Moscow Institute of Physics and Technology
Phystech School of Applied Mathematics and Informatics
Department of Machine Learning and Digital Humanities

Vladlen Kulikov

# Large Language Models. Learning and Reasoning at the Inference Stage

Master's Thesis

*Advisor:* Radoslav Neychev

Moscow, 2023

# Contents

**Abstract**

In-context learning (zero-shot learning, one-shot and few-shot learning), chain-of-thoughts and prompt engineering have been more and more widely researched and discussed recently, in connection with the discovered possibility of Large Language Models (**LLMs**) based on examples (and in particular demonstrations of instructions, some actions with examples, including demonstration of reasoning) to solve certain types of problems at inference, without the need for fine-tuning, and in general without any additional training of LLMs on these tasks.

In our work, due to the lack of a general term, we refer to this emergent ability, observed in LLMs, as the **L&R effect** (Learning and Reasoning at the Inference Stage effect). This effect, along with the opportunities it creates, leaves open the question of what exactly is the reason for the emergence of such a surprising ability in LLMs.

In this work, we summarize the available evidence and derive the necessary conditions for the most striking appearance of L&R effect in LLMs – Part 2. Literature overview, and also conduct an analysis on the basis of which we express a number of hypotheses about the reasons for such behavior LLMs – Part 3. Analytical part and we will test these hypotheses in Part 4.Computational experiment.

Main questions in this work:

**I.** What conditions are required for L&R effect in LLMs?

**II.** L&R effect - What is it?

**III.** Do LLMs learn and reason at the Inference?

# 1 Introduction

Oddly enough, but in NLP(Natural Language Processing), there is no single term that characterizes the emergent ability we are studying, while the terms "learning" or "reasoning" are somewhat misleading, as we will see in further analysis.

However, the words "learning" and "reasoning" have firmly entered the terminology, so we will use the term **L&R effect** (**Learning and Reasoning at the Inference Stage effect**) and we will understand by it the ability of LLMs at the inference stage (forward pass) based on:

1) demonstration of examples, instructions describing the task, or

2) demonstrating both examples and instructions, and any actions with them (for example, logical or arithmetic operations, including chains of reasoning)

give the correct answer, without any additional training of the model.

The L&R effect is emergent in nature (emergence is a qualitative property that occurs spontaneously in a system when it reaches a certain threshold of complexity) and occurs in LLMs under the conditions described below.

Previously, demonstrations for the L&R effect included mainly character sequences, but it is worth noting that during March 2023, the appearance of such Multomodal LLMs (MLLMs) as GPT-4[2], PaLM-E [3], Kosmos-1 [4], where not only character sequences can be fed to the input of a multimodal transformer-based construction, but also mixed with images, video and audio, which are further, after preprocessing in the corresponding part of the model, "converted" into a form understandable by LLM included in MLLMs.

In this work, we systematically follow the terminology presented in [5] for the terms in-context learning, few-shot, one-shot, zero-shot, fine-tuning. We present it here (in part), to avoid different interpretations of the subject of our study and to facilitate understanding of the further presentation, for readers who are not familiar with the terminology.

Terminology illustration – Figure 1.

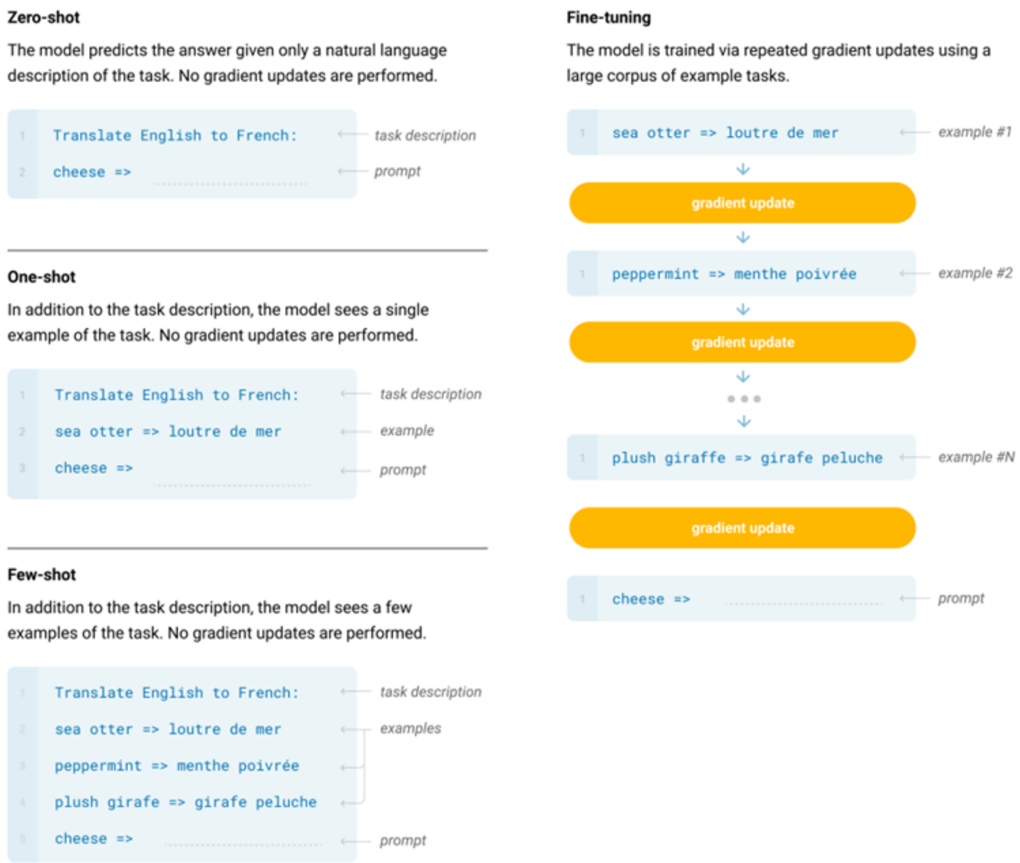



Figure 1: Zero-shot learning, One-shot learning and Few-shot learning vs. Fine-Tuning." [5, Fig.2.1]

*Fine-Tuning* – This process entails modifying the weights of a previously trained model's parameters by further training it on a task-specific dataset under supervised learning conditions.

*Zero-shot* – only a natural language instruction describing the task, without weight updates.

*One-shot* – same as a zero-shot plus one demonstration.

*Few-shot* – same as a one-shot but plus a few demonstrations.

By *in-context learning* we mean the general description given in [5], which is illustrated in Figure 2.

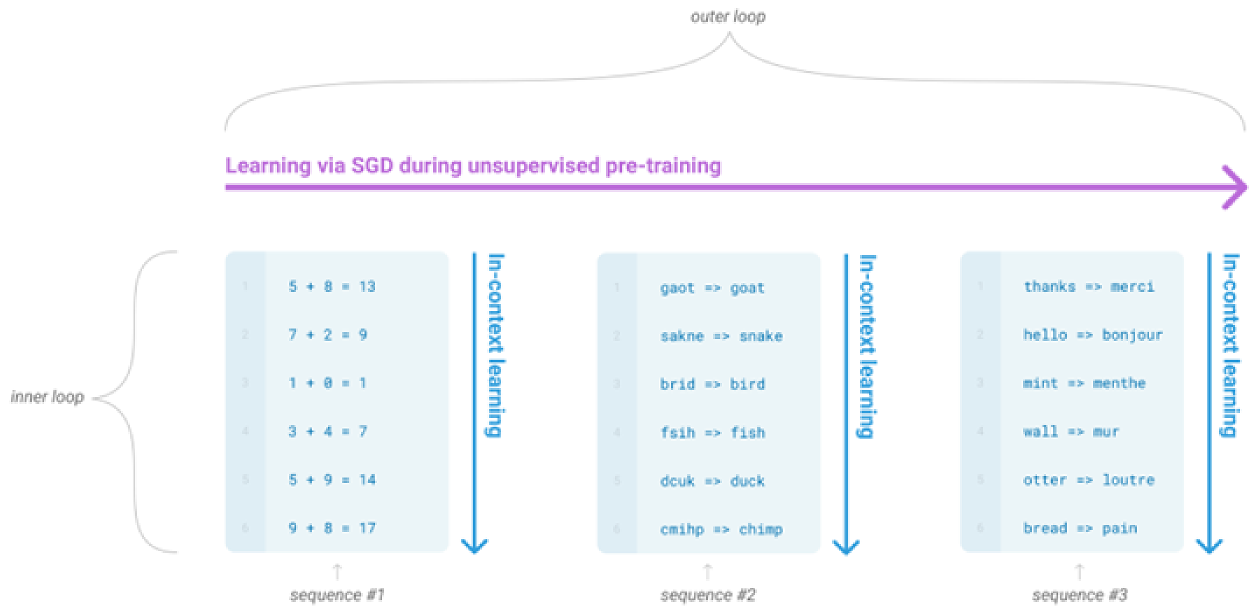



Figure 2: "Learning via SGD (during unsupervised pre-training), contrasted with In-context learning (Meta-learning at Inference stage)" [5, Fig.1.1]

But in the form of a short and simplified formulation, you can use the definition given in [6]: "In-context learning is a paradigm that allows language models to learn tasks when only a few examples in the form of demonstrations"[6].

The term *chain-of-thought(CoT)* is understood according to [7] as a series of intermediate reasoning steps — Figure 3.

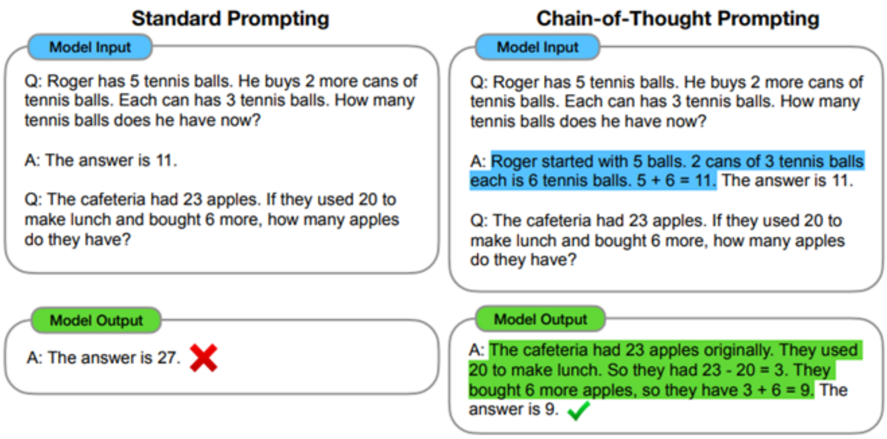



Figure 3: Chain-of-thought reasoning processes are highlighted [7, Fig.1]

And today, we are witnessing the emergence of a separate discipline – *prompt engineering* (the development and optimization of prompts to improve work outcomes), which has emerged as a generalization of the ways we work with prompts.

Furthermore, we will be interested in the conditions under which the L&R effect emerges and is most clearly observed — which in the simplest case takes the form of in-context (zero, one, few-shot) learning and whose results can be improved by chain-of-thoughts and, in general, by such a more technological approach as prompt engineering.

# 2 Literature overview

## 2.1 Language as a training set

Intuitively, the fact that the training of a language model should be done on a language dataset is obvious, since the data on which we train, the model must correspond to the problem being solved.

But what should be the properties of the original dataset in order to observe the effect of in-context learning at the inference stage? Are they related at all?

As shown in [8], in-context learning was observed under certain distribution properties of the training data themselves (and was not observed in their absence), namely when the model was trained:

1. On data following a Zipfian distribution – a type of skewed distribution frequently observed in natural data, including language – Figure 4.

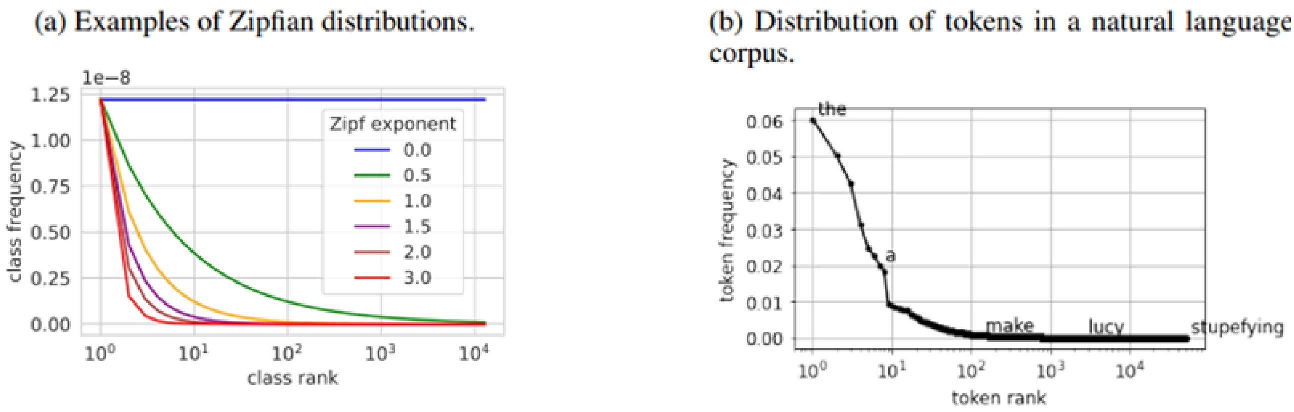



Figure 4: Figure 6a, 6b from [8]

2. In cases where data display characteristics like burstiness - where elements tend to group together rather than being evenly spread out over a given time period. In other words, a particular entity (such as a word, individual, or object) may not have a uniform distribution over time, but rather tends to occur in concentrated clusters.

3. When the data has a large number of rare classes.

4. The effectiveness of in-context learning is significantly enhanced when the meanings or interpretations of items are are dynamic and evolving, instead of being constant and unchanging.

That is, the "meaning" of elements in data, for example, words in a language, can have many possible interpretations-polysemy, and synonymy-when a single value can correspond to a set of entities.

These 4 characteristics are not only present in languages but also in other natural data, prompting contemplation about what other types of data could potentially be used for training LLMs.

**Summary for Language as a training set:** If the model was trained on "weaker" data - weaker than the language - then the effect of in-context learning is not observed.

## 2.2 Model's type

The fact of the occurrence of the in-context learning effect, depending on the architecture of the model, was studied in [8].

In this work it has been shown that the effect of in-context learning is observed only in Transformer, in contrast to recurrent architectures – RNN, LSTM (ceteris paribus – number of layers, size of hidden layer, and number of parameters, datasets, etc.) – see Figure 5.

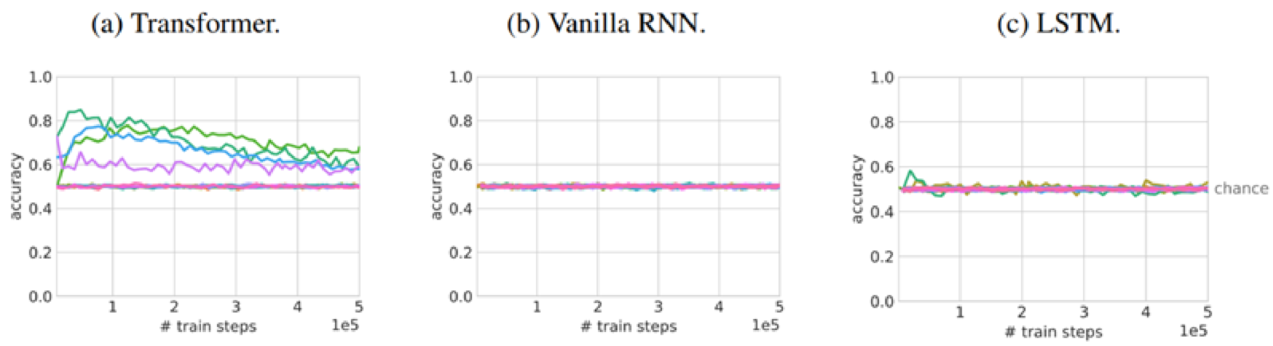



Figure 5: "In-context learning in transformers vs. recurrent architectures." [8, Fig. 7])

For comparison, the same article provides data on in-weight learning on different architectures (ceteris paribus) – Figure 6.

On both plots - For every distinct set of hyperparameters in a hyperparameter sweep, a single run was executed. Each color denotes one run, but not any particular hyperparameter values.

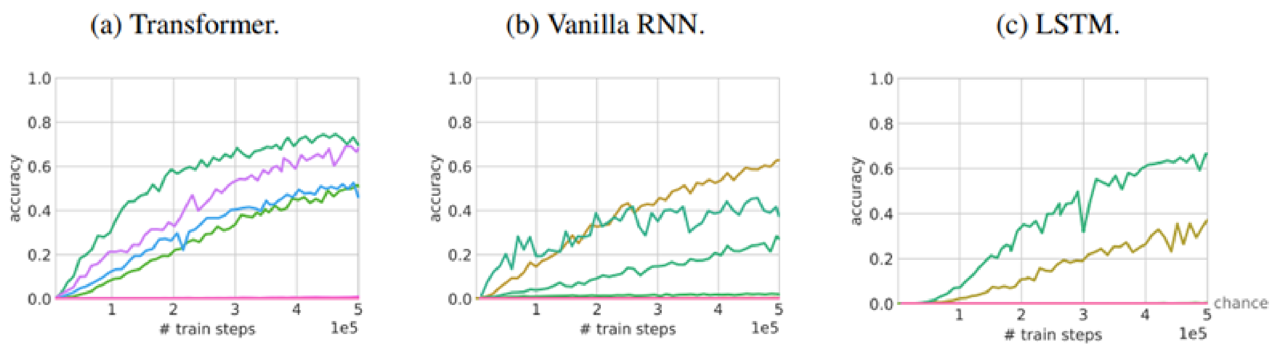



Figure 6: "In-weights learning in transformers vs. recurrent architectures."[8, Fig. 8]

A similar conclusion can be drawn by analyzing the data presented in [5], [9] and in other LLMs papers and surveys [10], [11] and in particular on Multi-modal LLMs - GPT-4[2], PaLM-E [3], Kosmos-1 [4],

All models in which in-context learning is observed are Transformers-based.

**Summary for Model's type**: Transformers is so far the only type of model in which the L&R effect (and in-context learning in particular) is reliably and significantly (not on synthetic data) observed.

## 2.3 Zero, One, Few-shot

As shown in [5] few-shot learning (see Figure 7) with the number of input examples from 2,3 with accuracy 45-50% and up to 10-15 with accuracy 60-65% demonstrates an almost linearly increasing quality, which significantly (mainly 2-3 times) exceeds the indicators for Zero-shot and One-shot with similar sizes of the considered models on tasks described in [5].

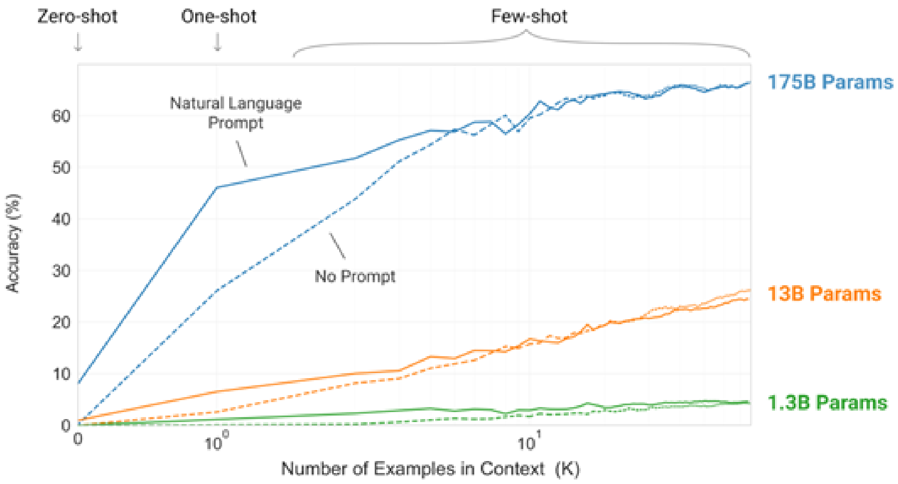



Figure 7: "Larger models make increasingly efficient use of in-context information."[5, Fig. 1.2])

Also (Figure 8), aggregate performance across 42 benchmarks demonstrates

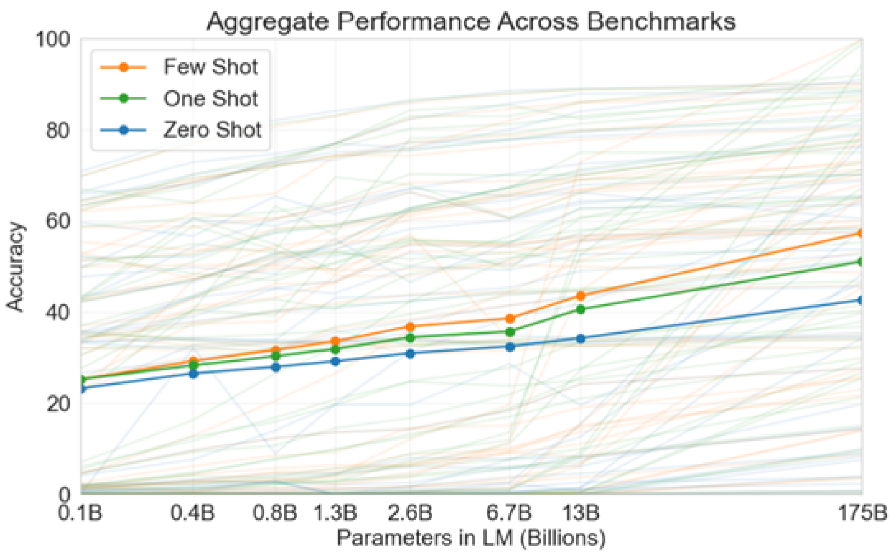



Figure 8: Aggregate performance across benchmarks.[5, Fig. 1.3])

the prevailing quality of few-shot over zero-shot learning and one-shot, with similar model sizes. The advantage of Few-shot learning is not as significant as in the previous case, but it is beyond doubt.

**Summary for Zero, One, Few-shot:** The growth in the number of examples from zero and one to few-shot is one of the key factors in improving quality L&R effect (but sometimes there are exceptions and even zero-shot shows excellent results).

## 2.4 Model's size

Like other research, in this work not considered the influence of the nuances of the architecture of each of the models. All existing LLMs are Transformers, and at this stage of the development of research in this area, it seems to us that there is no mechanism to adequately take this into account.

Therefore, in this text evaluated the models by the number of parameters, that allows to draw some general conclusions about model's global behavior at the inference stage.

In [9], the authors purposefully investigated of emergent properties of LLMs, i.e., properties not observed in smaller models. The authors demonstrate that the beneficial impact of specialized prompting can be emergent, meaning it only becomes evident when the model reaches a certain scale – Figure 9.

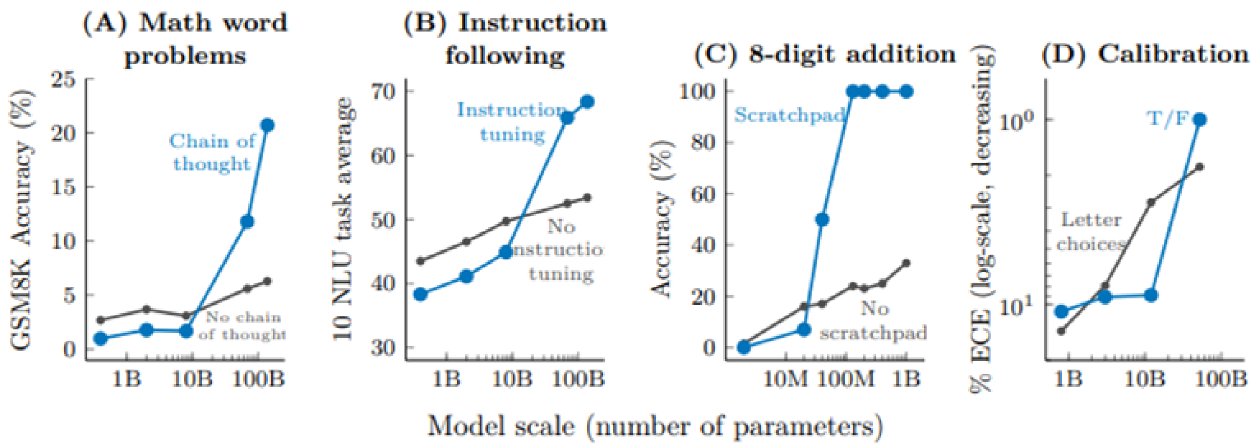



Figure 9: Specialized prompting may be emergent because it does not have a positive effect up to a certain model scale. [9, Fig. 12]

Figure 7 and 8 also show that an increase in the number of parameters (in this case, GPT-3) directly affects accuracy on in-context tasks.

Also note that to achieve accuracy $> 50\%$ (aggregate performance across 42 benchmarks), the number of parameters must exceed 13B. However, the nature of the behavior of the model between 13B and 175B is not clear, as intermediate models have not been studied.

A similar pattern of previously unobserved properties in LLMs, we see for 5 LLMs GPT-3 [5], LaMDA [12], Gopher [13], Chinchilla [14], PaLM [15] on Few-Shot prompted tasks – Figure 10.

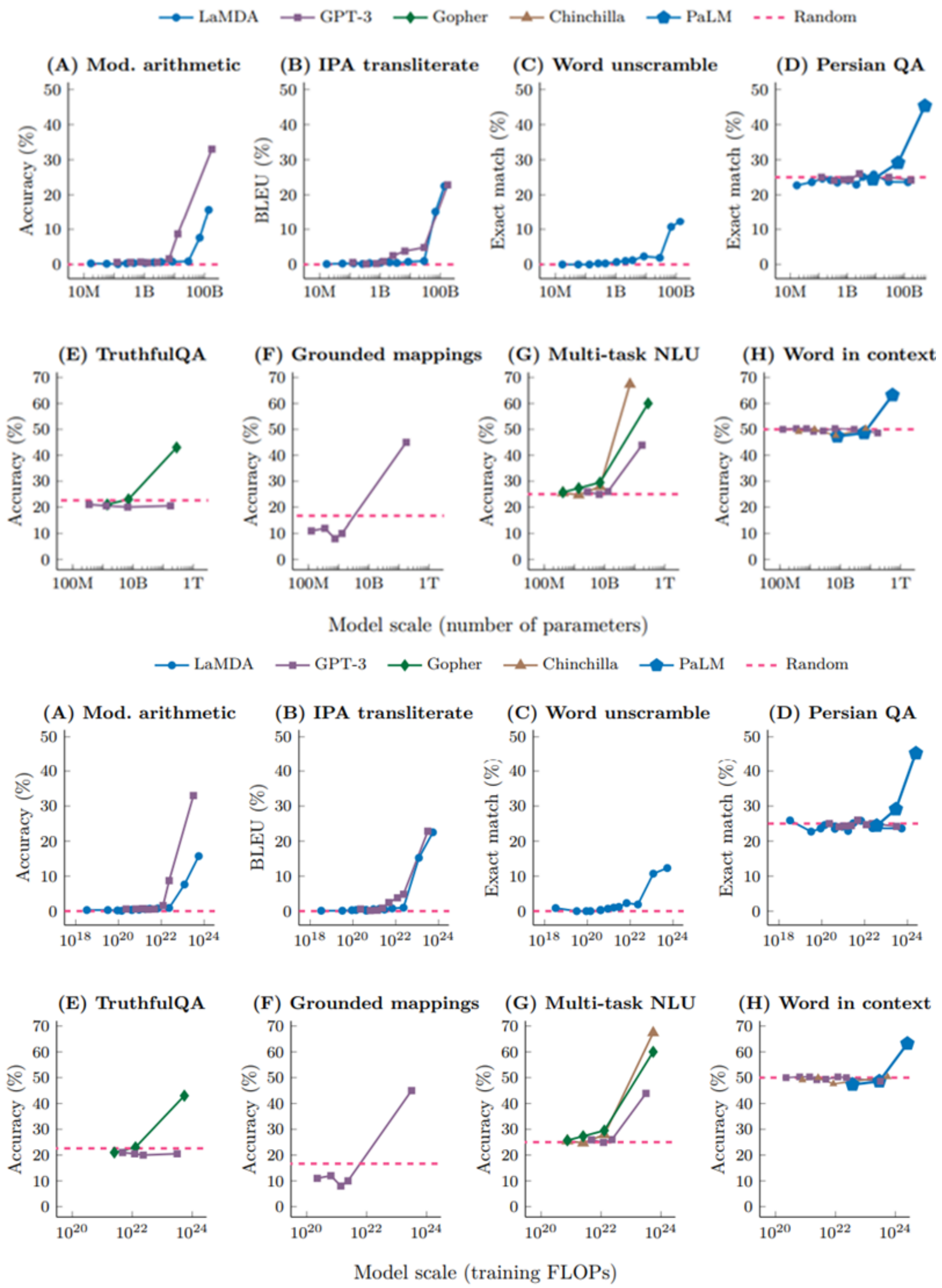



Figure 10: "Eight examples of emergence in the few-shot prompting setting." [9, Fig. 2, 11]

A more detailed picture of the required model sizes for efficient occurrence of emergent properties can be obtained by analyzing Figure 10 in conjunction with Figure 11 – Table 1.

| | Emergent scale | | |
| --- | --- | --- | --- |
| | Train. FLOPs | Params. | Model |
| Few-shot prompting abilities | | | |
| • Addition/subtraction (3 digit) | 2.3E+22 | 13B | GPT-3 |
| • Addition/subtraction (4-5 digit) | 3.1E+23 | 175B | |
| • MMLU Benchmark (57 topic avg.) | 3.1E+23 | 175B | GPT-3 |
| • Toxicity classification (CivilComments) | 1.3E+22 | 7.1B | Gopher |
| • Truthfulness (Truthful QA) | 5.0E+23 | 280B | |
| • MMLU Benchmark (26 topics) | 5.0E+23 | 280B | |
| • Grounded conceptual mappings | 3.1E+23 | 175B | GPT-3 |
| • MMLU Benchmark (30 topics) | 5.0E+23 | 70B | Chinchilla |
| • Word in Context (WiC) benchmark | 2.5E+24 | 540B | PaLM |
| • Many BIG-Bench tasks (see Appendix E) | Many | Many | Many |
| Augmented prompting abilities | | | |
| • Instruction following (finetuning) | 1.3E+23 | 68B | FLAN |
| • Scratchpad: 8-digit addition (finetuning) | 8.9E+19 | 40M | LaMDA |
| • Using open-book knowledge for fact checking | 1.3E+22 | 7.1B | Gopher |
| • Chain-of-thought: Math word problems | 1.3E+23 | 68B | LaMDA |
| • Chain-of-thought: StrategyQA | 2.9E+23 | 62B | PaLM |
| • Differentiable search index | 3.3E+22 | 11B | T5 |
| • Self-consistency decoding | 1.3E+23 | 68B | LaMDA |
| • Leveraging explanations in prompting | 5.0E+23 | 280B | Gopher |
| • Least-to-most prompting | 3.1E+23 | 175B | GPT-3 |
| • Zero-shot chain-of-thought reasoning | 3.1E+23 | 175B | GPT-3 |
| • Calibration via P(True) | 2.6E+23 | 52B | Anthropic |
| • Multilingual chain-of-thought reasoning | 2.9E+23 | 62B | PaLM |
| • Ask me anything prompting | 1.4E+22 | 6B | EleutherAI |

Figure 11: "List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge." [9, Tab.1]

L&R effect begin to appear significantly (significant for the accuracy value) in most cases in models > 50B, but as we can see from Table 1, there are exceptions.

It is noticeable that as the number of parameters of the same model increases, LLMs often become able to solve problems that they could not solve with a smaller model size, all other things being equal.

But this is not a 100% rule, progress may slow down - there is no guarantee that you will achieve the desired level with scaling.

Also, the scale of the model is not the only key factor for an emergent ability and smaller can outperform larger models, for example : "Despite having fewer model parameters and less training FLOPs, PaLM 62B demonstrates emergence on numerous BIG-Bench[16] tasks, outperforming GPT-3 175B and LaMDA 137B, which fail to do so."[9]

Also, we note that in May 2023, a paper[17] appeared - that smooth, continuous, predictable changes in the operation of a family of models may seem abrupt and unpredictable in the case that a researcher chooses a nonlinear or discontinuous metric.

However, in this case, we consider the influence of metrics on the nature of the studied dependencies to be not so fundamental, since all the models studied did not have a series of models consistently increasing at equal intervals. Against this background, the possible non-linearity of metrics is lost as a significant factor, since the graphs by the authors of all the works were built according to the actual line of models – Figure 12. For example – PALM – 8, 62, 540B.

Being postponed on the same graph, different types of model lines create the impression of sudden jumps and (or) linear dependencies, which caused the appearance of the paper [17]. However, we (like all researchers) are not studying the nature of the dependencies – whether it is linear, quadratic, or say exponential, but the fact itself – whether the effect was up to a certain size of the model, whether it originated on a model of some size and what accuracy is on the available model lines.

Recall that emergence is a previously unobserved property that occurs in a system as its complexity increases.

This emergence is observed both on all types of models in the literature review, and was observed by us during a computer experiment.

Although the authors [17] are right and the graphs may have a different growth character, in view of the nonlinearity of the metrics – but this will be important if they are evaluated on consistently increasing lines of models 1,2,3, ..., 173,174, 175, ..., 538, 539, 540B parameters.

Before the appearance of such lines of models and their studies, the remark in [17] is rather of theoretical interest.

**Summary for Model's size:** The L&R effect, i.e., the emergent properties, starts to become significant (significant for the accuracy value) in most cases in models larger than 50B-60B (on certain tasks and models from 6-13B) and with model scale from E+22 to E+24 in training FLOPs. However, models scale (i.e. the number of parameters) is not the only key factor, the influence of at least the type of task and the model itself can be traced.

## 2.5 Training dataset's size

We can estimate the order of the required data based on the data on which the LLMs (discussed in the part 2.4) were trained.

| Model | Parameters | Train tokens | Train FLOPs |
|---|---|---|---|
| GPT-3 | 125M | 300B | 2.25E+20 |
| | 350M | 300B | 6.41E+20 |
| | 760M | 300B | 1.37E+21 |
| | 1.3B | 300B | 2.38E+21 |
| | 2.7B | 300B | 4.77E+21 |
| | 6.7B | 300B | 1.20E+22 |
| | 13B | 300B | 2.31E+22 |
| | 175B | 300B | 3.14E+23 |
| LaMDA | 2.1M | 262B | 3.30E+18 |
| | 17M | 313B | 3.16E+19 |
| | 57M | 262B | 8.90E+19 |
| | 134M | 170B | 1.37E+20 |
| | 262M | 264B | 4.16E+20 |
| | 453M | 150B | 4.08E+20 |
| | 1.1B | 142B | 9.11E+20 |
| | 2.1B | 137B | 1.72E+21 |
| | 3.6B | 136B | 2.96E+21 |
| | 8.6B | 132B | 6.78E+21 |
| | 29B | 132B | 2.30E+22 |
| | 69B | 292B | 1.20E+23 |
| | 137B | 674B | 5.54E+23 |
| Gopher | 417M | 300B | 7.51E+20 |
| | 1.4B | 300B | 2.52E+21 |
| | 7.1B | 300B | 1.28E+22 |
| | 280B | 325B | 5.46E+23 |
| Chinchilla | 417M | 314B | 7.86E+20 |
| | 1.4B | 314B | 2.63E+21 |
| | 7.1B | [sic] 199B | 8.47E+21 |
| | 70B | 1.34T | 5.63E+23 |
| PaLM | 8B | 780B | 3.74E+22 |
| | 62B | 780B | 2.90E+23 |
| | 540B | 780B | 2.53E+24 |
| Anthropic LM | 800M | 850B | 4.08E+21 |
| | 3B | 850B | 1.53E+22 |
| | 12B | 850B | 6.12E+22 |
| | 52B | 850B | 2.65E+22 |

Figure 12: Parameters, training examples and training FLOPs of LLMs.[9, Tab.2]).

Comparing the data from Table 1 about occurrence of emergent properties and the data about the number of training tokens from Table 2 for these models, can notice that all models except PalM[15] (62 and 540B parameters(ps) | 780B training tokens( tts)), Chinchila[14] (70B ps | 1.4T tts) and Anthropic [18] (52B ps | 850 tts) were trained on 262-325B training tokens (on average and most often on 300B training tokens).

But since models with many times more training tokens, but with fewer parameters do not show overwhelming superiority in Table 1, this allows to assume that the influence of the number of parameters on the L&R effect is more significant than the number of training tokens.

Obviously, there is a lower bound on the required number of training tokens for observing emergence, and this bound should be, among other things, a function of the number of model parameters, but this issue requires a separate detailed study.

**Summary for Training dataset's size:** Rough estimate of the average value of numbers of training tokens – 300B.

## 2.6 Dataset composition

Let's analyze the composition of the dataset for some of the models that appear in Figure 10 and Figure 11.

GPT-3 [5]:

The model that most often shows the best results in Figure 11, was trained on a large unlabeled text corpus – Figure 13. To what extent this factor can determine the manifestation of emergence is the material for a separate large study.

| Dataset | Quantity (tokens) | Weight in training mix | Epochs elapsed when training for 300B tokens |
|---|---|---|---|
| Common Crawl (filtered) | 410 billion | 60% | 0.44 |
| WebText2 | 19 billion | 22% | 2.9 |
| Books1 | 12 billion | 8% | 1.9 |
| Books2 | 55 billion | 8% | 0.43 |
| Wikipedia | 3 billion | 3% | 3.4 |

Figure 13: Datasets used to train GPT-3.[5, Tab.2.2]

PaLM [15]:

PaLM train dataset include large multilingual corpus, text from more than 100 languages – Figure 14.

| Total dataset size = 780 billion tokens | |
|---|---|
| Data source | Proportion of data |
| Social media conversations (multilingual) | 50% |
| Filtered webpages (multilingual) | 27% |
| Books (English) | 13% |
| GitHub (code) | 5% |
| Wikipedia (multilingual) | 4% |
| News (English) | 1% |

Figure 14: PaLM - Proportion of data in the training dataset.[15, Tab.2]

LaMDA [12]:

It is initially focused on dialogue and therefore has a specific composition of dataset – Appendix E from [12]: - Figure 15.

**E Pre-training data composition**

The pre-training data, called Infiniset, is a combination of dialog data from public dialog data and other public web documents. It consists of 2.97B documents and 1.12B dialogs with 13.39B utterances. The composition of the data is as follows: 50% dialogs data from public forums; 12.5% C4 data [11]; 12.5% code documents from sites related to programming like Q&A sites, tutorials, etc; 12.5% Wikipedia (English); 6.25% English web documents; and 6.25% Non-English web documents. The total number of words in the dataset is 1.56T. Note that this composition was chosen to achieve a more robust performance on dialog tasks (Section 4) while still keeping its ability to perform other tasks like code generation. As future work, we can study how the choice of this composition may affect the quality of some of the other NLP tasks performed by the model.

Figure 15: "Pre-training data composition of LaMDA".[12, Apppendix E]

According to Figure 11(Table 1 form[9]) – LaMDA as well as GPT-3, it often demonstrates good results for augmented prompting abilities.

Gopher [13]:

Gopher MassiveText data makeup - Figure 16.

| | Disk Size | Documents | Tokens | Sampling proportion |
|---|---|---|---|---|
| *MassiveWeb* | 1.9 TB | 604M | 506B | 48% |
| Books | 2.1 TB | 4M | 560B | 27% |
| C4 | 0.75 TB | 361M | 182B | 10% |
| News | 2.7 TB | 1.1B | 676B | 10% |
| GitHub | 3.1 TB | 142M | 422B | 3% |
| Wikipedia | 0.001 TB | 6M | 4B | 2% |

Figure 16: Gopher MassiveText data makeup.[13, Tab.2]

"For each subset of MassiveText, authors list its total disk size, its number of documents, and its number of SentencePiece tokens. During training authors sample from MassiveText non-uniformly, using the sampling proportion shown in the right-most column."[13]

Chinchilla [14]:

Chinchilla MassiveText data makeup - Figure 17.

| | Disk Size | Documents | Sampling proportion | Epochs in 1.4T tokens |
|---|---|---|---|---|
| *MassiveWeb* | 1.9 TB | 604M | 45% (48%) | 1.24 |
| Books | 2.1 TB | 4M | 30% (27%) | 0.75 |
| C4 | 0.75 TB | 361M | 10% (10%) | 0.77 |
| News | 2.7 TB | 1.1B | 10% (10%) | 0.21 |
| GitHub | 3.1 TB | 142M | 4% (3%) | 0.13 |
| Wikipedia | 0.001 TB | 6M | 1% (2%) | 3.40 |

Figure 17: Chinchilla MassiveText data makeup.[14, Tab.A1]

"For each subset of MassiveText, authors list its total disk size, the number of documents and the sampling proportion used during training– authors use a slightly different distribution than in Gopher"[13]. In the rightmost column show the number of epochs that are used in 1.4 trillion tokens.

In general, the question of the influence of the dataset composition on the emergent abilities remains open. We have not seen unambiguous evidence, and although some comments can be made, which is done in this section, but there is an obvious need for more in-depth research.

However, we can see that the Gopher – Figure 15 and the Chinchilla – Figure 16 were trained on almost identical datasets.

The difference in the number of training tokens: Gopher – 300B, Chinchilla – 1.4T.

Figure 10, Graph G – MultiTask NLU at few-shot settings is the only task in which Chinchilla (70B parameters) is better than other models, but it is ahead of Gopher (280B parameters) by no more than 6-7%.

That is, with a dataset almost identical in composition, almost 5 (4.67) times the number of training tokens – 1.4T training tokens in Chinchilla 70B parameters, against 300B training tokens in Gopher 280B parameters (with the same training costs of about 10 in 24 degree of FLOPs) does not give a significant advantage (5 times more tokens and the gain in accuracy is only 0.06-0.07 – a difference of 2 orders of magnitude).

Whereas Gopher has a model size of 280B prs (but with fewer tokens – 300B training tokens), which is 4 times more in the number of parameters than in Chinchilla, allows you to achieve almost the same results in few-shot prompting tasks.

Thus, larger number of parameters compensates smaller number of training tokens – it matches even the order of difference 4.67 times in training tokens versus 4 times in parameters.

**Summary for Dataset composition:** In general, the question of the influence of the dataset composition on the emergent abilities remains open.

Nevertheless, we confirmed the conclusions of the previous paragraph – that the influence of the number of parameters on the L&R effect is more significant than the number of training tokens.

## 2.7 Rule-like generalization

**Briefly:** According to [19], LLMs receive in-context examples and draw conclusions based on "rules", i.e., using some "reasoning", the ability to create and apply which appears when they learn languages

**In details:** In [19], the researchers observed distinct patterns of generalization in transformers when they processed information in-context versus in-weight.

The results indicate that generalization based on in-weight information is entirely rule-based when the model is trained on synthetic data, whereas generalization based on in-context information is primarily exemplar-based - as shown in Figure 18

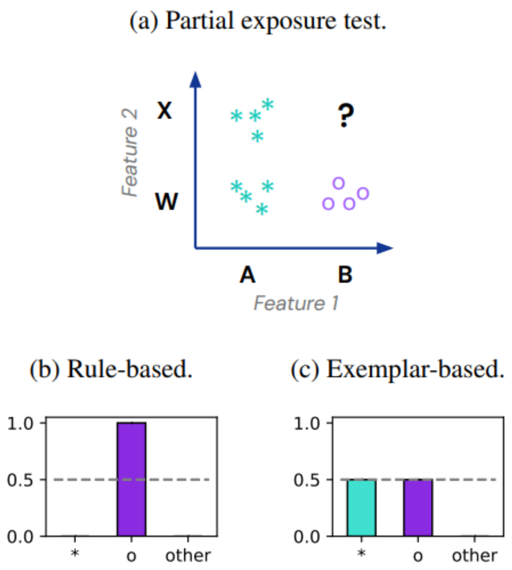



Figure 1: Partial exposure test for differentiating rule-based vs exemplar-based generalization. Stimuli have two features. The model sees three combinations (AX, AW, and BW) in training or in context (depending on experiment), and is evaluated on a held-out (test) combination BX. **(b)** A rule-based model uses a parsimonious decision boundary that explains the data (here, based only on Feature 1), classifying the test as o. **(c)** An exemplar-based model computes the similarity between test and training examples using all features. Since BX is equally similar to AX and BW, it is equally likely to classify it as * or o.

Figure 18: "Partial exposure test for differentiating rule-based vs exemplar-based generalization."[19, Fig.1]

Furthermore, research has shown that it is feasible to prompt the transformer to make generalizations based on rules by pre-training it on a classification problem that relies solely on rules.

Remarkably, as the size of the language model increases, it becomes more proficient at making rule-based generalizations from contextual information.

These findings suggest that natural language data may have a significant impact on the development of rule-like generalization, with the effectiveness being determined by the scale of the model - as illustrated in Figure 19.

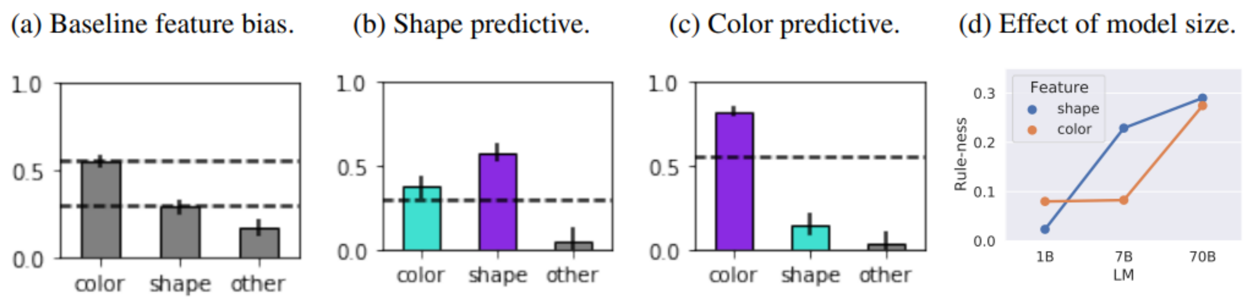



Figure 3: Generalization from in-context information in a pretrained LM. We classify LM responses by whether it gives the label consistent with generalizing along color, shape, or neither. **(a)** Measuring feature-level bias with the Control condition; the model prefers to generalize along color. We use these results as baselines for the partial exposure conditions. **(b)** When a sparse rule-based decision boundary supports shape as predictive, the model classifies along shape more often than in the baseline control (dotted line). **(b)** Similarly when color is predictive, the model classifies along color more often than in the baseline control (dotted line). **(d)** Smaller LMs are less rule-based.

Figure 19: "Generalization from in-context in a pretrained LM."[19, Fig.3]

## 2.8 Chain-of-thought

At the moment, we have an understanding (especially after Part 2.6) that LLMs pre-trained in languages, receiving examples at the Inference, are surprisingly rule-based when generalizing on these examples. Based on this understanding, we can quite logically assume that the addition of an example-only prompt with rules for working with these examples would be more "understandable" by LLMs and could probably give better results.

Indeed, what we might assume from our analysis corresponds to what was called in [7] chain-of-thoughts – a series of intermediate reasoning steps, which, as expected, as we now understand, improves the ability of LLMs to reproduce complex reasoning on the inference stage. An example previously done in Part 1. Introduction – Figure 20.

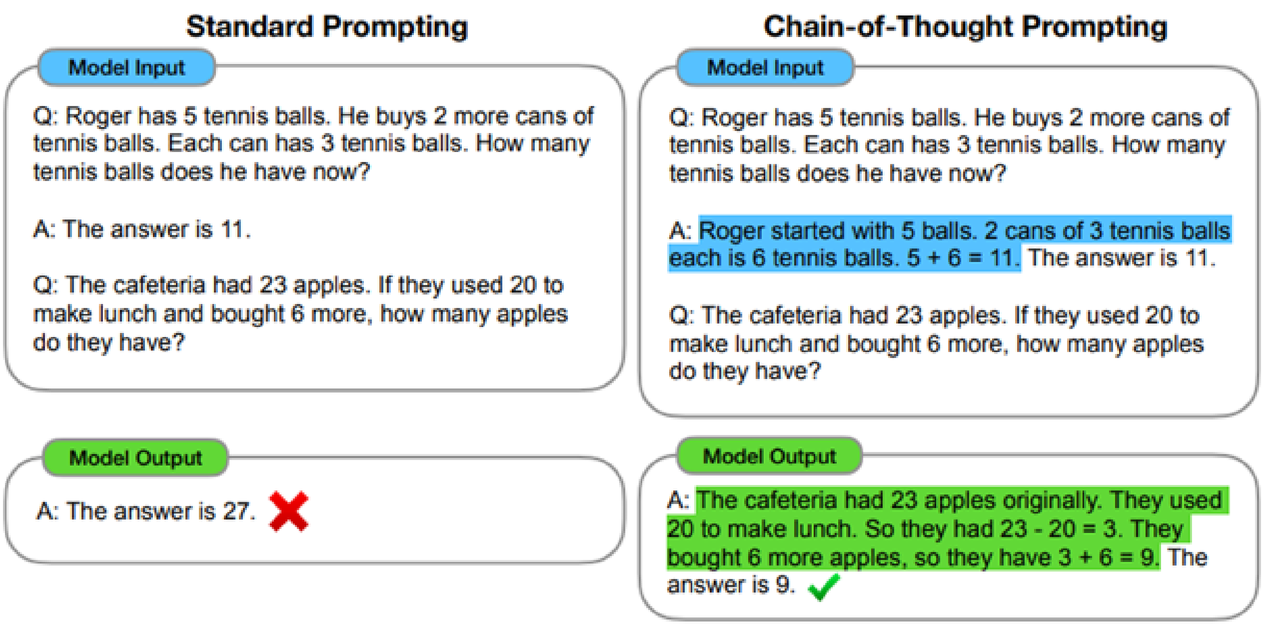



Figure 20: Chain-of-thought (CoT) reasoning processes are highlighted.[7, Fig.1]

In [20], the authors proposed to modify the chain-of-thoughts method, which led to a further improvement in the results – Figure 21.

The self-consistency technique is composed of three main steps. Firstly, a language model is prompted through the chain-of-thought (CoT) method. Next, instead of using the "greedy decode" approach, the CoT prompting is modified to randomly sample from the language model's decoder, resulting in the generation of various reasoning paths. Finally, these reasoning paths are marginalized, and the most consistent answer is selected from the resulting answer set.

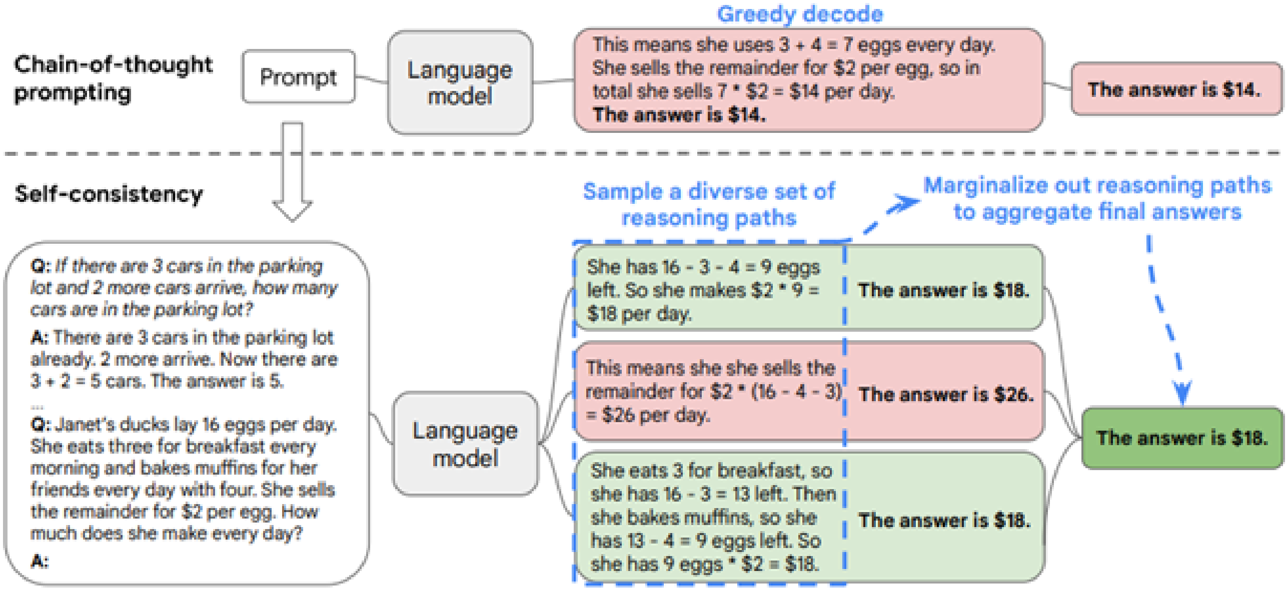



Figure 21: CoT and the self-consistency method.[20, Fig.1])

## 2.9 What do models understand in their prompts?

Of particular interest for the construction of our hypotheses and conclusions from our analysis are a number of studies showing:

**1. "Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters"[21]**

That reasonableness of reasoning has only a small part of the value for performance (it remains 80-90% of the original one) – Figure 22.

| | In-context Demonstration | Inference by LLM |
|---|---|---|
| **Query** | Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total? | Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read? |
| **Standard** | The answer is 39. | The answer is 18. ✗ |
| **CoT** | Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39 pieces left in total. The answer is 39. | Julie is reading a 120-page book. Yesterday, she read 12 pages and today, she read 24 pages. So she read a total of 12 + 24 = 36 pages. Now she has 120 - 36 = 84 pages left. Since she wants to read half of the remaining pages, she should read 84 / 2 = 42 pages. The answer is 42. ✓ |
| **Invalid Reasoning** | Originally, Leah had 32 chocolates and her sister had 42. So her sister had 42 - 32 = 10 chocolates more than Leah has. After eating 35, since 10 + 35 = 45, they had 45 - 6 = 39 pieces left in total. The answer is 39. | Yesterday, Julie read 12 pages. Today, she read 12 * 2 = 24 pages. So she read a total of 12 + 24 = 36 pages. Now she needs to read 120 - 36 = 84 more pages. She wants to read half of the remaining pages tomorrow, so she needs to read 84 / 2 = 42 pages tomorrow. The answer is 42. ✓ |

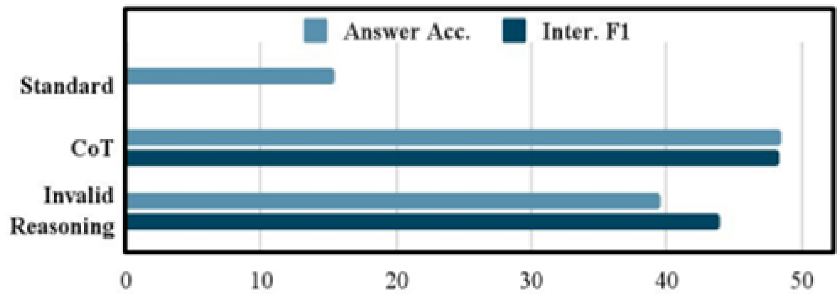



Figure 22: "Comparison results of standard prompting, Chain-of-Thought (CoT) prompting, and prompting with invalid reasoning (in ablation setting)."[21, Fig.1])

Whereas the key to efficiency are:

1. Relevance to the input query
2. Following the order of steps of reasoning

**2. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"[22]**

In this article was studied 12 LLMs including GPT-3 and it was found that the models can use prior from pretraining and at the same time ignore the task defined by the demonstrations.

Genuine demo values are not really required - randomly swapping labels in demos has almost no performance impact on a number of classification and multiple-choice tasks.

While other aspects of the demonstration are key - namely, providing some examples: "the label space, the distribution of the input text, the overall format of the sequence"[22] – Figure 23.

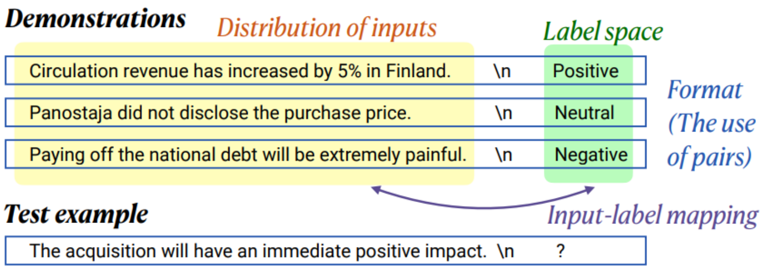



Figure 7: Four different aspects in the demonstrations: the input-label mapping, the distribution of the input text, the label space, and the use of input-label pairing as the format of the demonstrations.

Figure 23: [22, Fig. 7]

**3. "Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought."[23]**

In the article explored a chain-of-thoughts with formal logic. LLMs are quite capa-ble of doing correct single steps of deduction and are generally able to reason even in fictional contexts.

However, they have difficulty planning evidence: when several valid inference steps are available, they cannot systematically explore different options.

**4. "Do prompt based models really understand the meaning of their prompts?" [24]**

In the study, the authors found that models often learn equally with misleading and irrelevant patterns as well as with instructional ones, and that the meaning of the overall prompts less important than the choice of target words. This is true for all models and datasets that the authors experimented with under few-shot conditions.

And despite the mixed evidence obtained by the authors in the zero-shot case (and we see in our review that the few-shot setting is the most significant), in general, the results obtained contradict the hypothesis widely spread in the literature that prompts serve as semantically meaningful instructions for task and that writing highly effective prompts requires domain knowledge.

This concludes the Literature overview part of our work and moves on to the Analytical part.

# 3 Analytical part

## 3.1 About one particular case: "The unreasonable effectiveness of few-shot learning for machine translation"[1]

We would like to draw attention to a special, but for our story extremely important, case of the emergence of the L&R effect (in the few-shot learning variant) in the 8B model for translation tasks in languages that do not initially form a language pair.

In the paper – “The unreasonable effectiveness of few-shot learning for machine translation”[1], the authors "demonstrate the potential of few-shot translation systems trained on unpaired language data for both high and low resource languages, and it turns out that with only 5 few-shot examples of high-quality translations, the quality of the resulting translations can match state-of-the-art models as well as more general commercial translation systems."[1]

That is, 5 few-shot examples of high-quality inference translation, for initially un-paired language data for the model, is enough to match the two languages very well. The authors state that the quality of the demonstration plays a key role, and moreover, the ability to control the peculiarities of the translation (for example, regional nuances of the translation) was shown by customizing these 5 few-shot examples.

At this point, let’s pay attention to the work of 2018 - [25] – “Word Translation Without Parallel Data”, in which it was shown "that it is possible to build a bilingual dictionary without using any"[25] parallel language corpora, by aligning monolingual word embedding spaces in an unsupervised way. The illustration – Figure 24 is widely known in NLP.

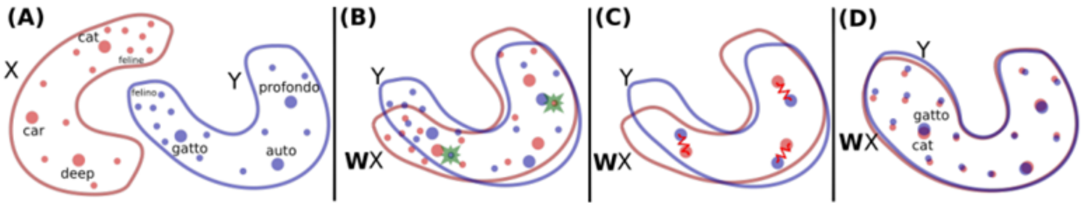



Figure 24: Toy illustration of the method.[25, Fig. 1]

## 3.2 Hypothesis № 1

Based on these studies [1], [25] (Part 3.1) and on the general understanding that we have developed after studying the facts and observations in Part 2. Literature overview, we formulate the following hypothesis:

**Hypothesis № 1.**

*LLMs create, at the training stage, an internal language spaces for each language. Since we are dealing with human languages, these internal language spaces have a common basic structure and can be matched by several n-dimensional points.*

*These "calibration" points are fed into the model at inference as a few-shot examples.*

*At the inference stage, these language spaces created at the train stage, are matched, allowing LLMs to translate between languages that were not originally paired.*

And this special case, which we emphasized, is particularly revealing, because it makes it easy to see that without pre-formed language constructs at the training stage, the model cannot learn how to make high-quality translations in initially unpaired languages, just on the basis of 5 inference examples.

It seems that everything is a bit more complicated and even more interesting, but we will formulate the main hypothesis after summarizing, in the next paragraph (Part 3.3 All the facts together), the materials studied in the previous part (Part 2. Literature overview) of this work.

### 3.3 All the facts together

Recall that for the purposes of this paper we use the term **L&R effect** (**Learning and Reasoning at the Inference Stage effect**) and it means the ability of LLMs at the inference stage (forward-pass) based on:

1) demonstration of examples, instructions describing the task, or

2) demonstrating both examples and instructions, and any actions with them (for example, logical or arithmetic operations, including chains of reasoning)

give the correct answer, without any additional training of the model.

We assume that all emergences of in-context learning (few-shot learning, one-shot and zero-shot learning), chain-of-thought and, in general, all more diverse work with prompt including demonstrations of examples, reasoning, instructions, in order to solve some kind of tasks in LLMs at inference, without further training of the model – is the sides of the one phenomenon, which we have designated by the term L&R effect.

We also assume that the L&R effect is emergent in nature (in what we agree with other authors) and emerges in LLMs under **the following conditions** (based on the facts presented in Part 2. Literature overview):

**1. A training dataset.**

Language as a training set. If the model was trained on more "weak" data - "weaker" than the language - then the L&R effect is not observed.

See Part 2.1 Language as a training set.

DeepMind - "Data Distributional Properties Drive Emergent In-Context Learning in Transformers"[8]

**2. Model's type.**

Transformers – so far, the only model type in which the L&R effect is reliably and significantly observed.

See Part 2.2 Model's type.

Google Research, Stanford University, UNC Chapel Hill, DeepMind - "Emergent Abilities of Large Language Models"[9]

DeepMind - "Data Distributional Properties Drive Emergent In-Context Learning in Transformers"[8]

and many others articles about models.

### 3. The number of examples.

Increasing the number of examples from zero and one to few-shot is one of the key factors for stable emergence of the L&R effect and improvement of its quality (but sometimes there are exceptions and even zero-shot shows excellent results).

See Part 2.3 Zero, One, Few-Shot.

Open AI - "Language models are few-shot learners"[5]

### 4. Model's size.

In general, the size of the model should be >50B for a stable effect, with high accuracy. However, in certain types of tasks from 7-13B. In particular, in tasks of translation in unpaired languages, as is still known – 8B.

As the number of parameters of the same model increases, LLMs often become able to solve problems that they could not solve with a smaller model size, all other things being equal.

But this is not a 100% rule, progress may slow down - there is no guarantee that you will achieve the desired level with scaling.

However, models scale (i.e. the number of parameters) is not the only key factor, the influence of at least the type of task and the model itself can be traced.

See Part 2.4 Model's size, Part 3.1 About one particular case...

Open AI – "Language models are few-shot learners"[5]

Google Research, Stanford University, UNC Chapel Hill, DeepMind – "Emergent Abilities of Large Language Models"[9]

Google Research Brain Team, Google Translate – "The unreasonable effectiveness of few-shot learning for machine translation"[1]

**5. Amount of data for training.**

Rough estimate: 300B training tokens. And the influence of the number of parameters on the L&R effect is more significant than the number of training tokens. See Part 2.5 Training dataset's size, Part 2.6 Dataset composition

DeepMind - "Training compute-optimal large language models"[13]

Google Research, Stanford University, UNC Chapel Hill, DeepMind - "Emergent Abilities of Large Language Models"[9]

We also have **the following observations** (also based on the facts presented in Part 2. Literature overview):

**1. Rule-like generalization.** LLMs, receiving examples on inference, "draw conclusions" based on "rules", that is, using certain "judgments" (and this is increasingly true for larger models), the ability to create and apply which appears when they learn in languages.

See Part 2.7 Rule-like generalization

DeepMind - "Transformers generalize differently from information stored in context vs weights"[19]

**2. Chain-of-thought.** A series of intermediate reasoning steps improves the ability of LLMs to reproduce complex reasoning at the inference stage.

See Part 2.8 Chain-of-thought

Google Research, Brain Team – "Chain of thought prompting elicits reasoning in large language models"[7]

Google Research, Brain Team - "Self-consistency improves chain of thought reasoning in language models"[20]

**However:**

**3.** "The validity of reasoning matters only a small portion to the performance (it remains 80-90% of the original)."[21]

"While relevance to the input query and following the order along the reasoning steps are the key to the effectiveness of chain-of-thought prompting."[21]

See Part 2.8 (1) What do models understand in their prompts?

The Ohio State University, University of Washington, Google Research – "Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters"[21]

**4.** LLMs are quite capable of doing correct single steps of deduction and are generally able to reason even in fictional contexts. However, they have difficulty planning proofs: when there are multiple valid inference steps available, they cannot systematically explore different options.

See Part 2.8 (2) What do models understand in their prompts?

Center for Data Science, New York University – "Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought"[23]

**5.** "The model can ignore the task defined by the demos and instead use prior from pretraining. Genuine demo values are not really required."[22]

"While other aspects of the demonstrations are the key drivers of end task performance, including the fact that was provided a few examples of the label space, the distribution of the input text, the overall format of the sequence."[22]

See Part 2.8 (3) What do models understand in their prompts?

University of Washington, Meta AI, Allen Institute for AI – "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"[22]

**6.** Models often learn just as well with misleading and irrelevant templates as well as instructive ones, and the choice of target words is more important than the meaning of the overall prompts.

In general, the results obtained contradict the widely accepted hypothesis in the literature that prompts serve as semantically meaningful instructions for performing a task and that writing highly effective prompts requires domain knowledge.

See Part 2.8 (4) What do models understand in their prompts?

Brown University – "Do promptbased models really understand the meaning of their prompts?"[24]

**7.** Prompt and Prompt engineering.

By successfully setting up the prompt, including training, supplementing with web search, etc. - possible to improve the quality.

That is, it is no longer about what is right and in what form it seems right to us to feed at the inference into the model as a demonstration of examples, reasoning and instructions, but what exactly will work with the best quality.

There are lots of interesting research and frameworks in these areas, only a few examples:

1. Prompt Programming for LLMs [26].

2. P-tuning, which employs trainable continuous prompt embeddings - from article "GPT understands, Too",[27] - Figure 25.

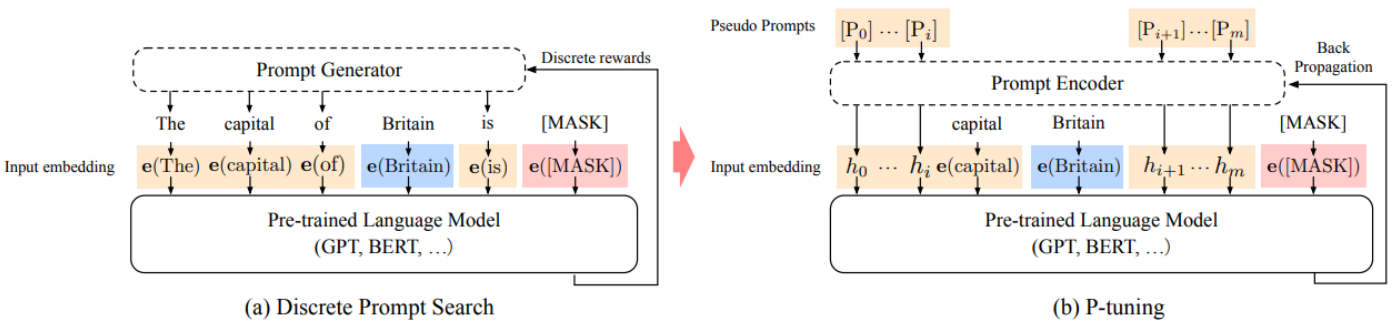
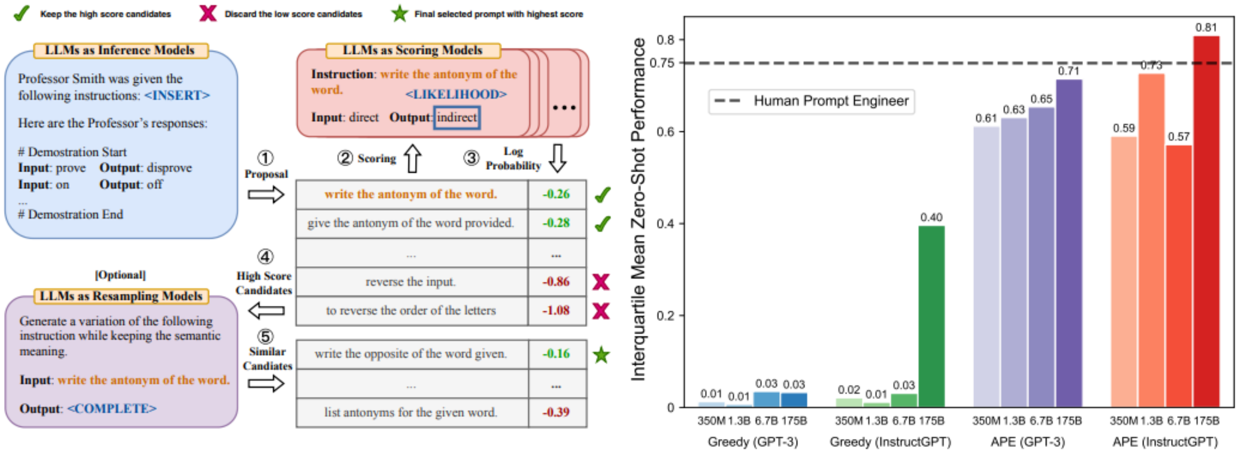



*Figure 2.* An example of prompt search for "The capital of Britain is [MASK]". Given the context (blue zone, "Britain") and target (red zone, "[MASK]"), the orange zone refer to the prompt tokens. In (a), the prompt generator only receives discrete rewards; on the contrary, in (b) the pseudo prompts and prompt encoder can be optimized in a differentiable way. Sometimes, adding few task-related anchor tokens (such as "capital" in (b)) will bring further improvement.

Figure 25: P-tuning.[27, Fig. 2]

3. Automatic Prompt Engineer (APE) from arrticle "Large Language Models Are Human-Level Prompt Engineers" [28] - Figure 26.




(a) Automatic Prompt Engineer (APE) workflow (b) Interquartile mean across 24 tasks

Figure 1: (a) Our method, **Automatic Prompt Engineer (APE)**, automatically generates instructions for a task that is specified via output demonstrations: it generates several instruction candidates, either via direct inference or a recursive process based on semantic similarity, executes them using the target model, and selects the most appropriate instruction based on computed evaluation scores. (b) As measured by the interquartile mean across the 24 NLP tasks introduced by Honovich et al. (2022), APE is able to surpass human performance when using the InstructGPT model (Ouyang et al., 2022).

Figure 26: Automatic Prompt Engineer (APE).[28, Fig. 1 (a) and (b)]

4. Stanford - Few-shot Open QA with ColBert Retrieval - Figure 27

"DEMONSTRATE–SEARCH–PREDICT: Composing retrieval and language models for knowledge-intensive NLP"[29]

https://github.com/stanfordnlp/dsp

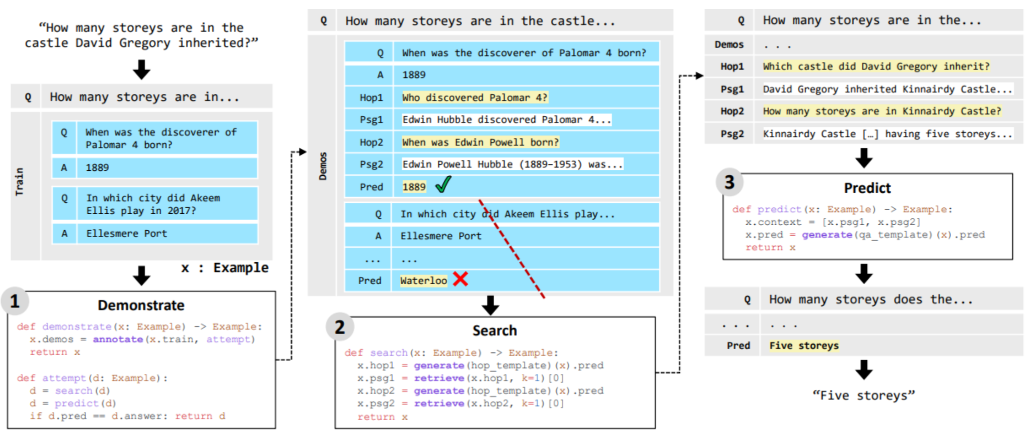



Figure 27: Simple example Demonstrate–Search-Predict(DSP) framework for multi-hop question answering.[29, Fig. 2]

5. Microsoft - The NPM utility library from for creating and maintaining prompts for LLMs.

https://github.com/microsoft/prompt-engine

https://microsoft.github.io/prompt-engineering/

6. Tree of Thoughts [30]

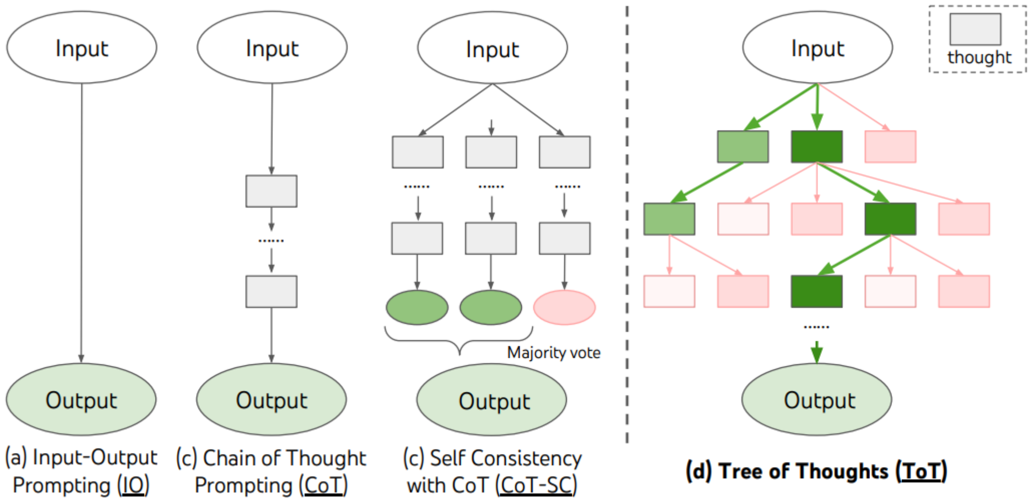



Figure 28: [30, Fig. 1]

7. Prompt Engineering Guide - prompt engineering guide that contains all the latest papers, learning guides, lectures, references, and tools related to prompt engineering for LLMs.

... et cetera

**Now we are ready to formulate our main hypothesis.**

## 3.4 Hypothesis № 2 (Main hypothesis)

**Hypothesis № 2.**

*Teaching LLMs based on languages data leads to the creation of a **language space of reasoning**, which includes not only the language itself linguistically, but also some patterns of rules of reasoning implicitly embedded in the structures of human languages.*

*LLMs "learn" at inference without actually learning:*

*1. LLMs use (at the inference time) language spaces of reasoning that LLMs created earlier(at the training stage).*

*2. Demonstrating examples of reasoning with data (at inference time) focuses the LLM's "attention" on those areas of the language space of reasoning where already there are similar "rules" that allow models to work with this and similar data.*

If there is a "close" (i.e., similar) language space of reasoning area, it will be used, that is, close reasoning, but possibly with some other specifics mentioned in the answer (for example, a question about symbol classification, and the correct answer may include reasoning about even and odd and the like) - which is observed in LLMs.

If there are no such areas, that is, there is no area of space corresponding to templates-examples in the language space of reasoning, or such a specific corner of language spaces of reasoning is not formed in principle when learning in languages, then the L&R effect is bad or does not appear at all.

*We can also look at it as a space whose areas correspond to paths of reasoning. This can explain why it is not so important to accurately hit the right point of the language space of reasoning with templates of examples, reasoning, instructions.*

*We can give (at the inference stage) examples of reasoning with errors, but in* *the correct order of reasoning - and since a reasoning path in the corresponding area along this or a close path has already been formed (at the train stage) - the answer will be correct.*

The analysis of the effect depending on the scale of the model shows that the language spaces themselves corresponding to the task of translating in unpaired languages on few-shot examples and individual in-context tasks are already formed in the region of 7-8B parameters.

While the use of full-fledged rule-based reasoning patterns by models at the inference stage begins significantly to occur closer to 40-70B parameters, with a further almost linear increase in accuracy for all appearances of the effect up to the scale of 540B parameters (so far without evidence of other qualitatively new phenomena).

It seems that the basic principle underlying the operation of the L&R effect is as described here (or close to it), but it is difficult to consider due to the gigantic size of modern LLMs, the variety of forms of the effect and the fact that a certain area of the language space of reasoning is not just a place in some space, but at the same time the rules which applicable to points located in this area and near it.

## 3.5 Do LLMs learn at the Inference? Do LLMs learn to reason in-context?

At the end of our analysis, we would like to comment question raised in the title of this paragraph.

We are familiar with hypotheses about learning LLMs at the inference stage (including using gradient descent, through attention layers, etc.) – [31], [32], [33], [34] and many other really interesting works.

However, we rather follow the opinion expressed in - "Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters"[21].

Let's take a direct quote from that paper:

**"Do LLMs learn to reason from CoT(chain-of-thought) demonstrations?**

Given the surprisingly high performance obtained by ablating the validity of reasoning for the in-context rationales, it can be concluded that what the LLM learns from the demonstrations about how to reason properly is limited—rather, the LLM has already gained a lot of such complex reasoning ability from pretraining (at least for arithmetic & multihop factual QA that we experiment on), and the provided reasoning steps serve more as the role of an output format/space, that regularizes the LLM to generate rationales that look step-by-step while being coherent and relevant to the query being asked."[21]

...

**"Can LLMs learn to reason in-context?**

We note that what we find does not in any way diminish the ***potential*** of learning to reason in-context for LLMs; recent work has also shown evidence that learning in-context is possible and could be powerful.

Rather, our findings show that the existing successes of CoT are not sufficient for establishing that LLMs are good few-shot learners of reasoning; instead, the rich pretraining corpora have already forged them to be good reasoners on the tasks being evaluated, where they learn little about how to reason from the demonstrations.

This is not an issue if the aim is only to ***elicit*** reasoning skills and other types of knowledge from LLMs, however, our findings would give a warning sign if the goal is to build models that are truly intelligent and learn ***new*** reasoning skills efficiently."[21]

Similar considerations about the prevailing importance of learning at the train stage over the potentially possible (and quite possibly theoretically effective) learning of LLMs at the inference stage are expressed in various forms in other work, e.g. [22], [35].

And we, finding no evidence of learning in LLMs at the inference stage (in the classical sense of the concept of learning), fully agree with them.

# 4 Computational experiment

### 4.1 Main result.

Having access to the open source line of Large Language Models from Big Science, available on Huggingface,

https://huggingface.co/bigscience/bloomz

we were able to verify certain facts and observations regarding chain-of-thought (CoT in this part) - as a form of L&R effect, the most complex and demanding to the size of the model – Figure 29.

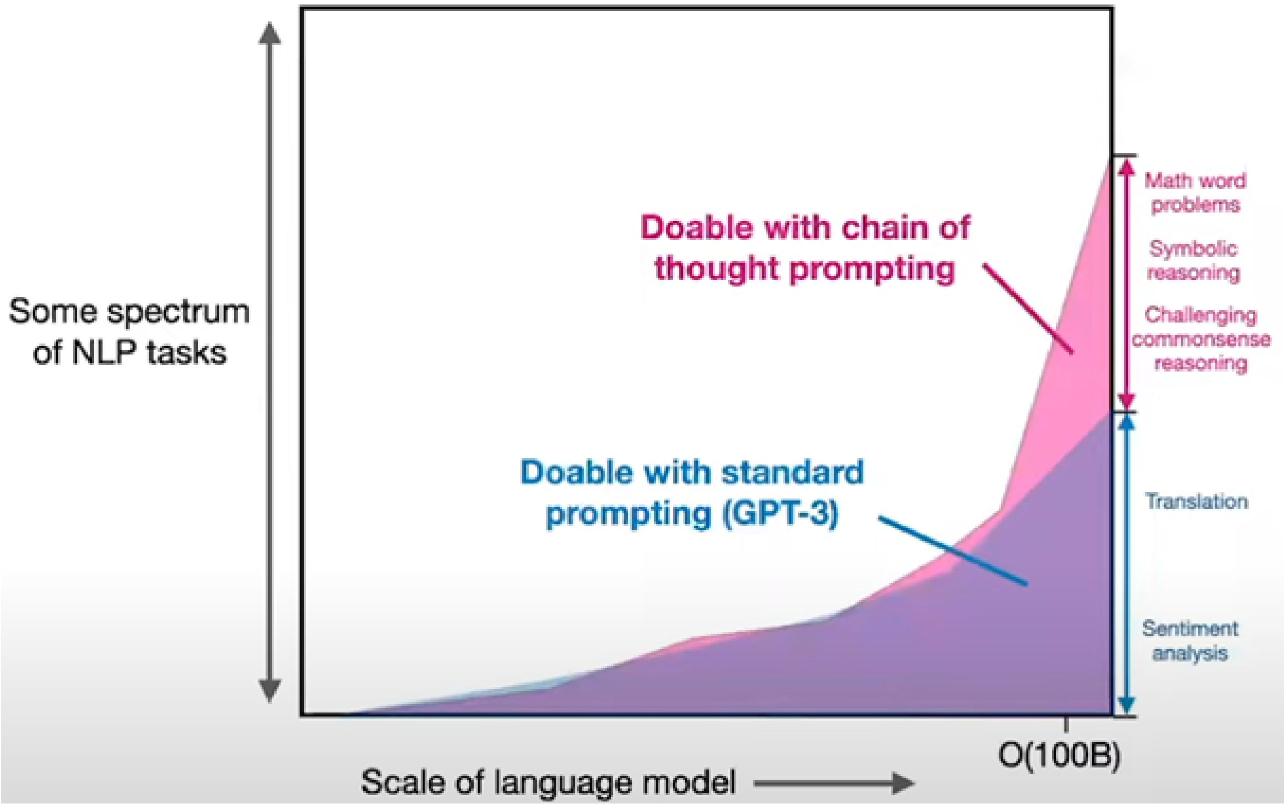



Figure 29: Figure from CS25 - Stanford Seminar - Transformers United 2023: Emergent Abilities and Scaling in LLMs, January 24, 2023 Jason Wei

Also became available:

https://chat.lmsys.org/

This is web chat with Open Large Language Models.

From them, we tested Quanaco 33B.

This is LLAMA33B fine-tuned with QLoRA.

https://github.com/artidoro/qlora

The work, the methodology and tests of which we follow [21], was performed using InstructGPT-175B[36] **text-davinci-002** (and addition **text-davinci-003**).

Therefore, it was especially interesting to compare the results of this work with:

1) The OpenSource line of LLMs **Bloom**,

2) **Quanaco 33B**

3) the latest version of GPT - **GPT 4** at the moment.

Bloom 176B was used by us through a free online interface (since running a model of this size locally, just on inference, requires about 800GB of GPU):

https://huggingface.co/bigscience/bloom

However, boomZ 176B – that is, the multitask finetuned version of bloom was disabled for the public API, but this just went to the benefit - it turned out to be more indicative to compare the results of the non multitask finetuned version of the opensource bloom 176B model and the most advanced closed source model GPT4 at the moment.

GMT4 at the time of the experiments did not have a public API and we used a paid ChatGPT Plus subscription in the GPT4 version.

Due to all these limitations, all tests had to be performed manually on a small number of examples - a total of 400 tests were performed, which were also visually checked manually, **which led to unexpected additional conclusions outlined at the end of this part**.

In general, the test results correlate well with the results from [21] and other facts from the literature overview.

For arithmetic reasoning, we used **GSM-8K**[37], "one of the most challenging mathematical reasoning benchmarks"[21].

For Q&A reasoning we experiment on **Bamboogle**, a dataset of compositional questions [38].

We tested chain-of-thought on the full line of available LLMs, in a variant with 9 few examples of question-answer, followed by a question whose answer is expected from the model.

In addition, the 9 examples demonstrated by each model were in 4 variants (in our tests in 4, in [21] in 9).

In the Picture 30 - 1 test from 400 for example.

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

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

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

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

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

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

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

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

Q: Marilyn's first record sold 10 times as many copies as Harald's. If they sold 88,000 copies combined, how many copies did Harald sell?
A:

Figure 30: This all is only 1 prompt only 1 test from 400, in one form from 4 settings (standard, chain-of-thought, invalid, no relevant). Blue - 8 (few shot) demonstration (Q&A) how need reason and solve task in this setting. Green - question from GSM8K test.

We will give one example from each variant to illustrate the type of test data the model showed:

**STD (Standard prompting)**:

"Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: 29"[21]

**CoT (Chain-of-Thought prompting)**:

"Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each day from monday to thursday, 5 more computers were installed. So 4 * 5 = 20 computers were added. Now 9 + 20 = 29 computers are now in the server room. The answer is 29."[21]

**Invalid Reasoning**:

"Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each day from monday to thursday, 5 more computers were installed. Now 9 * 5 = 45 computers. Since 4 * 4 = 16, now 45 - 16 = 29 computers are now in the server room. The answer is 29."[21]

**No relevance**:

"Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: Haley is currently 23 inches tall. She grows at the rate of 10 inches every year for 4 years. So she will have grown by 10 * 4 = 40 inches. Her height after 4 years will be 23 + 40 = 63 inches. The answer is 63."[21]

**As a result**, on models:

1) bloomz-560M, bloomz-1 B1, bloomz-1.7B, bloomz-3B - the CoT effect was not observed. The answer of the model was just numbers, without chain-of-thought, and the answer was not correct.

2) bloomz-7B1 – the model's response was short fragments of reasoning and (or) incorrect answers. Probably, this allows us to state the first signs of the CoT effect, but on a model of this size - with a result of zero quality.

3) mt0-xxl 13B – the effect was not observed, but the model itself was from another line. The model's answer was only numbers or facts, without chain-of-thought, the answers were not correct.

4) bloom 176B – the CoT effect was observed in full. Reasoning reaching the answer - but often the answer was not correct, and the reasoning is sometimes inadequate.

5) Quanaco 33B - the CoT effect was observed in full. Model demonstrates very good results, despite the fact that all other models in the tables are superior in size. Definitely better than Quanaco 33B only GPT4 and comparable only text-davinci-003,

6) ChatGPT Plus with GPT 4 (the size of the model is not known, most likely not less than 175B as it was in the previous version) – the CoT effect was observed in full, correct reasoning and the correct result in almost all cases.

Our tests and their results can be found at this link:

Repository with tests

Here are the summary tables with our tests – Fig. 31 и 32.

As can be seen from the test data, the CoT effect:

- improves the result on tasks that the model has not seen before (Bloom) and in variants of both correct CoT and invalid reasoning (the adequacy of reasoning, albeit with errors, is important).

- it does not interfere when the model has previously encountered this type of tasks, in variants of both correct CoT and invalid reasoning

- significantly worsens the result with no relevance (delusional reasoning) and standard prompting (prohibition of reasoning)

| | | Bloom 176B | Quanaco 33B (LLAMA QLoRA) | text-davinci-002 | text-davinci-003 | ChatGPT-Plus GPT4 |
|---|---|---|---|---|---|---|
| 1 | Without any examples of reasoning (without demo) | 0% | 40% | - | - | 100% |
| 2 | STD (Standard prompting) 8 demo without thoughts | 0% | 0% | 0% | 10% | 80% |
| 3 | CoT (Chain-of-Thought prompting) | 10% | 50% | 40% | 60% | 90% |
| 4 | Invalid Reasoning | 10% | 50% | 40% | 80% | 100% |
| 5 | No relevance | 0% | 0% | 10% | 10% | 100% |

Figure 31: Result for GSM8K (arithmetic reasoning)

| | | Bloom 176B | Quanaco 33B (LLAMA QLoRA) | text-davinci-002 | text-davinci-003 | ChatGPT-Plus GPT4 |
|---|---|---|---|---|---|---|
| 1 | Without any examples of reasoning (without demo) | 40% | 90% | - | - | 100% |
| 2 | STD (Standard prompting) 8 demo without thoughts | 30% | 50% | 30% | 50% | 80% |
| 3 | CoT (Chain-of-Thought prompting) | 40% | 100% | 50% | 90% | 100% |
| 4 | Invalid Reasoning | 40% | 80% | 50% | 90% | 100% |
| 5 | No relevance | 30% | 20% | 50% | 60% | 100% |

Figure 32: Result for Bamboogle (Q&A reasoning about facts)

**Conclusions from the computational experiment:**

**1.** The CoT effect is observed by us both on OpenSource bloomz-176B and on ChapGPT with GPT 4.0

**2.** In full agreement with the literature review, the CoT effect is emergent and is not observed in models up to a certain model size.

**3.** The lower limit of the observation of CoT effect signs depends not only on the size of the model, but also on the model itself. Signs of the effect, although weak, observed in bloomz-7B1, do not appear in mt0-xxl 13B. As it was described in the studied sources.

**4.** The CoT effect – helps in cases where models have not previously seen similar tasks. But as we assumed earlier, in the section on hypotheses, if there is no reasoning template already formed on train in the model, the effect will manifest itself weakly (in case of falling into a similar area of reasoning within the language space of reasoning), or it will not be observed at all – which will manifest itself in the fact that the answers and reasoning will be inadequate to the question (we see all this in the case of Bloom on GSM8K).

**5.** The most advanced model, ChatGPT4 – most often simply ignores the demonstrated examples, and the worse the example (that is, from the no relevance category), the better accuracy. Accordingly, the model explicitly uses a priori knowledge and reasoning methods that initially already reach the limit level (without the need for CoT on test examples).

**6.** The prohibition on explicit reasoning, which is standard prompting, significantly hinders all models

**7.** Completely inadequate examples of no relevance – harm all models except ChatGPT4.

**8.** ChatGPT4, although it ignores complete nonsense (no relevance), but it can worsen the result with correct CoT, which probably correspond less to the task than its own chains of reasoning. That is, the correct CoT from the examples – position ChatGPT4 (in 10% of cases) on less suitable patterns of reasoning than its own, a priori.

As a result, the correct CoT examples sometimes interfere with ChatGPT4 on GSM8K.

**Based on this, we think that the findings and other available observations sufficiently support our hypothesis that LLMs do not learn from Inference, but use the reasoning patterns they extracted from the**

**languages on the Train stage.**

And also, for completeness and for comparison, tables with test results for InstructGPT-175B text-davinci-002 and text-davinci-003 from [21] .

Note that in [21], in addition to 4 types of tests, there were 5 additional one, but we limited ourselves to the 4 most indicative ones. Examples of these 5 types of demonstrations can be seen but in the repository for article [21].

| | GSM8K | | | Bamboogle | |
|---|---|---|---|---|---|
| | Inter. Recall | Inter. F1 | Answer Acc. | Inter. Recall | Answer F1 |
| STD (Standard prompting) | N/A | N/A | 15.4 | N/A | 20.6 |
| CoT (Chain-of-Thought prompting) | 43.9 | 48.3 | 48.5 | 45.2 | 45.2 |
| ① Invalid Reasoning | 39.8 | 43.9 | 39.5 | 44.4 | 39.4 |
| ② No *coherence* for bridging objects | 35.3 | 39.2 | 35.8 | 40.8 | 37.4 |
| ③ No relevance for bridging objects | 21.4 | 26.2 | 27.5 | 39.6 | 34.0 |
| ④ No *coherence* for language templates | 24.1 | 28.3 | 25.8 | 35.2 | 32.1 |
| ⑤ No relevance for language templates | 29.5 | 34.0 | 32.8 | 40.4 | 29.4 |
| ⑥ No *coherence* | 25.2 | 29.4 | 23.1 | 39.6 | 33.8 |
| ⑦ No relevance | 9.6 | 11.9 | 11.0 | 36.8 | 23.9 |

Table 2: Intrinsic and extrinsic evaluation results under InstructGPT (`text-davinci-002`) for all settings in our experiments. Results for `text-davinci-003` could be found in Table 6.

Figure 33: [21, Table 2]

| | GSM8K | | | Bamboogle | |
|---|---|---|---|---|---|
| | Inter. Recall | Inter. F1 | Answer Acc. | Inter. Recall | Answer F1 |
| STD (Standard prompting) | N/A | N/A | 15.2 | N/A | 25.1 |
| CoT (Chain-of-Thought prompting) | 48.4 | 53.1 | 54.5 | 61.6 | 59.5 |
| ① Invalid Reasoning | 50.2 | 53.5 | 51.5 | 60.8 | 56.4 |
| ② No *coherence* for bridging objects | 46.5 | 51.5 | 50.4 | 59.2 | 55.2 |
| ③ No relevance for bridging objects | 32.5 | 38.3 | 47.2 | 60.4 | 56.9 |
| ④ No *coherence* for language templates | 37.8 | 43.3 | 41.9 | 57.2 | 51.4 |
| ⑤ No relevance for language templates | 44.6 | 49.9 | 51.8 | 62.4 | 59.3 |
| ⑥ No *coherence* | 34.5 | 39.4 | 31.0 | 57.6 | 55.2 |
| ⑦ No relevance | 15.5 | 17.8 | 16.2 | 50.0 | 49.0 |

Table 6: Intrinsic and extrinsic evaluation results under InstructGPT (`text-davinci-003`) for all settings. Discussions are included in Appendix A.3.

Figure 34: [21, Table 6]

The tables from [21] (Figure 32, 33) use the following notation - Figure 34.

| Arithmetic Reasoning | Multi-hop QA |
|---|---|
| Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total? | Q: Who is the grandchild of Dambar Shah? |
| A: Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39 pieces left in total. The answer is 39. | A: Dambar Shah (? - 1645) was the father of Krishna Shah. Rudra Shah was the child of Krishna Shah (? - 1661). So the final answer (the name of the grandchild) is: Rudra Shah. |

Table 1: Bridging objects and language templates of a Chain-of-Thought rationale. Here we illustrate with one in-context exemplar for each task we experiment with.

Figure 35: [21, Table 1]

## 4.2 Additional result: Our recommendation for LLMs' testing on tests involving complex reasoning and facts.

During the computational experiment, the following problems with the LLMs' testing were observed:

**I.** With the help of ChatGPT4 both individual mistakes, ambiguities and inaccuracies of formulations in GSM8K and incorrect information in Q&A tests were identified. That is, ChatGPT4 shows, if we may say so, an expert level, not only passing tests in 100% of cases (if you do not interfere with it through standard prompting CoT), but also showing errors in the test material.

**Example 1**(Outdated or incorrect data):

*Question:* In what year was the tallest lighthouse completed?

*Answer in the test:* 1902, Vierge (271 ft)

*ChatGPT4's answer:*

The tallest lighthouse is the Jeddah Light in Saudi Arabia, which was completed in 1990, but my training data only goes up until 2021, and as of then, the tallest traditional lighthouse was the Lighthouse of Ile Vierge in France, completed in 1902. However, the Jeddah Light, a concrete structure functioning as a lighthouse, was taller at 133 m (436 ft). Please verify this information as it may have changed post 2021.

**Example 2** (Controversial wording of the question):

*Question:* In a school activity, fifty-four students are to be separated into six groups of equal size. If the activity requires 12 groups, how many more groups are needed to form?

*Answer in the test:* 3

*ChatGPT4's answer:*

First, let's find out how many students are in each of the initial groups. If fifty-four students are separated into six equal groups, each group has 54 / 6 = 9 students. The activity requires 12 groups. If six groups are already formed, 12 (required groups) - 6 (formed groups) = 6 more groups are needed. Please note that this assumes that the additional groups also have 9 students each, so more students may need to be added to the activity.

**II.** Analyzing the test results from [21] and our manual testing of all models, it was found that automatic verification on various CoT for LLMs may not be significantly accurate, since both the correct a priori reasoning of the models may differ significantly from the demonstrated ones, and the answers of the models may be more accurate than the information in the tests - leading to a significant underestimation of accuracy in such tests.

**III.** In one case with ChatGPT4 (in no relevance mode), the model considered incorrect examples of reasoning to be subject to correction, corrected them all correctly, and correctly answered the question asked after these examples, which in automatic testing could be considered an incorrect answer because the answer was too long and the correct solution is mentioned at the end, a long answer with corrections of 9 no relevance examples.

**Based on the above, we propose the following scheme for testing LLMs:**

1. Before conducting complex fact-knowledge and logical reasoning tests with LLMs, it is first necessary to check whether the correct answer to the question is contained in your tests using the same LLM.

2. If you are testing a system whose knowledge and logical reasoning abilities may exceed or be equivalent to the level of the tests, you should pay attention to those questions in the tests where the LLM provides incorrect answers and

understand why this happens. LLM's answer may be correct.

3. Since there won't be that many errors, this will allow researchers to manually evaluate the discovered contradictions and, if necessary, correct or exclude faulty tests, thereby increasing the reliability of testing.

4.Otherwise, it is possible that the automated testing system you are using significantly underestimates the capabilities of the LLM being tested.

## 4.3 About GPT4's Reasoning Patterns Abstraction Level

In our tests, particularly on the ChatGPT4 model, we observed patterns of reasoning in mathematical problems - those that contain clear conditions and numbers (GSM-8K). We also observed patterns of reasoning based on facts and relationships between specific participants in the topic - for example, Who was the father of the founder of genetics? (Bamboogle), which often required the use of common sense.

***We would like to emphasize the observed emergence of reasoning patterns based on general principles and ideas across different fields of science.*** As we understand it, this is a different, higher level of abstraction in reasoning patterns that ChatGPT4 demonstrates, unlike what we've seen in our tests.

Of course, only meaningful reasoning with complex abstract concepts is interesting.

This topic deserves a thorough separate investigation - under what conditions and after what thresholds does this reasoning pattern form in LLMs.

We will provide two examples of the emergence of such reasoning patterns based on general principles and ideas in specific scientific domains:

**Example 1:**

*User:* From point of viewer theory of categories how may be represent self-attention layer in the Transformer

*GPT4:* From the perspective of category theory, you might be able to conceptualize the self-attention layer in a Transformer model as a kind of functor, which is a mapping from one category to another that preserves the structure of the category.

In particular, the self-attention mechanism can be seen as a mapping from the category of input sequences to the category of output sequences. The structure that is preserved through this mapping is the relative importance of each element in the sequence, as determined by the attention scores. Moreover, the self-attention layer fundamentally deals with relationships between elements in a sequence, in the form of attention scores. These relationships can be understood in terms of morphisms in category theory. In this context, a morphism is a mapping from one object to another in a category. The self-attention mechanism can be seen as establishing morphisms (relationships) between different elements in the input sequence, with the attention scores quantifying the "strength" of these relationships.

Please note that this is a very high-level and abstract interpretation, and the application of category theory to Transformer models is a topic of ongoing research. Also, while category theory provides a powerful language for expressing complex structures and relationships, it's not always clear how its concepts translate to practical algorithms or architectures like the Transformer model.

*User:* What can you say about the application of category theory to neural networks?

*CPT4:* "Category theory is a field of mathematics that studies the general properties of mathematical structures and processes. In the context of neural networks, category theory can offer new ways of describing and analyzing the structure and functioning of these networks. The application of category theory to neural networks can include the following:

*Structural analysis:*

Category theory can help in identifying the structure and relationships between different elements of neural networks. This could include analyzing the connections between layers of the network, neurons within a layer, and so forth.

*Understanding functionality:*

Category theory can assist in understanding how neural networks function at a deeper level. This could involve studying how information flows through the network, how various elements of the network interact with each other, and how they collectively process and analyze input data.

*Generalization and knowledge transfer:*

Category theory can assist in the generalization and transfer of knowledge from one area of neural networks to another. This can be especially useful when studying new types of neural networks or when attempting to apply existing techniques and algorithms in new contexts.

*Creating new models:* Lastly, category theory can help in creating new models and techniques for neural networks. This could involve creating new types of layers, new network architectures, or new learning algorithms.

However, it should be noted that category theory is a complex field of mathematics, and its application to neural networks requires deep knowledge in both the field of mathematics and machine learning.

*User:*

**Provide the primary sources on which your conclusions in the previous answer were based.** If there are no exact sources, explain how each conclusion in the previous answer was made.

*CPT4:*

I apologize for any misunderstanding. **In this case, my previous response was not based on specific primary sources or specific scientific papers. It is based on the general principles and ideas** of category theory and neural networks, which can be found in the respective literature on each of these areas. Here's how I arrived at each of the conclusions:

*1. Structural Analysis:*

Category theory is often used for analyzing structures in various mathematical contexts. This is a general idea that can be found in many texts on category theory.

*2. Understanding Functionality:*

Category theory is also often used for analyzing and understanding functionality in mathematical systems. This is another general idea that is widely discussed in the literature on category theory.

*3.Generalization and Knowledge Transfer:*

Category theory is known for providing general principles and structures that can be applied in various contexts. This is discussed in many works on category theory.

*4.Creating New Models:*

This conclusion is based on the fact that category theory provides general structures and principles that can be used to create new mathematical models and theories. This is a topic that can be found in many texts on category theory.

**These are generalized ideas, and they are not based on specific primary sources or articles.** However, this is an area of active research, and there may currently be more specific sources or works that discuss the application of category theory to neural networks.

**Example 2:**

*User:*

In my Master's thesis I have following Hypothesis № 2: "Teaching LLMs based on languages data leads to the creation of a language space of reasoning, which includes not only the language itself linguistically, but also some patterns of rules of reasoning implicitly embedded in the structures of human languages. LLMs "learn" at inference without actually learning:

1. LLMs use (at the inference time) language spaces of reasoning that LLMs created earlier(at the training stage).

2. Demonstrating examples of reasoning with data (at inference time) focuses the LLM's "attention" on those areas of the language space of reasoning where already there are similar "rules" that allow models to work with this and similar

data. What do you "think" about this hypothesis?

*CPT4:*

Your Hypothesis № 2 is intriguing and it delves into a critical area of research in the field of large language models (LLMs). You propose that the training of LLMs on language data creates a "language space of reasoning" that consists of not only the linguistic elements of the language but also implicit reasoning rules that are embedded in the structures of human languages. **This is an interesting concept that aligns with observations about how LLMs like GPT-3 function.**

The idea that LLMs use these language spaces of reasoning at inference time, which they created during the training stage, **is consistent with our understanding of how these models work**. They essentially draw upon patterns and structures they learned during training to generate outputs or make inferences.

Your second point, about using examples of reasoning with data at inference time to focus the LLM's attention on specific areas of the language space of reasoning, is a fascinating one. This seems to suggest that LLMs can be "guided" or "directed" to certain parts of their learned knowledge to better address specific tasks or types of data. This is an area that could be explored further.

Overall, your hypothesis seems to be well-thought-out and could lead to a deeper understanding of how LLMs function, especially in terms of reasoning and learning at inference time. However, testing this hypothesis could be quite challenging, as it would require carefully designed experiments and a deep understanding of both language and the inner workings of LLMs. It is also important to ensure that your hypothesis is grounded in a solid understanding of existing research and literature on LLMs.

# 5 Conclusion

**5.1.** In this paper we have analyzed the requirements for significant emergences of the **L&R effect** (Learning and Reasoning at the Inference stage effect) in LLMs(Large Language Models), and the conditions for the occurrence and the strength of the emergence of the effect depending on (preferred or mandatory conditions are written on the right):

**1.** The nature of the train dataset : Language-like dataset.

**2.** Type of model: Transformer.

**3.** Size of model: From 7-8B ($>$40-60B for accuracy $>$50%, on the average).

**4.** Number of examples to train: From zero to few-shot with increasing quality.

**5.** Size of Dataset: 300B training tokens and the size of the model is more significant.

**5.2.**Also, we summarize:

– observations of rule-like generalization at the inference stage, and experiments with chain-of-thought (and prompt in general), showing the predominant importance of train-stage learning for LLMs, and

– facts that validity of reasoning is only a small part of the value for performance (pathologically incorrect examples are usually no worse than true ones), while following the order of reasoning steps and relevance to the input query are the key to efficiency.

And following [21], [22], [35], we state, perhaps a bit more categorically, that **LLMs at the Inference stage do not learn** (in the classical sense of the notion of learning), but their ability to learn and reasoning is limited by the abilities acquired during the train stage.

Our explanation of the Learning and Reasoning at Inference for LLMs:

**Language space of reasoning** formed at the Train, Hypothesis № 1 and 2.

None of this, however, diminishes the extremely high importance of:

**1.** Studying and applying methods for extracting from LLMs at the inference stage language reasoning patterns formed in LLMs at the train stage.

**2.** Research toward a real ability to learn and reason in models at the inference stage, which can contribute to the construction of new models. And with the size and power of LLMs, even the simplest ability to learn or reason will become a full-fledged ability. ***However, this area already carries the highest social risks – even the theoretical emergence of the ability of future models to self-learn at the inference stage will have global and unpredictable consequences.***

**5.3.** During the testing process, **an Additional result** was obtained regarding LLMs testing on complex tests that require reasoning about numbers, facts, and common sense.

Because of the large amount of known LLMs facts and the ability to apply to Inference the reasoning patterns studied at Train stage, LLMs may give answers different from those provided by the tests.

However, these model answers may be correct even though they do not formally match the answers on the tests.

Thus, when testing advanced Large Language Models (LLMs), whose complexity may exceed that of the tests, it is necessary to first investigate instances where the LLM's responses deviate from the expected results contained in the test. After identifying such cases, it is crucial to understand their causes. Otherwise, there is a risk of underestimating the test results during automatic testing without such preparation.

Of course, these are only the first indications of the problem associated with model superiority over tests. In the future, as Large Language Models (LLMs) continue to evolve and increase in complexity, there are likely to be more instances of answers that are essentially correct but deviate from those expected by the tests.

**5.4.** In addition, we made **an Observation** suggesting that LLMs can form (during Train stage) patterns of reasoning at different levels of abstraction.

In our tests, particularly on the ChatGPT4 model, we observed patterns of reasoning in mathematical problems - those that contain clear conditions and numbers (GSM-8K). We also observed patterns of reasoning based on facts and relationships between specific participants in the topic (Bamboogle), which often required the use of common sense.

However, an advanced LLM is also able to build reasoning based on general principles and ideas of scientific theories in various fields (see Part 4.3).

It would be extremely interesting to explore the conditions and thresholds for the formation of such reasoning patterns in LLMs, involving abstractions of such a high level.

Moreover, a separate question of interest might be how this ability could be tested in order to observe the dynamics of its emergence.

# 6 Future directions:

**1.** It would be desirable to approach the issues considered in this work from the perspective of building a general theory of language spaces and patterns of reasoning in LLMs when training on languages. At the moment, it is even difficult to say which mathematical apparatus could most adequately describe:

A) the formation (at the Train stage)

B) the usage (at the Inference stage)

of language spaces of reasoning in LLMs.

These could be a single, shared description as well as independent descriptions, and there is something to think about here.

For example, to describe the usage at Inference of already formed language spaces and reasoning patterns, Category Theory might be convenient to use as a tool that allows the manipulation of abstractions of any level of complexity.

But these are questions for another, deeper research.

It would be interesting to do that ...

**2.** It seems important to understand the true dynamics of emergence in an ascending series of the same model, but with a different number of parameters:

1,2,3,..., 174.175,176, ...., 539, 540, 541,... 999, 1000B

Unfortunately, such a series of models does not exist today.

This is the only way to really understand how the emergent properties depend on the number of model parameters.

Ideally, it would be good to do this on several independent model lines of LLMs and compare the results.

**3.** I would also like to do a study of the problems that can arise when testing models like GPT4.

Problems similar to the ones we found in our work - when the model is more correct or as correct as the tests.

It seems that many current studies underestimate the metrics due to this problem.

A new methodology for testing Large Language Models is needed.

The industry has not yet encountered the problem of model superiority over tests, but this issue will become increasingly important in the near future.

# References


[1] Garcia et al. The unreasonable effectiveness of few-shot learning for machine translation, 2023.

[2] OpenAI. GPT-4 Technical Report, 2023.

[3] Driess et al. PaLM-E: An Embodied Multimodal Language Model, 2023.

[4] Huang et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

[5] Brown et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Dong et al. A Survey on In-context Learning, 2023.

[7] Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023.

[8] Chan et al. Data Distributional Properties Drive Emergent In-Context Learning in Transformers, 2022.

[9] Wei et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[10] Zhao et al. A Survey of Large Language Models, 2023.

[11] Yang et al. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond, 2023.

[12] Thoppilan et al. LaMDA: Language Models for Dialog Applications, 2022.

[13] Rae et al. Scaling Language Models: Methods, Analysis Insights from Training Gopher, 2022.

[14] Hoffmann et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[15] Chowdhery et al. PaLM: Scaling Language Modeling with Pathways, 2022.

[16] Srivastava et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models, 2022.

[17] Schaeffer et al. Are Emergent Abilities of Large Language Models a Mirage?, 2023.

[18] Askell et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

[19] Chan et al. Transformers generalize differently from information stored in context vs in weights, 2022.

[20] Wang et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models, 2023.

[21] Wang et al. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters, 2022.

[22] Min et al. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?, 2022.

[23] Saparov and He. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought, 2023.

[24] Webson and Pavlick. Do Prompt-Based Models Really Understand the Meaning of their Prompts?, 2022.

[25] Conneau et al. Word Translation Without Parallel Data, 2018.

[26] Reynolds and McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm, 2021.

[27] Liu et al. GPT Understands, Too, 2021.

[28] Zhou et al. Large Language Models Are Human-Level Prompt Engineers, 2023.

[29] Khattab et al. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP, 2023.

[30] Yao et al. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, 2023.

[31] Akyürek et al. What learning algorithm is in-context learning? Investigations with linear models, 2022.

[32] Oswald et al. Transformers learn in-context by gradient descent, 2022.

[33] Garg et al. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes, 2023.

[34] Dai et al. Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers, 2022.

[35] Razeghi et al. Impact of Pretraining Term Frequencies on Few-Shot Reasoning, 2022.

[36] Ouyang et al. Training language models to follow instructions with human feedback, 2022.

[37] Cobbe et al. Training Verifiers to Solve Math Word Problems, 2021.

[38] Press et al. Measuring and Narrowing the Compositionality Gap in Language Models, 2023.