of rules. For example, the Rule$_1$ in the previous subsection, 'starting with ab', can be represented as ^ab.*. The syntax of regular expressions include operators such as ('.','^', '\$','|','[ ]','[^]','*','+','$\{m, n\}$'), that provide control for iteration (*i.e.* $*$ matches any of the previous characters any number of times), logic (*i.e.* | can be used as an OR operator), while other characters are considered terminals. Rules are combined to form complex relationships with terminal character symbols such as letter or strings. For example "[dog,]." would match "dog,dog,dog...". Complex strings can be matched with sufficiently complex expressions.

Finite Automata (FA) contain transitions between states. For example, the expression "[a-z]*" defines all strings that contain any number of lower case characters from the English alphabet. The states would be characters 'a' through 'z' and a terminal state. Nondeterministic FA (NFA) define probability of transition between states, *i.e.* probability of 'a' followed by 'a'. We can interpret regex in this work as NFA where at each state we randomly sample a transition at an even probability. ReAnalogy is the only dataset that supports the full regex functionality and for unicode character-set; an artifact of our efficient implementation in generating regexes.

3.3 In-Context Learning

In-Context Learning (ICL) [Brown *et al.*, 2020] is a paradigm that trains LMs with a limited set of demonstrations. Different from fine-tuning, which updates parameters of pre-trained networks by training on datasets of new tasks, ICL requires no parameter optimization for a downstream tasks. Instead, ICL uses as input a 'prompt' that can guide the model to perform a task. Given that a LM has been pre-trained on large-scale corpus, it can perform tasks that were not contained in the training data such as generating new data via *generative sampling*. We find that ICL is the correct setting to evaluate the generalization ability of a LM as it is not biased in training the model for a specific task, *i.e.* Rule Synthesis. While the prompt can be used to guide a the pre-trained model via next-token prediction on Facts or Rules and generate additional Rules or Facts.

Generative Sampling uses a scoring function to calculate the probability of a sequence token given all previous tokens (*i.e.* the prompt). There are several ways to perform sampling, such as beam-search over all candidate options. To avoid the bias introduced by the sampling method we evaluate the generative ability of a model using 'top-k' sampling, where we randomly sample from the top-k most probable tokens.

4 Method

4.1 ReAnalogy

As the current datasets with regex lack sufficient complexity to effectively evaluate the performance of LMs and probe their limitations, we contribute a new dataset ReAnalogy, which consists of 60,368 regex acquired from open-source repositories and augmented with 43,896 python regex from [Davis *et al.*, 2019].

Similarly to [Davis *et al.*, 2019], we find that the majority of the expressions collected form the web are not compilable, contain bugs, vulnerabilities, are too large, or lack complexity (e.g., when a large portion of the expression is a fixed pattern, usually code). We mine additional examples from open-source repositories using the GitHub API and search for code that utilizes the 're' library and we extract the string literals used for those expressions. We find that a significant portion of the expressions we mine contain natural language and code *i.e.*

```
expr1="While linting files .*"
```

We use the expressions to model Rules, that describe Facts, such as text that match the Rule pattern. There are several avenues in finding Facts that match the Rule. The expression has a specific use-case and it can be difficult to find multiple string examples *in-the-wild* that provide an exact match at scale *i.e.* matching *expr1* with a natural language corpus. We attempt to use expressions directly to large corpus on dataset like Common Crawl[1] and Github repositories. Based on sub-sample of our results it is computationally expensive to perform exhaustive search and did not lead to sufficient matches from a limited set of examples we sampled.

As such, we generate Facts by randomly traversing the regex execution graph where we modify XEGER[2]. We per-

[1] https://commoncrawl.org/

[2] https://github.com/crdoconnor/xeger

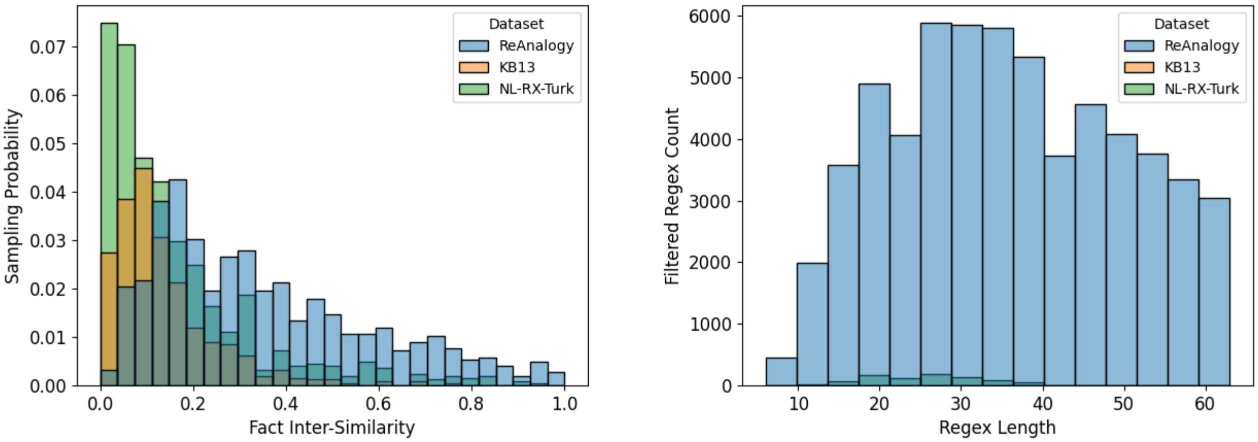

Figure 2: Quantitative evaluation of `ReAnalogy` **Left** Probability of sampling examples by a regex for a given inter-similarity range and as measured by the Jaccard index. A higher inter-similarity signifies larger distance between generated Facts where 0 would be as good as random *i.e.* generated using the expression ".*". On the contrary, Facts with high inter-similarity would be generated by fixed regex *i.e.* "hello world". For `ReAnalogy` the Fact inter-similarity is evenly distributed signifying diversity and complexity of generated expressions, with a median of 0.70. Contrary KB13 and Deep-Regex that have 0.90 and 0.88 respectively. **Right** Distribution of the filtered regex by length for each dataset. `ReAnalogy` is significantly larger than KB13 [Kushman and Barzilay, 2013] and Deep-Regex [Locascio *et al.*, 2016] where the latter dataset are dwarfed by the size of `ReAnalogy` distribution. Additionally, `ReAnalogy` has diversity on the size of expressions that can be observed by the even distribution in the histogram.

form modifications to set a stop condition for some operations *i.e.* ‘*’ that limit the number of terminal repetitions to 10. Additionally, the existing library provides poor support for specific binary terminals *i.e.* unicode characters. The vocabulary of `ReAnalogy` is composed of 242 terminals. Our implementation is performant enough that the generation can be performed on-the-fly and is not the bottleneck in training a LM.

Filtering To enhance the diversity and complexity of the generated data, we filter for regex that can not produce more than 10 unique examples over multiple random repetitions. The goal of filtering is to avoid evaluating or training on very simple examples, such as `[a,a,...]` → `a`. For example, regex for fixed strings such as ’GRID_PPEM’ were removed from the dataset. While there were patterns that were not compilable in Python such as "\\G . . . " due to lack of support of operator \\G. Additionally, operators such as "\b" (backspace) are not applicable in our generative setting.

4.2 Fact Score

We define a metric to evaluate the membership of generated Facts implicitly by a Ground Truth Rule under ambiguity of reasoning. We find that directly predicting and evaluating the Rule is biased, as a model can learn to synthesize a Rule from Facts but does not answer whether the model can learn to do the same under more general conditions. Throughout this work we use the term $P(\mathtt{R}|\mathtt{F})$ to describe this setting.

As such, we propose to evaluate generated Fact with the ground-truth Rule ($\mathtt{R}^*$) as opposed to performing equivalence comparison of an inferred Rule ($\hat{\mathtt{R}}$) and $\mathtt{R}^*$. First, two Rules can be equivalent but expressed differently. In the context of regular expressions it can be trivial to evaluate *e.g.* `[a-zA-Z0-9_] == \w`, but in natural language rule equivalence can be ambiguous. Second, more than one Rule can interpret the Facts, *i.e.* $\hat{\mathtt{R}} \rightarrow \{F\}$, $\mathtt{R}^* \rightarrow \{F\}$ and $\hat{\mathtt{R}} \neq \mathtt{R}^*$. Under an ambiguous setting, $\hat{\mathtt{R}}$ can be considered correct in the context of probing the ability of a model to reason. For example `F = {ab, ac, ad}` → $\hat{\mathtt{R}}$ has many ambiguous and correct interpretations $\hat{\mathtt{R}} \in \{\mathtt{a}\star, \ldots, \mathtt{a[bcde]}\}$.

We disambiguate the evaluation and contrary to comparing between Rules, we propose to compare if the generated Facts satisfy $\mathtt{R}^*$; e.g., whether $\mathtt{R}^* \rightarrow \mathtt{ae}$. The Fact Score (FS) is computed as the mean accuracy score of $\mathtt{R}^* \rightarrow \mathtt{F}_{i+1}$ and calculated as:

$$\mathtt{FS} = P(\mathtt{R}^* \rightarrow \mathtt{F}_{i+1}|\{\mathtt{F}_1, .., \mathtt{F}_i\}); \forall \mathtt{R}^* \in \mathtt{ReAnalogy} \quad (1)$$

Fact Score can be extended beyond regex to natural language whereby $\mathtt{R}^* \rightarrow \mathtt{F}_{i+1}$ can be evaluated by an auxiliary model or human feedback. However, we limit our work to regex as a proving ground for our evaluation protocol, as a regex can be unambiguously evaluated without significant manual effort in curating a dataset.

4.3 `ReGPT`

We use `ReGPT` to implement Inductive In-Context Learning (IIL) and evaluate on whether LMs can learn to infer Rules given Facts. The goal of `ReGPT` is to probe the reasoning ability of LMs and not to introduce a novel architecture. As such, we compare with a baseline `GPT`. Both models are trained on a set of Facts. The difference for `ReGPT` is that we replace a Fact with a Rule injected in the training sequence but without direct or explicit supervision to associate the Rule with the Facts, Figure 1. To ablate the influence of model structure, `ReGPT` uses the same backbone as `GPT`, *i.e.* Transformer. `ReGPT` is used to evaluate two hypothesis:

1. Do LMs learn to infer Rules from the data?

2. Do LMs learn to associate Facts and Rules implicitly?

We evaluate 1. with GPT probed with ReAnalogy to generate Facts and evaluated using FS, Section 4.2 with 83% performance. We evaluate 2. with Inductive In-Context Learning (IIL) and probe ReGPT to generate rules and evaluate those rules by using them to generate facts. There are two aspects of our approach that can extend beyond our work. Firstly, generating Rules is unambiguous and even in natural language. *i.e.* inferring a rule such that "dogs are mammals" or "octopuses are mammals" are both unambiguous but not necessarily correct. Secondly, evaluating whether the inferred rule is *correct* (given the facts) is evaluated by the ability of the model to produce names of mammals given a list of name of animal *i.e.* [bird, . . . , horse, $\texttt{Fact}_{i+1}$] → $\texttt{Fact}_{i+1} =$ "octopus" or "dog". The evaluation is performed on whether the generated fact is correct. *i.e.* "mammal" → "octopus".

The advantage of this approach is that the evaluation metric used from Section 4.2, can account for ambiguity of the probing facts. We find that the performance of a model is influenced both by the quality and quantity of facts, Section 5.2. Inductive In-Context Learning can probe on whether the model has successfully learned associations between concepts and whether it can reason on this association and extrapolate beyond the evidence present in the prompt.

Our approach has similarities to *Neuro-Symbolic* methods, where we evaluate the correctness of the inferred Rule in an unambiguous manner, for example on whether an expression can be compiled or a natural language syllogism follows a template. While generating facts can be seen as deductive reasoning. As such our approach can be summarized as induction (inferring sound Rules from Facts) followed by deduction (inferring Facts from Rules).

5 Experiments

We use ReAnalogy composed of 60,368 expressions that we use to synthesize examples on the fly during training of both GPT and ReGPT. Based on the insights from [Telle *et al.*, 2019] and to avoid the learning bias from the size of examples and regular expressions we limit their length to 64 characters. The total sequence length can reach 1024 characters which is identical to [Radford *et al.*, 2019]. In Section 5.1 we evaluate and compare ReAnalogy with a baseline method Deep-Regex[Locascio *et al.*, 2016]. We evaluate on their dataset NL-RX-Turk and KB13 [Kushman and Barzilay, 2013] that are regex dataset used in literature and show both the difference in complexity, size as well as applicability of each dataset to LMs. Next in Section 5.2 we evaluate and compare ReGPT with the baseline GPT to evaluate the effects of Inductive In-Context Learning using Fact Score from Section 4.2. We use Adam with 0.001 LR for all our experiments. For reasons of brevity on hyper-parameters and implementation details we refer the reader to our open-source repository[3] that is well-documented.

Model	Train Dataset	Eval. Dataset	FS
Deep-Regex	NL-RX-Turk	NL-RX-Turk	0.58
ReGPT	NL-RX-Turk	NL-RX-Turk	0.66
ReGPT	ReAnalogy	NL-RX-Turk	**0.80**
Deep-Regex	KB13	KB13	0.65
ReGPT	KB13	KB13	0.20
ReGPT	ReAnalogy	KB13	**0.85**

Table 1: Cross-Evaluation and comparison of ReGPT when trained and evaluated using existing dataset from literature. NL-RX-Turk and KB13 are too small and not complex enough to be used with LMs, Figure 2. When trained and evaluated on the same dataset ReGPT performs poorly compared to baseline Deep-Regex. ReGPT performance is proportional to the complexity and size of the dataset where KB13 $\ll$ NL-RX-Turk. When ReGPT is pre-trained on ReAnalogy it outperforms the baseline. Our results highlight the in-applicability of existing dataset in training LMs, as well as the effectiveness of ReGPT when compared to a baseline.

5.1 ReAnalogy

For the experiments in this section we train 8 ReGPT of different sizes [2,6,24] layers on Datasets Deep-Regex and KB13 as a way to evaluate the applicability of LMs on the respective datasets. We found that smaller models worked better for the smaller dataset, and as such we report the best performance from the range. We compare with a ReGPT model of 24 layers pre-trained on ReAnalogy and evaluated on each respective dataset. We find that both NL-RX-Turk and KB13 are too small to be used with a LM where their in-dataset evaluation performance correlates with their size 824 for KB13 and 10000 for Deep-Regex with 0.20 and 0.66 FS respectively, Table 1. On the contrary, evaluating the same dataset using a pre-trained model leads to contrary results where ReGPT performs better for a simpler dataset KB13. We conclude that both datasets are inapplicable to use with a LM. Additionally, pre-trained ReGPT outperforms a baseline method on the datasets, where Deep-Regex also suffers in performance on KB-13 due to the small size.

Next we evaluate the characteristics of the two datasets qualitatively and quantitatively. Figure 2 shows the distribution of inter-similarity between facts and the distribution of the length for the expressions present in the filtered dataset. The Facts sampled by ReAnalogy have relative evenly distributed inter-similarity, while those in KB13 and Deep-Regex are concentrated in intervals with low inter-similarity. Therefore, ReAnalogy is the only that has a large number of diverse expressions with complex characteristics. During our filtering process described in Section 4.1 we find 5% of 'fixed' string regex in ReAnalogy compared to 0.5% for KB13 and 0.3% Deep-Regex. The observation is an artifact of expressions that contain natural language, *i.e.* Figure 3.

5.2 Inductive In-Context Learning

We compare and evaluate a GPT model with 24 layers and we ablate the importance of the number of facts as well as the sampling 'freedom' to generate facts ('top-k'). We train 8

[3] https://github.com/fostiropoulos/reanalogy

Listing 1: KB13 Examples

```
(.*[0-9].*){5,}
1.  Er9?=0mQL92:?$)\\BzG 1 ...
2.  xL'\2r{IngtO\f6Kd<R<1JM...
3.  2fYz6FX\XW6WH#a?=\rv0>...
```

Listing 2: NL-RX-Turk Examples

```
.*([0-9])|(dog)|(.).*
1. 23j
2. XrK%0
3. dog
```

Listing 3: ReAnalogy Examples

```
applying.*?jquery.*?script
1. applying+WjqueryaK-6py|w9$script
2. applyingV\f>\/Rk{qkTjqueryscript
3. applyings\tH'jquery#script
```

Figure 3: Qualitative evaluation on samples between KB13, NL-RX-Turk and ReAnalogy datasets. Compared to ReAnalogy, other dataset contain mostly random expressions for which the generated examples do not contain meaningful semantics. NL-RX-Turk contain expressions that contain natural language, while ReAnalogy is the only that contains complex quasi-natural language expressions.

different models for the experiments in this section until the validation loss plateaus (which differs for different 'N. Facts').

Our experiments on the number of facts provided to the model provide two insights. First, the performance of GPT plateaus and decreases for 'N. Facts' > 5 and 'Top-K'$= 2$ sampling; first row of Table 2. Our observation could be an artifact of the inability of the model to draw associations between Facts as complexity from additional facts arises.

In contrast, ReGPT with IIL outperforms a GPT under equivalent settings. When increasing 'N. Facts', the performance of ReGPT improves, as shown in the second row of Table 2. Additionally, ReGPT is *example efficient* where for fewer N.Facts the performance gap is larger between GPT and ReGPT with 0.54 and 0.71 FS respectively.

Top-K sampling is used to randomly sample from the K most likely terminals. Larger K result in samples with higher diversity. GPT benefits from the diversity of the generated facts where larger K improve FS, while the contrary can be observed for ReGPT where larger K can harm performance. Our results suggest that there is a *risk-creativity* trade-off where higher creativity can be exhibited by LMs at the risk of improbable Rules. This is an artifact that diversity in generation of rules leads to improbable outcomes and we hypothesize it could also explain hallucinations [Ji *et al.*, 2023], where on the one hand LMs display 'creativity' in generating impressive content and often catastrophically fail by producing improbable text. We conclude that hallucinations can be explained by the poor inference of the underlying structure of the data (the inferred Rules).

Top-k	N. Facts (FS)	2	5	8	12
2	GPT	0.54	0.71	0.71	0.70
	ReGPT	**0.71**	**0.80**	**0.83**	**0.86**
6	GPT	0.55	0.74	0.75	0.77
	ReGPT	0.69	0.78	0.80	0.85
10	GPT	0.58	0.75	0.79	0.81
	ReGPT	0.66	0.79	0.79	0.84
14	GPT	0.60	0.76	0.80	0.83
	ReGPT	0.66	0.78	0.80	0.84
18	GPT	0.61	0.77	0.82	0.83
	ReGPT	0.65	0.79	0.79	0.83

Table 2: Comparison of GPT and ReGPT evaluated with IIL using FS from Section 4.2, where we evaluate a generated Fact by the ground-truth Rule. We generate Facts using Top-K sampling with different hyper-parameters. We find that increasing Top-K can increase the performance of GPT, but harms the performance of ReGPT. Although ReGPT consistently outperforms GPT, we find this to be an artifact of generating more improbable Rules. Using IIL improve the Fact-Score accuracy by a significant margin 9% on average and as much as by 33%.

6 Discussion

Understanding the performance of LMs requires principled methods of evaluation that can disambiguate the inherent ambiguity of realistic reasoning tasks. One can argue that a limitation of our work is the quasi-natural language while we find that to be necessary and a novel aspect of our work. Quasi-natural language does not contain ambiguous reasoning arguments while being close enough to a natural language for a principled but challenging evaluation setting. We motivate that there should be a diverse set of benchmarks and datasets for evaluation of LMs. We introduce and evaluate IIL on a quasi-natural language setting as a starting point and find that additional work and open problems in this area would need to be addressed before applying Inductive In-Context Learning directly in natural language. Additional manual effort will be required for creating and curating a natural language dataset specific to IIL. We motivate that such benchmarks and dataset should be open and accessible to everyone by open-sourcing ReAnalogy and urge for other researchers to do the same. Our work is in the intersection of LMs and ILP and bares similarities to neuro-symbolic approaches. While a lot of work is emerging in this domain it is still unclear how to best compare between methods in a principled manner.

7 Conclusion

We find that previous benchmark and evaluation settings cannot provide a holistic view of the reasoning ability of a LM. Previous work explicitly optimizing the performance of a LM for the reasoning task and can be inapplicable in evaluating on whether reasoning emerges in LMs. We propose an evaluation setting using Inductive In-Context Learning where we probe the reasoning abilities of a LM by evaluating the generated Facts as opposed to directly evaluating the Rules it generates. We find that LM can perform at our benchmark without explicit supervision on the reasoning task. We use IIL with ReGPT and evaluate whether a LM can implicitly learn to associate Facts with Rules where we improve 'reasoning' performance of a GPT by 33%.

Acknowledgement

This work was supported by C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), DARPA (HR00112190134) and the Army Research Office (W911NF2020053). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

References

[Bender *et al.*, 2021] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[Bhagavatula *et al.*, 2019] Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*, 2019.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

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

[Chen *et al.*, 2020] Qiaochu Chen, Xinyu Wang, Xi Ye, Greg Durrett, and Isil Dillig. Multi-modal synthesis of regular expressions. In *Proceedings of the 41st ACM SIGPLAN conference on programming language design and implementation*, pages 487–502, 2020.

[Cornelio and Thost, 2021] Cristina Cornelio and Veronika Thost. Synthetic datasets and evaluation tools for inductive neural reasoning. In *Proceedings of the 30th International Conference on Inductive Logic Programming, ILP2020-21 @ IJCLR*, 2021.

[Cropper *et al.*, 2022] Andrew Cropper, Sebastijan Dumančić, Richard Evans, and Stephen H Muggleton. Inductive logic programming at 30. *Machine Learning*, pages 1–26, 2022.

[Davis *et al.*, 2019] James C Davis, Louis G Michael IV, Christy A Coghlan, Francisco Servant, and Dongyoon Lee. Why aren't regular expressions a lingua franca? an empirical study on the re-use and portability of regular expressions. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 443–454, 2019.

[Geirhos *et al.*, 2018] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[Hahn *et al.*, 2022] Christopher Hahn, Frederik Schmitt, Julia J Tillman, Niklas Metzger, Julian Siber, and Bernd Finkbeiner. Formal specifications from natural language. *arXiv preprint arXiv:2206.01962*, 2022.

[Jaimovitch-Lopez *et al.*, 2021] Gonzalo Jaimovitch-Lopez, David Castellano Falcón, Cesar Ferri, and José Hernández-Orallo. Think big, teach small: Do language models distil occam's razor? *Advances in Neural Information Processing Systems*, 34:1610–1623, 2021.

[Ji *et al.*, 2023] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[Kushman and Barzilay, 2013] Nate Kushman and Regina Barzilay. Using semantic unification to generate regular expressions from natural language. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 826–836, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[Lee *et al.*, 2016] Mina Lee, Sunbeom So, and Hakjoo Oh. Synthesizing regular expressions from examples for introductory automata assignments. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, pages 70–80, 2016.

[Li *et al.*, 2018] Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai. Deep rotation equivariant network. *Neurocomputing*, 290:26–33, 2018.

[Liu *et al.*, 2022] Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

[Locascio *et al.*, 2016] Nicholas Locascio, Karthik Narasimhan, Eduardo DeLeon, Nate Kushman, and Regina Barzilay. Neural generation of regular expressions from natural language with minimal domain knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1918–1923, Austin, Texas, November 2016. Association for Computational Linguistics.

[Min *et al.*, 2022] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and

Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[Misra *et al.*, 2022] Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. A property induction framework for neural language models. *arXiv preprint arXiv:2205.06910*, 2022.

[Mitchell, 2021] Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

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

[Rytting and Wingate, 2021] Christopher Rytting and David Wingate. Leveraging the inductive bias of large language models for abstract textual reasoning. *Advances in Neural Information Processing Systems*, 34:17111–17122, 2021.

[Sinha *et al.*, 2019] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019.

[Telle *et al.*, 2019] Jan Telle, Jose Hernandez-Orallo, and Cèsar Ferri. The teaching size: computable teachers and learners for universal languages. *Machine Learning*, 108, 09 2019.

[Teru *et al.*, 2020] Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pages 9448–9457. PMLR, 2020.

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

[Yang and Deng, 2021] Kaiyu Yang and Jia Deng. Learning symbolic rules for reasoning in quasi-natural language. *arXiv preprint arXiv:2111.12038*, 2021.

[Yang *et al.*, 2022] Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. Language models as inductive reasoners. *arXiv preprint arXiv:2212.10923*, 2022.

[Yang *et al.*, 2023] Yunhao Yang, Jean-Raphaël Gaglione, Cyrus Neary, and Ufuk Topcu. Automaton-based representations of task knowledge from generative language models. *arXiv preprint arXiv:2212.01944*, 2023.

[Yu *et al.*, 2023] Fei Yu, Hongbo Zhang, and Benyou Wang. Nature language reasoning, a survey. *arXiv preprint arXiv:2303.14725*, 2023.

[Zhang *et al.*, 2022] Hanlin Zhang, Yi-Fan Zhang, Li Erran Li, and Eric Xing. The impact of symbolic representations on in-context learning for few-shot reasoning. *arXiv preprint arXiv:2212.08686*, 2022.