

# Language Models as Inductive Reasoners

Zonglin Yang<sup>1\*</sup> Li Dong<sup>2</sup> Xinya Du<sup>3</sup> Hao Cheng<sup>2</sup>  
Erik Cambria<sup>1</sup> Xiaodong Liu<sup>2</sup> Jianfeng Gao<sup>2</sup> Furu Wei<sup>2</sup>

<sup>1</sup> Nanyang Technological University

<sup>2</sup> Microsoft Research

<sup>3</sup> University of Texas at Dallas

{zonglin.yang, cambria}@ntu.edu.sg {lidong1,cheng.hao,xiaodl,jfgao,fuwei}@microsoft.com xinya.du@utdallas.edu

## Abstract

Inductive reasoning is a core component of human intelligence. In the past research of inductive reasoning within computer science, logic language is used as representations of knowledge (facts and rules, more specifically). However, logic language can cause systematic problems for inductive reasoning such as disability of handling raw input such as natural language, sensitiveness to mislabeled data, and incapacity to handle ambiguous input. To this end, we propose a new task, which is to induce natural language rules from natural language facts, and create a dataset termed DEER containing 1.2k rule-fact pairs for the task, where rules and facts are written in natural language. New automatic metrics are also proposed and analysed for the evaluation of this task. With DEER, we investigate a modern approach for inductive reasoning where we use natural language as representation for knowledge instead of logic language and use pretrained language models as “reasoners”. Moreover, we provide the first and comprehensive analysis of how well pretrained language models can induce natural language rules from natural language facts. We also propose a new framework drawing insights from philosophy literature for this task, which we show in the experiment section that surpasses baselines in both automatic and human evaluations.

## Introduction

Inductive reasoning is to reach to a hypothesis (usually a rule that explains an aspect of the law of nature) based on pieces of evidence (usually observed facts of the world), where the observations can not provide conclusive support to the hypothesis (Salmon 1989). It is ampliative, which means that the hypothesis supports more than mere reformulation of the content of the evidence (Norton 2005). An example is shown in Table 1 that after observing three carnivorous plants each having a trapping structure, one might reach to a hypothesis (rule) that every carnivorous plant has a trapping structure. Inductive reasoning was firstly proposed by Aristotle in the 4th century B.C. in his *Posterior Analytics* (Aristotle 1994). Since then it is used as a fundamental tool to obtain axioms, and therefore subjects can be developed from these axioms. It is also recognized as a core component of human intelligence (Mercier 2018).

Past research works on inductive reasoning within computer science are investigated by Inductive Logic Programming (ILP) (Muggleton and De Raedt 1994). ILP investigates the inductive construction of first-order logic rules from examples and background knowledge (Muggleton and De Raedt 1994). However, ILP uses logic language as representation and uses symbolic algorithms as method, which results in systematic disadvantages (Cropper et al. 2022). Specifically, ILP systems heavily rely on human effort, since it typically assumes that the input has already been preprocessed into symbolic declarative form, otherwise ILP systems cannot handle raw inputs such as natural language and images. In addition, ILP systems are very sensitive to label error and ambiguity in data, since the final induced rules are required to satisfy all input facts, and symbolic systems can not recognize different symbols with the same meaning (e.g. be capable of, has the capability of, be able to).

To overcome the challenges above, we present a novel paradigm for inductive reasoning based entirely on natural language, i.e., inducing natural language rules from natural language facts. In particular, we create a first-of-its-kind natural language inductive reasoning dataset named DEER containing 1.2k rule-fact pairs<sup>1</sup>. Specifically, human-written natural language rule sentences are first collected. Based on the collected rules, we then ask human annotators to collect existing natural language texts as facts from the web where each fact can be possibly enough to induce the given rule. With this dataset, we investigate a modern approach to inductive reasoning where both facts and rules are in natural language, and pretrained language models (PLMs) are used as the inductive reasoner to induce natural language rules from natural language facts. Note that the inductive reasoning considered in this paper has several distinctions considered by other reasoning tasks over text (Clark, Tafjord, and Richardson 2020; Bhagavatula et al. 2020; Sinha et al. 2019). We defer a more detailed discussion to section 2.

With natural language as representation and PLMs as the reasoner, such an inductive reasoning system can avoid the systematic disadvantages of logic language and symbolic methods. Specifically, with natural language as representation, it can naturally handle raw input as natural language text. In addition, different from symbolic methods, PLMs

\*Contribution during internship at Microsoft Research.

<sup>1</sup>We will release our code and data after publication

Short fact 1	Short fact 2	Short fact 3	Rule
<p>The Venus flytrap is a <b>carnivorous plant</b> native to subtropical wetlands on the East Coast of the United States in North Carolina and South Carolina. It catches its prey—chiefly insects and arachnids—with a <b>trapping structure</b> formed by the terminal portion of each of the plant’s leaves, which is triggered by tiny hairs on their inner surfaces.</p>	<p>Pitcher plants are several different <b>carnivorous plants</b> which have modified leaves known as <b>pitfall traps</b>—a prey-trapping mechanism featuring a deep cavity filled with digestive liquid. The traps of what are considered to be “true” pitcher plants are formed by specialized leaves. The plants attract and drown their prey with nectar.</p>	<p>Drosera, which is commonly known as the sundews, is one of the largest genera of <b>carnivorous plants</b>, with at least 194 species. The <b>trapping</b> and digestion mechanism of Drosera usually employs two types of glands: stalked glands that secrete sweet mucilage to attract and ensnare insects and enzymes to digest them, and sessile glands that absorb the resulting nutrient soup.</p>	<p>If a plant is carnivorous, then it probably has a trapping structure.</p>

Table 1: An example of inductive reasoning in DEER dataset. We embolden the words in facts that contain the key information to induce this rule (just to explain the relation between facts and rule, in DEER there’s no special word annotations for fact).

contain knowledge via pretraining (Davison, Feldman, and Rush 2019) and use embedding for concepts (Mikolov et al. 2013), making it less affected by input errors (Meng et al. 2021) and more robust to paraphrasing.

Based on the proposed dataset, we study the PLM’s ability to induce (generate) natural language rules from natural language facts. Specifically, we analyze the performance of PLMs to induce natural language rules based on different First-Order Logic (Smullyan 1995) rule types and topics (e.g., zoology, botany, history), with varying input facts and PLM model sizes.

We also propose a new framework for this task, named chain-of-language-models (CoLM) which is shown in Figure 1. It draws insights from the requirements of rule induction in philosophy literature (Norton 2005). Specifically, CoLM consists of five modules all based on PLMs, where one model proposes rules (rule proposer M1), and the other four models (M2, M3, M4, M5) each classify whether a generated rule satisfies one particular requirement of induction. In our experiments, we find that our framework surpasses the baselines in terms of both automatic and human evaluations.

To sum up, our contributions are three-fold.

- We propose a new paradigm (task) of inducing natural language rules from natural language facts, which naturally overcomes three systematic disadvantages of past works on inductive reasoning. In particular, we create a first-of-its-kind natural language inductive reasoning dataset DEER containing 1.2k rule-fact pairs, where fact and rule are both written in natural language. New automatic metrics are also proposed for task evaluation, which shows strong consistency with human evaluation.
- We provide the first and comprehensive analysis of how well PLMs can induce natural language rules from natural language facts.
- Drawing insights from philosophy literature (Norton 2005), we propose a framework for inductive reasoning. Empirically, we show that it surpasses baselines substantially in both automatic and human evaluations.

## Related Work

### Definition of Inductive Reasoning

All non-fallacious arguments (an argument consisting of a premise and a conclusion) can be classified as a deductive argument or inductive argument (Flach and Kakas 2000). If

the premise can provide conclusive support for the conclusion, which means that if the premises of the argument were all true, it would be impossible for the conclusion of the argument to be false, the argument is called a deductive argument. Similarly, if the premise can only provide partial support for the conclusion, the argument is called inductive argument (Salmon 1989). Conclusions of inductive arguments amplify or go beyond the information found in their premises (Salmon 1989). In this paper, we call the premises as “fact”, and conclusions as “rule”.

### Inductive Reasoning & Neural Networks

Sinha et al. (2019) proposes CLUTRR dataset, which requires NLU system to make classification on kinship relations between characters in short stories. In many examples of this dataset, a set of facts that can make conclusive support to the target kinship relation is included in background information as input for each target relation, hence from the philosophical definition (Salmon 1989), these examples require to perform deductive reasoning more than inductive reasoning. Misra, Rayz, and Ettinger (2022) investigates using neural networks for the classification of synthetic language of sentences containing an object and a property. In contrast to our work, they only focus on synthetic language and classification problems. In addition, their classification targets are not more general rules, and most are irrelevant facts compared to input facts. One line of research that is related to induction is “inductive relation induction” (Teru, Denis, and Hamilton 2020). However, this task focuses on prediction of relation that involves unseen entities, which only involves an induction from specific entities to specific entities, where we focus on the induction from specific entities or individual phenomena to general knowledge. Yang and Deng (2021) also works on rule induction, but their induced rule is in quasi-natural language but not natural language. The reasoner they adopted is symbolic, while we use neural methods as PLM as inductive reasoner.

### Inductive Logic Programming

Inductive Logic Programming (ILP) is a subfield of machine learning that uses first-order logic to represent hypotheses and data. It relies on logic language for knowledge representation and reasoning purposes (De Raedt 2010). We propose a new paradigm that can naturally avoid three systematic disadvantages of ILP (Cropper et al. 2022).

Rule Template (First Order Logic)	Rule Template (Natural Language)
$\forall x, condition(x) \implies conclusion$	If ..., then ...
$\exists x, condition(x) \implies conclusion$	There exists ..., which ...
$\forall x, condition(x) [\wedge condition(x)]^+ \implies conclusion$	If ... and ..., then ...
$\forall x, condition(x) [\vee condition(x)]^+ \implies conclusion$	If ... or ..., then ...

Table 2: The mapping relation between basic first-order logic rule template and natural language rule template.

## Relation with Other Reasoning Tasks

The goal is quite different from deductive reasoning as given facts and rules and reach to new facts (Clark, Tafjord, and Richardson 2020; Liu et al. 2020; Talmor et al. 2020; Porada, Sordoni, and Cheung 2021). Rather, we want to induce rules from facts, where rules are more general statements than given facts. Our goal is also different from past works on abductive reasoning as given facts and finding the casual reasons for the facts (Bhagavatula et al. 2020). Rather, we want to induce rules that generalize over fact itself and possibly can fit other circumstances.

## Dataset Collection and Our Proposed Evaluation Metrics

In this section, we discuss the data collection process for our proposed dataset and our proposed metrics for automatic and human evaluation of the models developed for the task.

In general, we propose two datasets. The first one, termed DEER (inDuctive rEasoning with natural language REpresentation), contains 1.2k rule-fact pairs, where rules are written by human annotators in English, and facts are existing English sentences on the web. The other one, termed with DEERLET (classification of inDucEd rULEs with natuRAl Language representATion), including (fact, rule, label0, label1, label2, label3) tuples, where facts are the same as in DEER, rules are generated output from PLMs, and label0/1/2/3 are classification labels describing different aspects of induced rules. Specifically, rules in DEERLET are collected from GPT-J (Wang and Komatsuzaki 2021) using the in-context few-shot setting. We choose this setting because GPT-J is powerful enough (6 billion parameters) so that a proportion of the generated rules are reasonable, but not very accurate so that these generations and their annotations can benefit the models finetuned on it.

DEER is used as the main dataset for the task, and DEERLET is used to measure the classification performance of specific capabilities that are required by inductive reasoning according to philosophy literature (Norton 2005).

### Dataset Collection of DEER

We collect 1.2k natural language rule-fact pairs where rules cover 6 topics and 4 common rule types of First-Order Logic (Russell 2010). The 6 topics are zoology, botany, geology, astronomy, history, and physics. The 4 First-Order

Logic rule types are implications with universal quantifier ( $\forall x, condition(x) \implies conclusion$ ), implications with existential quantifier ( $\exists x, condition(x) \implies conclusion$ ), conjunctive implications with universal quantifier ( $\forall x, condition(x) [\wedge condition(x)]^+ \implies conclusion$ ), disjunctive implications with universal quantifier ( $\forall x, condition(x) [\vee condition(x)]^+ \implies conclusion$ ). As we hope to collect rules written in natural language, we translate logic rules to natural language using templates as shown in Table 2.

Natural language rule is firstly written by human experts, then for each rule 6 supporting facts, which consist of 3 long facts and 3 short facts are collected from existing human-written text from commercial search engines and Wikipedia. Long facts are paragraphs collected from different web pages to ensure their difference, and short facts are core sentences selected from corresponding long facts. Each fact itself should contain enough information that is possible to induce the full corresponding rule (an example of short facts for a rule is shown in Table 1).

Sixty percent of the rules in DEER are more general than any of their facts alone at least in one dimension. We describe this process as “inducing general rules from specific facts”. However, we find that there are many general statements (also referred to as general fact) of a rule on the web. Therefore, for rule induction systems to be able to utilize both “specific facts” and “general facts”, forty percent of the rules in DEER are equipped with general facts. We describe this process as “inducing general rules from general facts”.

To validate the correctness of the DEER dataset, we randomly split DEER data to 4 subsets, and 4 graduate students manually check each of the subsets on whether each fact contains enough information that is possible to induce the given rule and whether the specific/general labels are correct. The overall correctness of the sampled DEER data is 95.5%.

The reason that DEER is not larger is that it requires experts who are familiar enough with inductive reasoning and possesses a relatively high level of science knowledge to annotate.

### Dataset Collection of DEERLET

DEERLET is a dataset collected by a human expert in inductive reasoning for classification tasks to evaluate the specific capabilities required by inductive reasoning. It contains 846 tuples with format (fact, rule, label0, label1, label2, label3). Among the tuples, 546 are used for training, 100 for validation, and 200 for testing. Here, facts are directly from DEER, but the corresponding rules are collected from PLMs, and label0 to label3 are classification labels evaluating specific aspects of the generated rules. The reason in DEERLET we collect rules from the generation of PLMs is that we want to avoid human annotation biases (Amidei, Piwek, and Willis 2020).

We develop label 0/1/2 based on the requirements of induced rules in philosophy literature (Norton 2005), and develop label 3 based on a NLP aspect of the problem. In particular, label0 measures whether a rule is not in conflict with its fact; Label1 measures whether a rule fits commonsense;

	Generated rules with top 0%~top10% BLEU	Generated rules with top 10%~top20% BLEU	...	Generated rules with top 90%~top100% BLEU
Weight	$weight_0(45)$	$weight_1(35)$	...	$weight_9(-45)$
Recall	$recall_0$	$recall_1$	...	$recall_9$

Table 3: Illustration of the weights and recalls in WRecall, one of our proposed automatic evaluation metrics. Here weights reflect the importance of blocks of generated rules.

Label2 measures whether a rule is more general than its fact, as inductive reasoning is “ampliative”, and requires the induced rule to have higher coverage than facts (Norton 2005). Appendix 7 illustrates label2 with more details. Label3 measures whether a rule is not trivial (mostly incomplete sentence or the latter part is a repetition of its former part).

Inspired by Obeid and Hoque (2020), label 0/1/2 are annotated on a 3-point scale (true / partially true / false), and label 3 are annotated on a 2-point scale (true / false). More details on annotation of DEERLET are illustrated in Appendix 7.

## Adopted and Our Proposed Evaluation Metrics

**Human Evaluation Metric** DEERLET provides human annotations for evaluation of the generated rules from four different aspects. Here we use precision / recall / f1, and the four aspects in DEERLET for human evaluation,

**Automatic Evaluation Metric** For the DEER dataset, as it requires generating rules based on input facts, the first metric we adopt is METEOR (Banerjee and Lavie 2005), which has been widely used for evaluating machine-generated text quality. Appendix 7 compares METEOR and BLEU (Papineni et al. 2002), and illustrates the reasons why METEOR should be a better metric for this task.

More specifically, we calculate the averaged METEOR score of the generated rules (after filtering, if a model had a filtering phase). From the observation that even humans still constantly make mistakes on inductive reasoning, we assume any framework for this task might (but not necessarily) contain two phases as generation and filtering to obtain higher performance. However, if with a filtering phase, METEOR only considers the rules that are not filtered.

It makes the METEOR metric here a similar metric to “precision”, as it only calculates the score for rules that are classified as “true”. As a result, the model might have a low recall in that it might only keep the rule with the highest confidence score, and classify many reasonable good rules as “false”.

To measure the “recall” of inductive reasoning models, we propose “weighted recall (WRecall)” as the second automatic evaluation metric for this task. The difficulty lies in that we don’t have the ground truth labels for generated rules without human evaluation. To calculate WRecall, we make an assumption, which is that the higher METEOR a rule has, generally the higher probability it is a reasonable rule for given facts. This assumption is reasonable given

the relatively high correlation coefficient between METEOR and human evaluation shown in Appendix 7. Specifically, as shown in table 3, we can first calculate the METEOR for each generated rule, and sort them based on the value of METEOR. Then we calculate the recall value for each block of generated rules, during which we assume only the rules in that block have “true” ground truth label. We also add a linearly changing weight for each block according to their importance. To ensure WRecall is in the range [0,1], WRecall is linearly normalized:

$$WRecall = \frac{\sum_{i=0}^9 weight_i * recall_i + 125}{250} \quad (1)$$

Now that we have a METEOR metric that provides a similar measurement of “precision”, and WRecall for “recall”, we propose GREEN (**GeometRic mEan of METEOR aNd WRecall**) to consider METEOR and WRecall together. It is defined as a geometric mean instead of a harmonic mean because METEOR is not in the range [0, 1]. More specifically, GREEN is calculated as:

$$GREEN = \sqrt{METEOR * WRecall} \quad (2)$$

In general, compared with METEOR, GREEN gives a more comprehensive evaluation of the induced rules. Therefore GREEN can be a more favorable metric when the recall is an important factor (e.g., when computational power is limited). However, when the precision of the induced rules is more favored, METEOR should be a more proper metric than GREEN. Appendix 7 discusses more on the importance of each metric for this task.

## Methodology

In this section, we formally present the task definition and our proposed framework for natural language inductive reasoning. Figure 1 illustrates the general architecture of our proposed approach.

### Task Definition

DEER dataset is used as the dataset for the natural language inductive reasoning task. The data format for DEER is (*rule*, *fact*), where both *rule* and *fact* are natural language sentences. The goal of the task is to generate reasonable natural language rules given *fact*, where the rules should be more general and therefore cover more information than *fact*.

### Our Framework

Hypothetical Induction is an important induction type in inductive reasoning (Norton 2005). It can be understood as when people make observations, they might propose a hypothesis as a general rule that can entail the observations. For example, when people observe that the Sun rises and falls every day, they might induce a hypothesis that the Earth is rotating itself, which is more general than the observations as the hypothesis can also help to explain the observable movements of the other Milky Way stars relative to the Earth.

Hypothetical induction fits our task well, as in DEER we also want to induce a hypothesis as a more general rule that

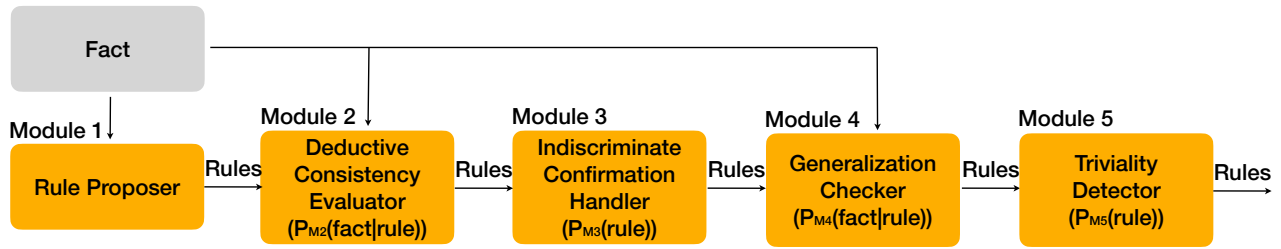


Figure 1: Our proposed framework (CoLM) for inductive reasoning with natural language representation task. Rule Proposer is a generative model based on input facts and desired rule template, aiming at generating (a large number of) rule candidates. Deductive consistency evaluator, indiscriminate confirmation handler, generalization checker, and triviality detector are classification models that filter improper rules according to four requirements of the induced rules in inductive reasoning.

can entail the facts. We borrow insights from the requirements for the induced rules in hypothetical induction to develop our framework. Specifically, there are mainly three requirements (Salmon 1989; Norton 2005). The first is that a correct hypothesis should be able to entail deductively as many observations as possible. The second is that the hypothesis should follow the laws of nature, as one could always concoct some imaginary hypothesis that is able to explain the observations but violates reality (e.g., the Earth is the center of the Universe so that the Sun orbits around the Earth). In inductive reasoning, the failure to recognize a rule that runs counter to reality is called “indiscriminate confirmation”. The third is a basic requirement for inductive reasoning, where the hypothesis should be a more general statement than the observations (Appendex 7 illustrates the meaning of “general”). We additionally introduce a fourth requirement from NLP aspects since this task uses natural language as knowledge representation. It is that a rule should not be trivial (e.g. incomplete sentence or the latter sub-sentence simply repeats its former sub-sentence).

More concretely, we define the requirements for designing our framework as 1) there should be as fewer contradictions between facts and the rule as possible, and 2) the rule should comply with commonsense, 3) the content in facts should be specific statements that are covered by the rule, 4) the rule should not be trivial.

Based on this, we develop our framework as shown in Figure 1. It consists of five modules, where module 1 (M1) is the rule proposer, module 2 (M2) is the deductive consistency evaluator, module 3 (M3) is the indiscriminate confirmation handler, module 4 (M4) is the generalization checker, and module 5 (M5) is the triviality detector. Specifically, M1 is in charge of the generation of rules. M2, M3, M4, M5 are independent classification models each verifying rules with different requirement. The role of M2/3/4/5 is similar to the verifier developed for deductive reasoning to make more solid reasoning steps (Yang, Deng, and Chen 2022). The independence of M2/3/4/5 makes it possible to run them in parallel.

In practice, we implement all five modules with pre-trained language models. We call our implementation as CoLM (Chain-of-Language-Models). The goal of M1 is to generate rules based on the input facts and a given rule template. Thus, M1’s input contains facts, a rule template, and

prompts that demonstrate the rule induction task. M2 and M4’s inputs include prompts that explain the rule-fact compatibility, a rule, and a fact; M3 and M5’s input includes again prompts that explain the task and a rule, as their targets are independent of fact.

More interestingly, although our framework is solely based on the insights from philosophy literature, we also find a mathematical interpretation of this approach. Here, we denote  $P(A)$  as the probability indicating whether  $A$  is valid for simplicity. Thus, M2 and M4 jointly measure the validity of a fact given the corresponding rule  $P(fact|rule) \approx P_{M24}(fact|rule) = P_{M2}(fact|rule)P_{M4}(fact|rule)$ , M3 and M5 directly measure the validity of the rule itself  $P(rule) \approx P_{M35}(rule) = P_{M3}(rule)P_{M5}(rule)$ . By using Bayes’ rule, we can easily show that the validity of a rule based on the input fact is

$$P(rule|fact) \approx P_{M24}(fact|rule)P_{M35}(rule). \quad (3)$$

Note that this score is merely a discrimination score and thus different from the generation probability from M1. In other words, the rules proposed by M1 are then selected by M2/3/4/5 in a Bayesian inference fashion.

## Experiments

In this section, we discuss the evaluation metrics and baselines and then present the main results of our framework.

### Evaluation Metrics

We carry out evaluations for the overall framework (the rule generation task with DEER) and individual modules for classification using DEERLET.

For evaluation of the rule generation of the overall framework, we use METEOR, WRecall, and GREEN as automatic evaluation metrics; And use precision, recall, f1, and the four metrics in DEERLET as human evaluation metrics. WRecall, GREEN, and the four metrics in DEERLET are our newly proposed metrics for inductive reasoning introduced in Section 3.

For evaluation of the classification tasks on DEERLET, we use accuracy, f1, and averaged precision as metrics.

### Baselines

We use a non-neural method and a neural method as baselines for the framework. We call the non-neural baseline

Models	METEOR	WRecall	GREEN	precision (%)	recall (%)	f1	consistent	commonsense	general	non-trivial
R+F	11.20	0.50	2.37	9.0	<b>100.0</b>	0.17	0.90	0.15	0.28	0.85
M1	25.49	0.50	3.57	45.0	<b>100.0</b>	0.62	0.63	0.60	0.83	0.86
M1 + M2	25.77 / 27.71	0.52 / <b>0.59</b>	3.64 / 4.04	45.9 / 59.8	87.8 / 71.1	0.60 / 0.65	0.63 / 0.75	0.62 / 0.72	0.83 / 0.92	0.86 / 0.94
M1 + M3	25.57 / 27.44	0.50 / <b>0.59</b>	3.59 / 4.03	45.2 / 60.2	84.4 / 75.6	0.59 / <b>0.67</b>	0.63 / 0.77	0.60 / 0.74	0.83 / 0.89	0.87 / 0.91
M1 + M4	25.84 / 26.90	0.51 / <b>0.59</b>	3.62 / 3.99	<b>48.5</b> / 53.3	92.2 / 88.9	<b>0.64</b> / <b>0.67</b>	0.64 / 0.67	<b>0.64</b> / 0.65	<b>0.84</b> / 0.91	0.88 / 0.89
M1 + M5	25.54 / 25.97	0.50 / 0.53	3.58 / 3.72	46.1 / 48.1	97.8 / 97.8	0.63 / 0.65	0.64 / 0.66	0.61 / 0.63	0.83 / 0.83	0.88 / 0.91
CoLM	<b>26.30</b> / <b>29.07</b>	<b>0.53</b> / 0.57	<b>3.74</b> / <b>4.08</b>	48.1 / <b>70.0</b>	72.2 / 54.4	0.58 / 0.61	<b>0.65</b> / <b>0.81</b>	<b>0.64</b> / <b>0.80</b>	<b>0.84</b> / <b>0.94</b>	<b>0.90</b> / <b>0.97</b>

Table 4: Result of our proposed framework and baselines on DEER under in-context few-shot / finetuning setting. The first three metrics are automatic metrics, and the last seven metrics are human evaluation metrics.

“R+F”, as it randomly fills the given rule template with sentences or phrases from the given fact. The neural baseline we use is the rule proposer itself in Figure 1.

We use majority class and TF-IDF (Jones 1972) as baselines for individual modules. The majority class baseline always predicts “yes”, which is equivalent to not using M2/3/4/5 to filter rules from M1. TF-IDF is another reasonable baseline as the induced rules contain similar contents compared to input facts. In practice, each input fact-rule pair is assigned a TF-IDF value, and a threshold for correctness (to compare with the TF-IDF value) is tuned on the DEERLET validation set.

## Main Results

All modules are implemented with GPT-J (Wang and Komatsuzaki 2021), a pre-trained language model with 6 billion parameters. For better analysis, we conduct the experiments in three settings, including zero-shot setting, in-context few-shot setting (Liu et al. 2021; Brown et al. 2020a) and finetuning setting. The only exception is that we do not test finetuning setting on M1 (the only generative module), since we are mainly investigating (out of box) pretrained large language model’s ability. However if with finetuning, language model might perform worse on out-of-distribution data and lose their generality for input facts from different topics (Kumar et al. 2022). For this reason we do not implement with T5 (Raffel et al. 2020) but with GPT-J.

To save space, we only report the results of in-context few-shot setting and finetuning setting in Table 4 and Table 5, leaving the zero-shot results in the appendix. The thresholds of M2/3/4/5 used in Table 4 and Table 5 are tuned on the DEERLET validation set. More details on setting up thresholds are illustrated in Appendix 7.

The results on DEER are shown in Table 4. As expected, the M1 alone outperforms the R+F baseline across the board, indicating that the PLM has some rule induction capability. Augmenting the M1 with some filtering mechanism can reliably improve the generated rule quality further. Lastly, our full model, CoLM, outperforms all baselines justifying the effectiveness of our proposed framework for natural language inductive reasoning.

The results on DEERLET are summarized in Table 5. In this experiment, we investigate the classification performance of language models in terms of different aspects required by inductive reasoning, which includes deductive consistency, indiscriminate confirmation, and generalization

Metrics	Accuracy (%)	F1	Averaged Precision
<b>Deductive Consistency Evaluator (M2)</b>			
Majority class	62.5	0.769	0.63
TF-IDF	62.5	0.769	0.69
GPT-J	61.5 / <b>74.0</b>	0.71 / <b>0.83</b>	0.75 / <b>0.83</b>
<b>Indiscriminate Confirmation Handler (M3)</b>			
Majority class	60.0	0.750	0.60
TF-IDF	60.0	0.750	0.64
GPT-J	56.0 / <b>70.5</b>	0.57 / <b>0.77</b>	0.66 / <b>0.79</b>
<b>Generalization Checker (M4)</b>			
Majority class	83.0	0.91	0.83
TF-IDF	83.0	0.91	0.86
GPT-J	71.0 / <b>86.0</b>	0.82 / <b>0.92</b>	0.87 / <b>0.97</b>
<b>Triviality Detector (M5)</b>			
Majority class	86.0	0.93	0.86
TF-IDF	86.0	0.93	0.90
GPT-J	78.5 / <b>89.5</b>	0.87 / <b>0.94</b>	0.89 / <b>0.94</b>

Table 5: Results on DEERLET for different modules under in-context few-shot / finetuning settings.

/ triviality classification. It shows that TF-IDF achieves the same performance with majority class baseline in accuracy and f1 metrics. The reason is that the best thresholds obtained for TF-IDF are all zero, which means that TF-IDF value is not effective for the four tasks. It also shows that the few-shot GPTJ performs worse than the majority class baseline, while finetuned GPTJ steadily performs better.

## Analysis

In this section, we investigate the question of “how well can pretrained language models perform inductive reasoning?”. Specifically, we provide analyses in terms of rule types, topics, variations of input fact, and scales of language models. Except for Table 9, the input used is short fact, 3 fact, full fact. Except for Table 2, the model used is GPT-J. All experiments in this section are based on the in-context few-shot setting, each averaged by 5 runs. Similar trends are also observed in other settings. We report METEOR and GREEN as metrics in this section.

### Different Rule Types

Table 6 shows the breakdown evaluation of CoLM based on four basic rule types in logic language (Russell 2010). The

Models	If ..., then ...	There exists ..., which ...	If ... and ..., then ...	If ... or ..., then ...
R+F	9.87 / 2.22	17.45 / 2.95	10.63 / 2.30	12.53 / 2.50
M1	22.65 / 3.37	31.92 / 4.00	26.25 / 3.62	28.75 / 3.79
M1+M2	22.90 / 3.44	33.04 / <b>4.38</b>	26.44 / 3.66	28.61 / 3.72
M1+M3	23.01 / 3.48	32.16 / 3.99	25.69 / 3.44	29.03 / 3.87
M1+M4	22.43 / 3.26	32.44 / 4.18	<b>27.15 / 3.75</b>	<b>29.21 / 3.94</b>
M1+M5	22.70 / 3.38	32.47 / 4.14	26.27 / 3.63	28.72 / 3.79
CoLM	<b>23.23 / 3.51</b>	<b>33.46 / 4.38</b>	27.06 / 3.73	29.20 / 3.92

Table 6: Analysis of PLM (GPT-J)’s performance (measured in METEOR / GREEN) in with different rule templates.

Models	Zoology	Botany	Astronomy	Geology	History	Physics
R+F	9.65 / 2.20	10.24 / 2.26	13.09 / 2.56	13.28 / 2.58	11.07 / 2.35	11.44 / 2.39
M1	29.29 / 3.83	30.47 / 3.90	34.01 / 4.12	28.28 / 3.83	23.61 / 3.44	18.69 / 3.06
M1+M2	<b>30.01 / 4.04</b>	30.34 / 3.84	<b>34.34 / 4.21</b>	28.40 / 3.79	23.79 / 3.49	19.04 / 3.18
M1+M3	29.06 / 3.70	30.40 / 3.88	33.37 / 3.90	28.55 / 3.84	23.83 / 3.49	19.00 / 3.19
M1+M4	29.95 / 3.94	<b>31.02 / 4.03</b>	34.26 / 4.19	28.81 / <b>3.96</b>	24.47 / 3.63	18.76 / 3.10
M1+M5	29.34 / 3.84	30.47 / 3.91	34.12 / 4.15	28.40 / 3.79	23.53 / 3.39	18.77 / 3.07
CoLM	29.92 / 3.88	30.93 / 4.00	34.06 / 4.11	<b>28.95 / 3.94</b>	<b>24.94 / 3.71</b>	<b>19.54 / 3.35</b>

Table 7: Analysis of PLM (GPT-J)’s performance (measured in METEOR / GREEN) in under different topics.

mapping between the logic forms and corresponding natural language templates can be found in Table 2.

The table shows that “there exists ..., which ...” achieves the best performance. It is reasonable, as simply copying the contents of facts to compose a rule will be acceptable for  $\exists$  quantifier in logic.

## Different Topics

Table 7 shows the performance of CoLM over different topics. CoLM performs much worse on History and Physics than on the other topics. We attribute the reasons to that the rules in history and physics have high variance, demand a higher level of abstraction, and are not very similar to the input facts. For example, in physics, many rules are natural language descriptions of physical laws such as Newton’s law of universal gravitation, while the input facts might be the values of gravitational force and mass of specific objects. In contrast, CoLM achieves better performance in Botany. One possible reason is that many rules in botany can be very similar to the input facts (an example is shown in Table 1).

Models	Specific facts	General facts
R+F	10.15 / 2.25	12.79 / 2.53
M1	26.37 / 3.63	24.18 / 3.48
M1+M2	26.76 / 3.75	24.42 / 3.53
M1+M3	26.54 / 3.68	24.15 / 3.45
M1+M4	26.74 / 3.70	24.64 / 3.57
M1+M5	26.39 / 3.63	24.28 / 3.51
CoLM	<b>27.39 / 3.86</b>	<b>24.89 / 3.63</b>

Table 8: Analysis of PLM (GPT-J)’s performance (measured in METEOR / GREEN) in with specific or general input facts.

Models	Long facts 1 full facts	Short facts 1 full facts	Short facts 2 full facts	Short facts 3 full facts	Short facts 3 missing facts
R+F	9.35 / 2.16	10.87 / 2.33	11.16 / 2.36	11.20 / 2.37	11.52 / 2.40
M1	23.79 / 3.45	25.13 / 3.54	25.65 / 3.58	25.49 / 3.57	25.11 / 3.54
M1+M2	24.00 / <b>3.50</b>	25.36 / 3.63	25.89 / 3.64	25.77 / 3.64	25.30 / 3.59
M1+M3	23.94 / 3.49	25.39 / 3.61	25.87 / 3.63	25.57 / 3.59	25.33 / 3.62
M1+M4	23.92 / 3.44	25.27 / 3.55	25.93 / 3.62	25.84 / 3.62	25.35 / 3.55
M1+M5	23.80 / 3.46	25.30 / 3.61	25.74 / 3.61	25.54 / 3.58	25.15 / 3.56
CoLM	<b>24.15 / 3.50</b>	<b>25.79 / 3.68</b>	<b>26.48 / 3.76</b>	<b>26.30 / 3.74</b>	<b>25.73 / 3.66</b>

Table 9: Analysis of PLM (GPT-J)’s performance (measured in METEOR / GREEN) with different input lengths and whether fact contains enough information.

## Variations of Input Facts

Table 8 shows the result from specific vs general facts. In section 3 we have discussed that a rule induction system would be more widely applicable if it can utilize both specific fact and general fact. In table 8, general facts cases result in lower performance. We think one of the most possible reasons is that in DEER many general facts do not directly contain the content of the corresponding gold rules. For example, general facts can be mottos from philosophers such as Socrates, and rules can be an understandable description of such mottos in natural language rule format.

In table 9, long facts mean the paragraph-level facts in DEER, and short facts mean the core sentence-level facts selected from corresponding paragraph-level facts. The different number of facts indicates the different number of facts given as input that exhibit similar rule patterns (e.g. Lemon tree / orange tree / apple tree can conduct photosynthesis). We consider the number of facts as an important factor because psychological research shows that more facts with similar patterns can help with inductive reasoning (Heit 2000). Missing fact experiments are also conducted, where for each fact we randomly throw the former half or the latter half of the sentences. It is an important setting as it is hard for the input facts to cover all the elements of the desired rule in a realistic scenario. As a result, it might be common that some required pieces of fact are missing. The results indicate that larger number of concise but full facts are beneficial for rule induction, while too many facts with similar patterns might not be helpful.

## Different Scales of PLMs

Figure 2 shows the influence of the scale of pre-trained language models (under in-context few-shot setting) on induction. Here, we consider GPT-Neo 125M, GPT-Neo 1.3B, GPT-Neo 2.7B, GPT-J 6B and GPT-NeoX 20B (Wang and Komatsuzaki 2021). The figure shows that generally performance of M1 steadily improves as the scale being larger, and M2/3/4/5 are only helpful since 6B parameters. The only exception is that both M1 and M2/3/4/5 might reach a plateau in 20B parameters. We did not use GPT-3 (Brown et al. 2020b) to analyze scale since M2/3/4/5 need embeddings for prediction, but the API of GPT-3 does not support return full embeddings.

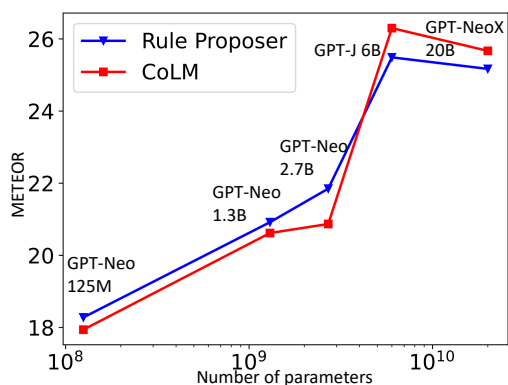


Figure 2: Influence of the scale of PLM on inductive reasoning task with DEER (measured with METEOR).

## Future Work and Challenges

The new paradigm of using natural language as the representation of knowledge and using PLMs as the inductive reasoner for inductive reasoning opens the possibility of automatically inducing rules on the countless web corpus. On the other hand, there are still remaining challenges in this direction as not all facts can be used to induce rules. Many fact pieces in DEER for a single rule are collected from different places on the web, so that the input contains enough and proper information to induce rules. However, when using the web corpus, it is hard to ensure that input facts contain such information. As a result, it is challenging to reliably obtain high-quality facts that can be utilized to induce rules.

## Conclusion

To overcome the systematic problems of using logic language for inductive reasoning, we propose a new paradigm (task) of inducing natural language rules from natural language facts, and correspondingly propose a dataset DEER and new evaluation metrics for this task. We provide the first and comprehensive analysis of pretrained language models' ability to induce natural language rules from natural language facts. We also propose a new framework drawing insights from philosophy literature, which show in the experiment section that surpasses baselines in both automatic and human evaluations.

## References

Amidei, J.; Piwek, P.; and Willis, A. 2020. Identifying Annotator Bias: A new IRT-based method for bias identification. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4787–4797.

Aristotle, A. 1994. *Posterior analytics*, volume 1. Clarendon Press Oxford, UK.

Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Goldstein, J.; Lavie, A.; Lin, C.; and Voss, C. R., eds., *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005*, Ann Arbor, Michigan,

USA, June 29, 2005, 65–72. Association for Computational Linguistics.

Bhagavatula, C.; Bras, R. L.; Malaviya, C.; Sakaguchi, K.; Holtzman, A.; Rashkin, H.; Downey, D.; Yih, W.; and Choi, Y. 2020. Abductive Commonsense Reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

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.; and Amodei, D. 2020b. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and 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*.

Clark, P.; Tafjord, O.; and Richardson, K. 2020. Transformers as Soft Reasoners over Language. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 3882–3890. ijcai.org.

Cropper, A.; Dumančić, S.; Evans, R.; and Muggleton, S. H. 2022. Inductive logic programming at 30. *Machine Learning*, 111(1): 147–172.

Davison, J.; Feldman, J.; and Rush, A. 2019. Commonsense Knowledge Mining from Pretrained Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1173–1178. Hong Kong, China: Association for Computational Linguistics.

De Raedt, L. 2010. Inductive Logic Programming.

Flach, P. A.; and Kakas, A. C. 2000. Abductive and inductive reasoning: background and issues. In *Abduction and induction*, 1–27. Springer.

Heit, E. 2000. Properties of inductive reasoning. *Psychonomic Bulletin & Review*, 7(4): 569–592.

Jones, K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kumar, A.; Raghunathan, A.; Jones, R. M.; Ma, T.; and Liang, P. 2022. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning. In Bessiere,



- C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 3622–3628. ijcai.org.
- Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR*, abs/2107.13586.
- Meng, Y.; Zhang, Y.; Huang, J.; Wang, X.; Zhang, Y.; Ji, H.; and Han, J. 2021. Distantly-Supervised Named Entity Recognition with Noise-Robust Learning and Language Model Augmented Self-Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10367–10378.
- Mercier, H. 2018. The enigma of reason. In *The enigma of reason*. Harvard University Press.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Misra, K.; Rayz, J. T.; and Ettinger, A. 2022. A Property Induction Framework for Neural Language Models. *arXiv preprint arXiv:2205.06910*.
- Muggleton, S.; and De Raedt, L. 1994. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19: 629–679.
- Norton, J. D. 2005. A little survey of induction.
- Obeid, J.; and Hoque, E. 2020. Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model. In *Proceedings of the 13th International Conference on Natural Language Generation*, 138–147. Dublin, Ireland: Association for Computational Linguistics.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.
- Porada, I.; Sordani, A.; and Cheung, J. C. K. 2021. Does Pre-training Induce Systematic Inference? How Masked Language Models Acquire Commonsense Knowledge.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and 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.
- Russell, S. J. 2010. *Artificial intelligence a modern approach*. Pearson Education, Inc.
- Salmon, M. H. 1989. Introduction to Logic and Critical Thinking.
- Sinha, K.; Sodhani, S.; Dong, J.; Pineau, J.; and Hamilton, W. L. 2019. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4506–4515. Hong Kong, China: Association for Computational Linguistics.
- Smullyan, R. M. 1995. *First-order logic*. Courier Corporation.
- Talmor, A.; Tafjord, O.; Clark, P.; Goldberg, Y.; and Berant, J. 2020. Leap-Of-Thought: Teaching Pre-Trained Models to Systematically Reason Over Implicit Knowledge. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Neurips*.
- Teru, K.; Denis, E.; and Hamilton, W. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, 9448–9457. PMLR.
- Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.
- Yang, K.; and Deng, J. 2021. Learning Symbolic Rules for Reasoning in Quasi-Natural Language. *CoRR*, abs/2111.12038.
- Yang, K.; Deng, J.; and Chen, D. 2022. Generating Natural Language Proofs with Verifier-Guided Search. *CoRR*, abs/2205.12443.

## Appendix

### Annotation Details for DEERLET

In DEERLET, given fact(s) and a rule, the annotation targets are whether the rule satisfies four requirements.

Specifically, the requirements are “if the rule is deductively consistent with the fact”, “if the rule fits commonsense”, “if the rule is more general than the fact”, and “if the rule is not trivial”.

The first three requirements are annotated on a 3-point scale (true / partially true / false), and the last is annotated on a 2-point scale (true / false).

Here we explain the standards of annotation on the four requirements.

For “if the rule is deductively consistent with the fact”, a 2-point will be assigned if the rule is totally relevant and consistent with the facts; a 1-point will be assigned if the rule introduces new information that does not show in facts but is consistent with the given fact as well as some limited amount of commonsense knowledge related to the facts; a 0-point will be assigned if the rule is (1) in conflict with given facts or (2) totally irrelevant to given facts or (3) introduces new information that is obviously wrong.

For “if the rule fits commonsense”, a 2-point will be assigned if the rule totally fits commonsense; a 1-point will be assigned if the rule fits commonsense at most of the time; a 0-point will be assigned if (1) the rule is totally incorrect or (2) the rule is only occasionally correct.

For “if the rule is more general than the fact”, a 2-point will be assigned if (1) the rule is more general than the facts or (2) it is obvious that the rule is trying to be more general than the facts; a 1-point will be assigned if (1) it is even hard for humans to induce a more general rule from the given facts or (2) the rule copies part of the given facts that are already containing very general information; a 0-point will be assigned if (1) from the facts it’s easy for humans to induce a more general rule but the rule is not more general or (2) the rule is totally irrelevant to the facts.

For “if the rule is not trivial”, a 0-point will be assigned if (1) the rule is an incomplete sentence or (2) the latter sub-sentence of the rule only repeats the information in the former sub-sentence of the rule; otherwise, a 1-point will be assigned.

### METEOR or GREEN?

Since inductive reasoning over natural language is a new task, and new metrics are designed (e.g., WRecall, GREEN), it is important to understand which aspects each metric focus on and which metric should we pay more attention to.

As mentioned in section 3, METEOR can be seen as evaluating the “precision” of the final rules, while GREEN evaluates “precision” and “recall” at the same time.

However, it should be aware that the “recall” here is not as important as the “recall” in other tasks. More specifically, here “recall” measures how many good rules generated by M1 are filtered by M2/3/4/5. However, we can use M1 to generate a large number of rules, and as long as CoLM has good precision, it is easy to obtain a large number of high-

quality rules, especially considering that the computational cost of only inference of M1 is relatively very low.

Based on this observation, we argue that “precision” should be a much more important aspect of evaluation compared to “recall” (measured by WRecall) or even “f1” (measured by GREEN) for this task. More specifically, “recall” can be used to mainly measure at what efficiency can the system obtain rules with high precision.

This viewpoint of evaluation metrics, of course, can raise the question of whether some typical kinds of rules are mostly filtered when pursuing rules with high precision, and in the end inductive reasoning system with high precision might only be able to obtain some other typical kinds of rules. We leave this question as an open question for this task to solve in the future.

### Why METEOR not BLEU

We choose METEOR since METEOR has a higher correlation coefficient with human evaluation than BLEU.

More specifically, on DEERLET, we calculate the METEOR and BLEU for each generated rule with its golden rule in DEER and collect the human evaluation for the generated rule from label0/1/2/3 annotations in DEERLET (we normalize each label to [0,1] and use the product of label0/1/2/3 as the overall human evaluation score for the generated rule). Then, we can calculate the correlation coefficient between METEOR / BLEU and the overall human evaluation score.

On DEERLET, the correlation coefficient between METEOR and human evaluation is 0.29, it is statistically significant as its p-value is  $4.48 * 10^{-6}$ , smaller than the significance level (0.05). Similarly, the correlation coefficient between BLEU and human evaluation is 0.24, with p-value of  $1.17 * 10^{-72}$ , which is also significant.

It shows that there is a relatively strong correlation between METEOR and human evaluation, which means that METEOR can be a sound automatic evaluation metric for inductive reasoning in the DEER dataset. We attribute the relatively high correlation to the data collection process of DEER, where facts are collected to support the rule. Therefore the facts are highly related to the ground truth rule, and it is relatively hard to induce a reasonable alternative that is significantly different from the ground truth. The reason we hypothesize empirically for the correlation coefficient not being higher is that (1) synonyms and variants of phrases exist; (2) there is only one golden rule provided instead of multiple; (3) sometimes the rules are not totally relevant to the given fact, but extend it with information that is not mentioned in facts.

Developing better metrics for measuring the similarity between sentences is a challenging topic in NLP. Of course, METEOR is not a “perfect” automatic evaluation metric for inductive reasoning. We leave the question of “what is a better metric for inductive reasoning over natural language” as an open question for future works in the field. One good thing is that WRecall and GREEN can be applied with many metrics measuring sentence similarity such as METEOR and BLEU, so the evaluation of “recall” should be able to also

benefit from the advance of metrics that evaluate “precision”.

### **Meaning of “More General” Required by Inductive Reasoning**

The “more general” required by inductive reasoning means the scope of information coverage is larger.

For instance, if facts are about cats and dogs are good accompaniment of humans, then some examples of “more general” rule can be (1) mammals are good accompaniment of humans or (2) domesticated animals are good accompaniment of humans or (3) animals with four legs are good accompaniment of human.

In these examples, the rules cover a larger scope than the facts (e.g., mammals compared to cats; domesticated animals compared to cats), and therefore the rules are “more general” than the facts.

“More general” means not only about finding higher taxonomic rank, but can be in unlimited forms. For instance, if the fact is about the Sun rises and falls every day, then some examples of “more general” rule can be (1) the Earth is the king of the universe or (2) the Earth is rotating itself.

Both rule examples are “more general” than the given fact, since the rule can entail not only the given fact, but also other not mentioned facts such as the observable movements of the other stars in the Milky Way.

### **Set up Thresholds for M2/3/4/5**

Setting up thresholds is an important step for our framework, since different thresholds can lead to different inductive reasoning results. We discuss the details of setting up thresholds in the section.

We design the standard for setting up thresholds based on heuristics that the thresholds should be set up that each module (in M2/3/4/5) should filter some rules but a single module should not filter too many rules (in this case, since we have many modules, there might not remain a reasonable proportion of rules left).

More specifically, given a rule (and facts), M2/3/4/5 can produce a score on evaluating the validity of the rule from a specific aspect. The score is the ratio of the probability of the “yes” token and “no” token obtained from the last layer of PLM. The score is in the range of [0,1].

We find that getting a specific threshold for each module is more beneficial than using the default 0.5 threshold. We obtain the thresholds on the DEERLET validation set.

More concretely, on the validation set, if there exists a global optimal threshold that (1) achieves the best f1 or accuracy and (2) the threshold should not be very close to 0 or 1 and (3) recall is not very close to 0 (when close to 1, it should not be in the case that the threshold accepts nearly all generated rules but should be that the threshold already rejects some rules), then the global optimal threshold is adopted; if there is no such global optimal threshold, then find a local optimal threshold that (1) achieves the best f1 or accuracy compared to its neighboring thresholds and (2) the threshold should not be very close to 0 or 1, and (3) the recall range is in [0.7, 0.9], then the local optimal threshold is adopted.