

**From the holdings of the Department of Special Collections , Princeton University Library,
Princeton, NJ**

These documents are either in the public domain, or can only be used for educational and research purposes (Fair use) as per U.S. Copyright law (text below). By accessing this file, all users agree that their use falls in the public domain or within fair use as defined by the copyright law. We do not charge any permission or usage fees for the publication of images of material in our collections, including those provided by Princeton University Library, via our website, catalog, or directly from staff. For more information regarding copyright, credit, and citation please review the information on the Copyright, Credit and Citation Guidelines page of our website.

<https://library.princeton.edu/special-collections/policies/copyright-credit-and-citation-guidelines>. If you have any questions, including how to order higher-quality images from the collection, contact Special Collections Public Services via our Ask Us! Form. <https://library.princeton.edu/special-collections/ask-us>

Master's Theses, 2022-2026

Computer Science, 2023

2023

What In-Context Learning “Learns”

In-Context: Disentangling Task Recognition and Task Learning

Pan, Jane

Princeton, NJ : Princeton University

<http://arks.princeton.edu/ark:/88435/dsp01ms35tc92q>

U.S. Copyright law: The copyright law of the United States (Title 17, §108, United States Code) governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of fair use that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve a violation of copyright law. <https://www.copyright.gov/title17/title17.pdf>

WHAT IN-CONTEXT LEARNING “LEARNS”
IN-CONTEXT: DISENTANGLING TASK
RECOGNITION AND TASK LEARNING

JANE PAN

ADVISOR: DANQI CHEN

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN ENGINEERING
DEPARTMENT OF OPERATIONS RESEARCH AND FINANCIAL ENGINEERING
PRINCETON UNIVERSITY

MAY 2023

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Jane Pan

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Jane Pan

Abstract

Large language models (LLMs) exploit in-context learning (ICL) to solve tasks with only a few demonstrations, but its mechanisms are not yet well-understood. Some works suggest that LLMs only recall already learned concepts from pre-training, while others hint that ICL performs implicit learning over demonstrations. We characterize two ways through which ICL leverages demonstrations: *task recognition* (TR) captures the extent to which LLMs can recognize a task through demonstrations – even without ground-truth labels – and apply their pre-trained priors; *task learning* (TL) is the ability to capture new input-label mappings unseen in pre-training. Using a wide range of classification datasets and three LLM families (GPT-3, LLaMA, and OPT), we design controlled experiments to disentangle the roles of task recognition and task learning in ICL. We show that (1) models can achieve non-trivial performance with only task recognition, and task recognition does not further improve with larger models or more demonstrations; (2) LLMs acquire task learning as the model size scales; and (3) the correlation between model sizes and task learning performance strengthens with the number of demonstrations. Our findings unravel two different forces behind ICL and we advocate for discriminating them in future ICL research due to their distinct nature.

Acknowledgements

First and foremost, I thank the people at the Princeton NLP group. Both as colleagues and as friends, they defined my experience at Princeton, which has become the most formative years of my life. Their advice and encouragement was critical in supporting the first steps of my academic career. I do not exaggerate when I say that I regard each and every one of them as a role model.

I am infinitely fortunate to have Professor Chen as my advisor. The insight, direction, and kindness that I experienced under her guidance are the reason why I was lucky enough to gain the opportunities that I have today. I am also grateful for the help of my mentors, Tianyu and Howard. Without their help and patience (more than I deserve), this project would certainly never have been completed. I will always look up to them as model students and researchers.

In all my life, I have been impossibly lucky with my friendships. My greatest thanks to Noah, who never stopped believing in me. This unconditional faith has been the most precious gift I've received in my life; my dues to him grow deeper with each year. His kindness, wisdom, patience, and perseverance – I prize them above all else, and hope I can repay a fraction of what I owe.

I am grateful for Bonnie for her gracious support and good cheer, as well as Kathleen, for our sixteen years of best-friendship.

I am indebted also to Austin; rare are those who are an anchor in both happy and hard times. I thank Alex, too, who was always ready with good advice and a listening ear. (And to both for putting up with my haphazardly strewn shoes. Sorry!)

Finally, I am grateful for new hopes and new horizons, which bring out the best from me today.

To my family, to whom I owe everything.

Contents

Abstract	iii
Acknowledgements	iv
List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 The Rise of In-Context Learning	1
1.1.1 How Does In-Context Learning Work?	2
1.2 Research Design	4
1.2.1 Key Questions	4
1.2.2 Research Design	4
1.2.3 Key Findings	5
2 Related Work	6
2.1 Pre-trained Language Models	6
2.1.1 Pre-trained Language Model Families	7
2.2 Prompting	8
2.3 Scaling Models and Emergent Behavior	10
2.4 Explanations for On In-Context Learning	11
2.4.1 In-Context Learning Relies on Knowledge Learned in Pre-Training	11
2.4.2 The Mechanisms Driving In-Context Learning	13

2.4.3	In-Context Learning as Implicit Gradient Descent	13
3	Method	15
3.1	Basic Setup of In-Context Learning	15
3.2	Task Recognition and Task Learning	16
3.2.1	Task Recognition	16
3.2.2	Task Learning	17
3.3	Disentangling Task Recognition and Task Learning via Label Space Manipulation	17
4	Experiments	19
4.1	Datasets	19
4.2	Models	20
4.3	Task Setup	21
4.4	Prompt Template Design	21
4.5	Prompt Construction	22
5	Results	24
5.1	Task Recognition Is a Broader Capability across Scales	25
5.2	Task Learning Is Enabled with Scale	25
5.3	Which Abstract Label Form is the Best?	27
5.4	Task Difficulty Affects the Trend	28
5.5	Effect of Number of Demonstrations on Model Performance	28
6	Conclusion	30
6.1	Summary and Discussion	30
6.2	Limitations	30
6.3	Future Work	31

A	Datasets	33
A.1	Datasets	33
A.2	Prompt Templates	34
B	Result Tables	38
B.1	Additional Results	38
B.1.1	Task-by-Task Analysis	38
B.1.2	Task-by-Task Analysis	38

List of Tables

3.1	Input examples for our three settings for sentiment classification tasks. Please refer to Table A.1, A.2, and A.3 for all prompts we use. . . .	17
A.1	Natural prompts used as input in GOLD and RANDOM settings for single-sentence datasets. $\langle s \rangle$ denotes the input sequence; labels are illustrated in red.	35
A.2	Natural prompts used as input in GOLD and RANDOM settings for multi-sentence datasets. $\langle s1 \rangle$ and $\langle s2 \rangle$ denote the input sequences; labels are illustrated in red.	36
A.3	Minimal prompts used for ABSTRACT.	37
B.1	Single dataset accuracies across the GPT-3 model family, using 8 ex- amples.	44
B.2	Single dataset accuracies across the GPT-3 model family, using 16 examples.	44
B.3	Single dataset accuracies across the GPT-3 model family, using 32 examples.	45
B.4	Single dataset accuracies across the OPT model family, using 8 exam- ples.	45
B.5	Single dataset accuracies across the OPT model family, using 16 ex- amples.	46

B.6	Single dataset accuracies across the OPT model family, using 32 ex- amples.	47
B.7	Single dataset accuracies across the LLaMA model family, using 8 ex- amples.	47
B.8	Single dataset accuracies across the LLaMA model family, using 16 examples.	48
B.9	Single dataset accuracies across the LLaMA model family, using 32 examples.	48

List of Figures

1.1	Averaged performance on 17 classification datasets for OPT. <i>Task recognition</i> (TR; the random label setting in blue) does not scale with model size and number of demonstrations, while <i>task learning</i> (TL; the abstract label setting in orange) does.	3
5.1	Averaged accuracy across 17 datasets for GPT-3 (top), LLaMA (middle), and OPT (bottom). Left graphs plot model sizes from small to large against performance, using 32 examples. Variance is calculated across three prompt templates. Right graphs plots number of demonstrations against performance for GPT3-Davinci and OPT-66B. . . .	24
5.2	Averaged accuracy across 17 datasets for GPT-3 (top), LLaMA (middle), and OPT (bottom). X-axis shows model sizes from small to large. We run experiments with 8 (left), 16 (middle), and 32 (right) demonstrations respectively. Variance is calculated across three prompts. . .	26
5.3	Performance of three types of ABSTRACT labels – numbers, letters, and symbols – across all model sizes. X-axis shows the number of demonstrations used.	27
5.4	Average results of sentiment analysis datasets vs NLI datasets on GPT-3 models.	28

5.5	Averaged task performance across 17 datasets for OPT (bottom), LLaMA (middle), and 17 for GPT-3 (top). x-axis shows number of demonstrations in the prompt. For each model, we run experiments for RANDOM (left), ABSTRACT(middle), and GOLD (right) demonstrations. Variance is calculated across three templates.	29
B.1	Averaged task performance across 17 datasets for OPT (bottom), LLaMA (middle), and 17 for GPT-3 (top). x-axis shows number of demonstrations in the prompt. For each model, we run experiments for RANDOM (left), ABSTRACT(middle), and GOLD (right) demonstrations. Variance is calculated across three templates.	39
B.2	Average performance of sentiment analysis datasets.	40
B.3	Average performance of natural language inference/paraphrase detection datasets.	41
B.4	Average performance of toxicity detection datasets.	42
B.5	Average performance of topic/stance classification datasets. . . .	43

Chapter 1

Introduction

1.1 The Rise of In-Context Learning

In-context learning, or the ability to "learn" to perform a task purely from examples in the context without parameter updates, is now one of the most infamous phenomena of large language models. This powerful and flexible phenomenon enables LLMs to be used as general-purpose models that can perform any task with a small set of labeled examples (Brown et al., 2020). Indeed, the in-context learning performance of LLMs may be competitive with fully supervised models, without the overhead of fine-tuning parameters or large training datasets. As language models scale, both in terms of model size and volume of pretraining corpora, the natural language processing (NLP) community's has become increasingly focused on efficient and intuitive methods for leveraging LLMs' zero-shot task generalization.

In-context learning boasts several key advantages. Firstly, it is relatively data-efficient; it merely requires a small number of demonstrations for each prompt, which may be reused for subsequent test queries. Secondly, it demands far less computational power than fine-tuning. Indeed, in many circumstances, updating model weights is computationally prohibitive, and squanders the LLM's strong generaliza-

tion abilities. In addition, with respect to the end-user, providing examples is an extremely intuitive way for humans to describe the behavior they would like to elicit from the model. It requires no technical expertise on the part of the user, outside of the ability to generate representative task examples, and is therefore easily used by layman users. Finally, in-context learning is a strong prompting strategy that can be employed without access to model weights, i.e. on closed-source models. Through careful demonstration selection and prompt design, in-context learning can greatly boost black-box model performance on downstream tasks. Today, in-context learning has become both a fundamental and critical language model capability.

It is easy to regard in-context learning for granted; it may seem obvious that providing examples to the model in the prompt will improve its ability to perform a downstream task. However, at the inception of the large language model era, the realization that language models could perform in-context learning was extremely surprising. After all, language models are *not* explicitly trained to learn from examples. That the simplest of language model training objectives – next token prediction – could provoke a sophisticated ability to perform pattern recognition of input-output pairs was shocking.

Moreover, it was generally understood that language models perform better when their prompts are in-distribution with the pre-training corpora. Yet the input-output schema used in ICL is out-of-distribution compared to the documents used in pre-training. It seemed counterintuitive that adding out-of-distribution text *help* the model’s performance, rather than *harming* it.

1.1.1 How Does In-Context Learning Work?

Today, there is still no strong consensus on how in-context learning works. (More details on relevant work are discussed in Section 2.4, but we provide a brief summary here.)

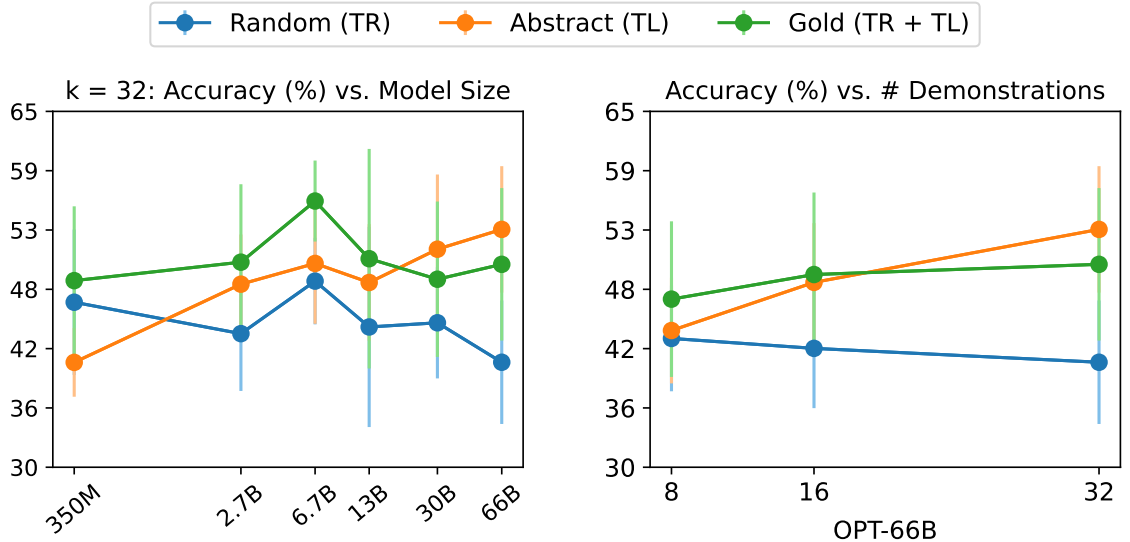


Figure 1.1: Averaged performance on 17 classification datasets for OPT. *Task recognition* (TR; the random label setting in blue) does not scale with model size and number of demonstrations, while *task learning* (TL; the abstract label setting in orange) does.

Some previous work hypothesizes that during pre-training, LLMs implicitly learn tasks required for downstream applications, and the in-context demonstrations merely provide information that allow the model to recognize which task is required (Xie et al., 2022). Min et al. (2022) show empirical evidence of this hypothesis by demonstrating that ICL performance is insensitive to the correctness of the labels¹.

On the other hand, Rong (2021) show that ICL is capable of operation overloading, e.g., overloading “+” to perform “subtract”. Akyürek et al. (2022); von Oswald et al. (2022a) construct theories demonstrating how Transformer-based models may perform implicit gradient descent to update an “inner-model”, and Dai et al. (2022) demonstrate similarities between in-context learning and explicit fine-tuning through a series of metrics on real-world datasets. Such hypotheses assume the correct input-output mappings are important and ICL actually performs implicit learning over demonstrations.

¹Yoo et al. (2022) provide a contradictory finding, showing that correct input-label correspondence can be critical as the datasets, model sizes, prompts, and metrics vary.

Both perspectives seem to explain different elements of in-context learning behavior. However, the question is now how we can compare these different driving forces of in-context learning in a unified framework.

1.2 Research Design

1.2.1 Key Questions

The key research questions we explore are as follows:

- Are there separate mechanisms for the ability to perform novel pattern recognition in-context, versus the ability to recognition of tasks (or other kinds of knowledge) seen in pre-training?
- If so, how can we disentangle these two components?
- Do these two mechanisms behave differently with respect to factors like model size or number of in-context demonstrations?

1.2.2 Research Design

In this paper, we propose to disentangle ICL into **task recognition** (TR), which recognizes the task from demonstrations and applies LLMs’ pre-trained priors, and **task learning** (TL), which learns a new input-label mapping from demonstrations. In the standard ICL scenario where ground-truth input-label pairs are provided, task recognition and task learning take effect simultaneously.

In addition to the standard GOLD setting, we propose two settings to tease them apart:

- RANDOM, where the labels are randomly sampled from the label distribution leaving only the disjoint input and label distributions for the LM to infer the task and apply task recognition (Min et al., 2022)

- ABSTRACT, where the labels are replaced with abstract symbols (e.g., numbers, letters, or other characters) that never co-occurred with the inputs in pre-training, forcing LLMs to apply task learning.

Additionally, we focus on how the two abilities in ICL evolve with two factors – different *model sizes* and *numbers of demonstrations*, which have been neglected in related literature².

We perform extensive experiments with a series of classification datasets on OPT (Zhang et al., 2022) and GPT-3 (Brown et al., 2020) models.

1.2.3 Key Findings

In summary, we find that:

- The gap between GOLD and RANDOM is small with smaller models, corroborating with Min et al. (2022). However, with larger models and more examples, the gap becomes larger.³ This suggests that task recognition plays a significant role in ICL, but it does not scale much with increasing parameters or examples.
- LLMs also perform task learning, which emerges quickly with larger models and more demonstrations – at the largest model ABSTRACT outperforms RANDOM, pointing to a paradigm shift in ICL at scale.

²Yoo et al. (2022) also have similar experiments to RANDOM and ABSTRACT, but those are not shown with changing model sizes and number of demonstrations.

³Most experiments in Min et al. (2022) are on ≤ 13 B models with 16 demonstrations.

Chapter 2

Related Work

2.1 Pre-trained Language Models

Pre-trained language models have become the backbone of NLP systems even before today’s large language model era. While training on a large corpora of textual data, language models learn universal language representations, which may then be leveraged to improve downstream task performance. This is beneficial for many reasons; we highlight two below.

First, downstream task datasets are relatively small (compared to the size of most pre-training corpora); if a language model with many parameters were to be trained from scratch on this data, it would likely overfit and generalize poorly. Thus, it is advantageous to first learn task-agnostic language representations and then fine-tune to adjust to a task-specific setting. In fact, the robust general-purpose features learned in pre-training may be so strong that only a relatively small amount of data is needed for fine-tuning. In this *few-shot* setting, it is often advantageous to adapt the downstream task to become more similar to the pre-training corpora distribution. More information on prompting and prompt engineering can be found in Section 2.2.

Second, sufficiently large language models trained on sufficient quantities of data

learn such strong representations that they may be able to perform tasks in the zero-shot setting, i.e. no fine-tuning required to perform a particular task. Thus, similarly to the few-shot fine-tuning setting, the key in this paradigm is to develop prompts that best enable the model to leverage knowledge learned in pre-training. Consequently, the idea of prepending the test query with input-output demonstrations as a means of describing a task by example led to the discovery of the in-context learning phenomenon. Brown et al. (2020) was the first to demonstrate the amazing in-context learning ability of LLMs; since then, many works have developed both better methods for in-context learning and different hypotheses about how in-context learning works.

2.1.1 Pre-trained Language Model Families

Here, we give a brief overview of the two model families we investigate in this paper.

GPT-3 Models

Brown et al. (2020) present the language model that arguably marked the inception of the large language model era: GPT-3 Davinci, a 175 billion parameter Transformer-based autoregressive language model. GPT-3 is trained on a filtered and postprocessed version of CommonCrawl and high-quality reference corpora using the standard language modeling objective. Despite its simple training objective and model architecture – the same as GPT-2, (Radford et al., 2019) – GPT-3 is capable of strong performance in the few-shot regime on a wide variety of tasks *without parameter updates or fine-tuning*. Specifically, (Brown et al., 2020) shows that GPT-3 demonstrates competitive performance with other state-of-the-art task-specific models on translation, closed-book question answering, and sentence completion, and more. In some cases, even its zero-shot performance only slightly underperforms their state-of-the-art counterparts.

OPT Models

Zhang et al. (2022) provide a suite of open-source models which were designed to be as closely comparable to the closed-source GPT-3 models as possible. The largest of these OPT models is also a 175 billion Transformer-based decoder-only model; they also provide eight other model sizes. Like Brown et al. (2020), OPT-175B demonstrates strong in-context learning ability, though its performance lags behind GPT-3 in the one- and few-shot regimes.

LLaMA Models

Touvron et al. (2023) introduces a suite of open-source models. The largest of these OPT models is 65 billion Transformer-based decoder-only model; the other three sizes are 7 billion, 13 billion, and 30 billion. Touvron et al. (2023) demonstrates that a smaller LLaMA model (13B) can outperform GPT-3 (175B), whereas its largest (65B) is competitive with state of the art models.

2.2 Prompting

Prompting is recognized as an important tool for improving model performance. By designing a prompt template that molds the downstream task to a format that is more in-distribution with the text corpora seen in pre-training, model performance could be improved both in the fine-tuning and in-context learning setting. Indeed, models were found to be highly sensitive to the choice and format of the prompt. Zhao et al. (2021) also found that the choice of prompt template was also critical to determining model performance. They design fifteen different templates for a single sentiment classification task and test each template’s performance on small sets of test queries. The templates’ distributions were highly variable with respect to one another, indicating that the model’s performance was sensitive to the choice of prompt

template.

In addition to the formatting of the input or output, models were also found to be highly sensitive to both the choice and order of examples. Zhao et al. (2021) show that varying the permutation and selection of examples leads to highly variable model performance, regardless of the overall number of examples. Lu et al. (2021) shows that this sensitivity persists even to larger models and more examples. They feed prompts constructed from different permutations of four examples and show that even GPT-3 Davinci remains sensitive to the example order on certain datasets. Moreover, permutations that work well for one model did not work for another. Thus scale alone may not reduce prompt and example order sensitivity to an acceptable degree.

Given the high variance attributable to prompt selection, many methods for improving prompts, have been explored over the recent years. Some focus on better template design. For instance, Mishra et al. (2021) introduces a *prompt reframing* technique, which manually decomposes task instructions into sequences of simpler tasks. They find that the reframed instructions lead to stronger performance across both small and large models and also reduce the number of examples needed for strong few-shot performance. Shin et al. (2020) develops an automated prompt-creation pipeline which uses a gradient-guided search over a "trigger token" space. Prompts are constructed by appending the set of optimal trigger tokens to the original test query. This led to stronger performance on cloze-style tasks for both BERT and RoBERTa (Devlin et al., 2019; Liu et al., 2019). Gao et al. (2020) presents LM-BFF, a suite of techniques for improving prompting in the few-shot fine-tuning setting. This suite includes an automated prompt generation pipeline which includes a search over manually designed templates and the space of label words.

With respect to better selection or ordering of in-context examples, Lu et al. (2021) demonstrates two metrics – Global Entropy and Local Entropy – to determine

which prompts may lead to unbalanced or overly confident predictions, respectively. Rubin et al. (2022) demonstrate a retrieval method to select in-context examples. They train a dense retriever to select the best candidate training example given a particular test query. The performance outperforms random example selection by a significant margin on a number of semantic parsing datasets.

2.3 Scaling Models and Emergent Behavior

The role of model scale in large language models was made obvious in Kaplan et al. (2020), which found that language modeling performance scaled as a power-law with model size. In particular, they noted that performance did *not* correlate strongly with model depth or width; it was primarily the *number of parameters* that demonstrated a strong trend with language modeling performance. These findings were corroborated by similar work demonstrating a similar relationship between model scale and performance in translation (Gordon et al., 2021) and autoregressive generation (Henighan et al., 2020).

However, Wei et al. (2022) later found that scaling models up not only improved downstream task performance, but also enabled sophisticated model behavior, such as in-context learning. These abilities were termed *emergent abilities*, as they emerged only after some sufficient threshold of model size had been met. That is to say, rather than scaling smoothly with model size, the presence of emergent abilities is *only* present in large-enough language models, but is *not* exhibited by any smaller models. On a variety of tasks, Wei et al. (2022) showed that in-context learning performance was only able to exceed random performance after a certain threshold of model size, though this threshold varied by both model family and task type. Srivastava et al. (2022) tested a wide range of model families on over two hundred tasks, finding that emergent abilities were especially noticeable for tasks involving multi-step reasoning

or compositions of multiple tasks.

2.4 Explanations for On In-Context Learning

The increasing academic focus on in-context learning yielded many proposed explanations for its emergence in large language models. In this section, we give an overview of some of the prevailing arguments.

2.4.1 In-Context Learning Relies on Knowledge Learned in Pre-Training

Xie et al. (2022) present a Bayesian perspective on in-context learning. They model the documents in the pre-training corpora as the emissions of a Hidden Markov Model, which is parameterized by θ . This θ may be thought of as a latent concept, such as a specific type of document, e.g. Wikipedia biographies. In pre-training, language models must learn to predict the next token of a document correctly. Since the document is parametrized by θ , the model is implicitly incentivized to identify the correct θ , since doing so improves its prediction of the next token. At inference time, the model leverages the in-context examples to identify the latent concept shared by the demonstrations.

More explicitly, while performing in-context learning, given output y and context c :

$$p(y|c) = \int_{\theta} p(y|\theta, c)p(\theta|c)d\theta \quad (2.1)$$

Thus in-context learning boils down to implicit Bayesian inference.

The key insight to note about this finding is that it relies on the supposition that all concepts have been learned in pre-training, and that the in-context examples are used merely to identify the particular underlying concept.

Min et al. (2022) provided strong empirical evidence for this assumption by showing that *ground-truth input-label mappings were not necessary for strong downstream task performance*. In particular, they show that randomizing the labels in the in-context demonstrations lowers task performance by only a small number of points. Thus, the primary source of information being leveraged by the model must not be the ground-truth input-label mappings. Min et al. (2022) provide a series of ablations, finding that the key components of a successful prompt includes (1) full demonstration of the label space, (2) examples of the input distribution, and (3) examples of the format demanded by the prompt template. In effect, the examples only need to provide enough information for the model to simply *identify* the task. This evidence synergizes easily with the intuition proposed by Xie et al. (2022). If the in-context examples allow the language model to identify the latent task or concept at hand, then the model can rely on its knowledge of that concept in order to produce the right answer, even though incorrect input-label mappings were provided.

Chan et al. (2022) investigate various qualities of the pre-training distribution that may give way to in-context learning behavior. They test a combined embedder and Resnet on an image classification dataset with image-label pairs. In particular, they show that several characteristics of the dataset lead to the emergence of in-context learning. For instance, these qualities include “burstiness”, or the nonuniform clustering of classes; multiple labels for the same class; and a long-tailed class distribution. While their empirical results do not include a direct investigation of language modeling, they do involve discussion on the large language models in-context learning capabilities. In particular, they emphasize that the distribution of natural language is Zipfian and tends to be temporally clustered (since discussion on a particular entity or topic tends to be concentrated rather than dispersed uniformly). Likewise, synonyms for the same concept – e.g. “positive”, “great”, “good” – fulfill the multiplicity of labels criterion. Thus, there is evidence to show that the distribution of natural

language lends itself to language modeling.

2.4.2 The Mechanisms Driving In-Context Learning

Olsson et al. (2022) present a mechanistic perspective on the subcomponents that drive pattern recognition and therefore in-context learning. They show that, in two-layer transformers, two heads in different layers collaborate in a subnetwork called *induction head*. They show empirically that these induction heads are responsible for the emergence of in-context learning for simple patterns, such as copying the input. Specifically, in this copying phenomenon, the tokens that follow the current token in the context attended to (“prefix matching”), and their corresponding logits are increased (“copying”). They extend their experiments to other pattern recognition tasks. These include novel mappings that are highly unlikely to have been prevalent in pretraining, such as mappings of random category pairs to numerical labels. While these experiments were performed in smaller neural nets with fewer layers, they are indicative of structural components that explicitly learn novel patterns in the context. This seems to contradict the assumption that all such patterns or mappings have been seen in pre-training.

Bansal et al. (2022) extend Olsson et al. (2022)’s findings to larger models, finding that specific subnetworks in OPT-66B are responsible for performing prefix matching and copying. They also show that the majority of attention heads can be removed without significant degradation of task performance, which may show that only a small proportion of parameters are actually responsible for in-context learning.

2.4.3 In-Context Learning as Implicit Gradient Descent

Dai et al. (2022) show that, under some theoretical assumptions, Transformer attention can be shown to be equivalent to gradient-based descent. They posit that a forward pass of the demonstrations through the model produces meta-gradients which

”updates” the original language model weights to build an implicit in-context learning model. They draw parallels between this process and the standard fine-tuning process where training examples are used to produce gradients via backpropagation, which is then used to directly update the model weights. They construct metrics of similarities to show that at the prediction, representation, and attention behavior levels, a fine-tuned model (simplified according to the theoretical assumptions made above) behaves similarly to the in-context learning performance.

von Oswald et al. (2022b) also theorize that the in-context learning is equivalent to gradient-based optimization. They provide a specific weight construction for a linear self-attention layer which induces an identical update to a single step of gradient descent (for regression loss). Moreover, they show that if Transformers transformers are further trained on linear regression datasets, the weights converge to the same weight construction as above. This implies that Transformers are implicitly implementing gradient descent with each forward pass of in-context examples.

Moreover, von Oswald et al. (2022b) provide empirical evidence for the existence of the induction head networks identified in Olsson et al. (2022). In particular, they train a two-layer Transformer model and measure its loss. At the point where the model’s performance jumps to that of a single step of gradient descent, the first layer suddenly becomes sensitive to the next token. This corroborates with the ”prefix-matching” and ”copying” process employed by the induction heads.

Like von Oswald et al. (2022b) and Dai et al. (2022), Akyürek et al. (2022) posit that Transformer-based in-context learning models implicitly perform learning algorithms like gradient descent and ridge regression. However, they perform their analysis on Transformer models explicitly trained on an in-context learning objective.

Chapter 3

Method

3.1 Basic Setup of In-Context Learning

A large language model (parameterized by θ) performs ICL by conditioning on the input-label pair demonstrations $D_{\text{demo}} = (x_1, y_1, x_2, y_2, \dots, x_K, y_K)$ and the test input x_{test} to predict the label $y_{\text{test}} \sim p_{\theta}(y \mid x_{\text{test}}, D_{\text{demo}})$, where the demonstrations elicit a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}, x \in \mathcal{X}, y \in \mathcal{Y}$.

The prompt is generated by passing the demonstrations through some template $\mathcal{T}(x, y)$ and concatenating the templated demonstrations and the test input x_{test} . The context passed into the model is therefore:

$$\mathcal{C} = \mathcal{T}(x_1, y_1) \ \mathcal{T}(x_2, y_2) \ \dots \ \mathcal{T}(x_K, y_K) \ x_{\text{test}},$$

Given test query x_{test} , the model $\mathcal{M}$ then produces the label prediction:

$$\hat{y} = y_{\text{argmax}_j} Pr_{\mathcal{M}}(y_j \mid \mathcal{C}, x_{\text{test}})$$

3.2 Task Recognition and Task Learning

We delineate two ways through which an LLM can leverage in-context demonstrations: *task recognition* and *task learning*. We posit that the two mechanisms occur under separate conditions, as recognizing an already learned task is easier than learning a new mapping.

3.2.1 Task Recognition

Previous works suggest that LLMs infer the mapping shown in the demonstrations by extracting the latent concept learned during pretraining (Xie et al., 2022). Accordingly, Min et al. (2022) show that that gold input-label associations in demonstrations are irrelevant (Min et al., 2022).

Task recognition (TR) represents a language model’s ability to recognize the mapping f purely by observing the input distribution $\{x_i\}_{i=1}^K$ and the label distribution $\{y_i\}_{i=1}^K$, without the provided (x_i, y_i) pairs. The LLM then applies its pre-trained priors to the recognized f .

More formally, when only task recognition is enabled,

$$p_{\theta}(y \mid x_{\text{test}}, \{x_i, y_i\}_{i=1}^K) = p_{\theta}(y \mid x_{\text{test}}, \{x_i\}_{i=1}^K, \{y_i\}_{i=1}^K),$$

which suggests task recognition does not rely on the pair information. Inferring the latent mapping f may be easier when the input and label distributions have a higher frequency in the pre-training corpus (more “natural” prompts). For example, an input distribution of movie reviews and a label distribution of “The sentiment is positive/negative” are easily recognized as a sentiment classification task due to their prevalence during pre-training, and LLMs can make reasonable predictions without explicitly “learning” the task via ground-truth demonstrations. This leads to observations that the model can still perform well, even if we provide wrong input-label

Setting	Example
GOLD	I like the movie.\n Sentiment: <i>positive</i>
RANDOM	I like the movie.\n Sentiment: <i>negative</i>
ABSTRACT	I like the movie.\n Label: 1

Table 3.1: Input examples for our three settings for sentiment classification tasks. Please refer to Table A.1, A.2, and A.3 for all prompts we use.

mappings, e.g., “The movie is great. The sentiment is *negative*” (Min et al., 2022).

3.2.2 Task Learning

Task learning (TL), on the other hand, characterizes how the model learns a new mapping from the input-label pairs through demonstrations. Unlike task recognition, task learning allows models to learn novel mappings and thus correct input-label pairs will be crucial. Thus,

$$p_{\theta}(y \mid x_{\text{test}}, \{x_i, y_i\}_{i=1}^K) \neq p_{\theta}(y \mid x_{\text{test}}, \{x_i\}_{i=1}^K, \{y_i\}_{i=1}^K),$$

We hypothesize that models are able to perform task recognition at a small scale, but this ability does not drastically improve with increasing model sizes and annotations; on the other hand, task learning improves significantly when model sizes and numbers of demonstrations increase.

3.3 Disentangling Task Recognition and Task Learning via Label Space Manipulation

We required a task-agnostic method that would allow us to disentangle task recognition and task learning. In particular, for task learning, we require that the mapping demonstrated in the context be highly unlikely to have been seen in the pre-training phase. However, task recognition demands that we choose a task that the language

model *has* seen in pre-training. In order to make a fair comparison between a model’s ability to perform task learning vs. task recognition, we apply *label space manipulation* of downstream classification tasks, disentangle task recognition and task learning through using three different setups. (Examples are shown in Table 3.1).

To test **task recognition**, we provide the minimum known information needed for the model to recognize the task. According to Min et al. (2022), models can leverage task priors when the demonstrations only provide the input distribution, the output space, and the sequence format. Thus, in this setting, we randomize the labels while preserving the input text and maintaining a consistent sequence format. We refer to these experiments as RANDOM LABEL.

To test **task learning**, the context must demonstrate an input-output mapping that is highly unlikely to have been seen in pre-training. In other words, we replacing the original natural language labels with new labels which are selected so that the models are unlikely to have been exposed to these input-label correspondences in pretraining. We replace labels with randomly selected numbers, letters, and abstract symbols ¹ – NUMBER LABEL, LETTER LABEL, and ABSTRACT LABEL respectively. Specifically, for *each* set of demonstrations, we randomly map the natural language labels to a randomly sampled set of new labels.

Finally, the GOLD setting uses standard demonstrations with gold input-label pairs.

We conduct experiments with the above three different settings under two axes – model sizes and numbers of demonstrations – and show how task recognition and task learning manifest under different conditions. Section 4.1 presents additional details on the experimental setup.

¹We choose ASCII symbols such as "*", "&", "#"

Chapter 4

Experiments

4.1 Datasets

We experiment on 17 classification datasets across four types of tasks: sentiment analysis, toxicity detection, natural language inference/paraphrase detection, and topic/stance classification.

For **sentiment classification**, we use the following datasets:

- SST-2 (Socher et al., 2013)
- financial phrasebank (Malo et al., 2014)
- emotion (Saravia et al., 2018)
- poem sentiment (Sheng and Uthus, 2020)

For **hate speech detection**, we use the following datasets:

- tweet eval hate (Basile et al., 2019)
- ethos race (Tsoumakas et al., 2020)
- ethos gender (Tsoumakas et al., 2020)

- ethos national origin (Tsoumakas et al., 2020)
- ethos religion (Tsoumakas et al., 2020)

For **natural language inference** and **paraphrase detection**, we use the following datasets:

- SICK (Marelli et al., 2014)
- SNLI (Bowman et al., 2015)
- WNLI (Levesque et al., 2012)
- MRPC (Dolan and Brockett, 2005)

Finally, for **topic classification**, we use the following datasets:

- TREC (Voorhees and Tice, 2000)
- tweet eval atheist (Mohammad et al., 2018)
- tweet eval feminist (Mohammad et al., 2018)

We provide further details in Section 4.3 and §A.1.

Our dataset selection largely follows Min et al. (2022), but we exclude multi-choice datasets since it is difficult to apply our ABSTRACT experiments on them.

4.2 Models

We use three state-of-the-art LLM families: OPT (Zhang et al., 2022), LLaMA (Touvron et al., 2023), and GPT-3 (Brown et al., 2020).

For OPT, we use checkpoints from the Transformers library (Wolf et al., 2020), with model sizes of 350M, 2.7B, 6.7B, 13B, 30B, and 66B parameters.

For LLaMA, we use model sizes of 7B, 13B, 30B, and 66B parameters.

We use GPT-3 `ada` (350M), `babbage` (1.3B), `curie` (6.7B), and `davinci` (175B) via the OpenAI API. For fair comparison to the OPT model suite, we use the non-instruction legacy models, which were not fine-tuned on human data or instructions.

4.3 Task Setup

In order to disentangle task recognition from task learning, we design two settings: one where the model must rely purely on task priors to perform the task and one where the task can only be learned via gold input-label mappings in the demonstrations. Refer to Section 3.3 for additional details about how we employ label space manipulation for these experiments.

We adopt the sample-based evaluation protocol; that is to say, each sampled test example has varying numbers of sampled demonstrations (8, 16, or 32) from the training set. Each prompt is sampled independently for each test example. For evaluation, we use test examples drawn from the validation sets of each dataset. We manually design 3 prompt templates for each type of classification tasks (details in 4.5 and §A.2) and report the mean of the templates for each datapoint. For GPT-3, we sample 150 examples for each dataset. We report the mean accuracy by averaging across dataset types and prompts, and variance across different prompts. For OPT and LLaMA, we sample 1350 examples for all datasets.

4.4 Prompt Template Design

We design two kinds of prompts: *natural language prompts* (Table A.1), which are similar to the manual prompts in Min et al. (2022), and *minimal prompts* (Table A.3), which remove any natural language instructions for the task.

For natural language prompt templates, we design task-specific templates, following the examples set in Min et al. (2022). The goal is to develop prompts that remain

in-distribution with the natural language labels; thus, we adopt the natural prompts for GOLD and RANDOM.

The minimal prompt templates are designed for the ABSTRACT setting so that there is no task-specific information leakage in the prompt template itself, as it is key that the model avoids reliance on pre-training priors. Moreover, these templates use less natural language and simpler templates, since the abstract symbols are out-of-distribution with the natural language prompts.

A list of all natural language and minimal prompts may be found in Tables A.1 and A.3, respectively.

4.5 Prompt Construction

To construct the GOLD prompts, we adopt the standard ICL setting where we use natural prompts and gold input-label pairs. This setup reflects both task recognition and task learning abilities.

To construct the RANDOM prompts, we adopt the approach taken by Min et al. (2022). We use the same natural prompts as GOLD and randomly sample labels for demonstrations. This setup reflects the task recognition performance only.

To construct the ABSTRACT prompts, we use minimal prompts (which provide no task information) and characters with no clear semantic meanings as the label for each class. We test three types of label choices. Given a classification task with N classes, the labels are selected from:

- *Numbers*: $0, \dots, N - 1$
- *Letters*: N letters from A, B, C, ...
- *Symbols*: first N symbols of “@”, “#”, “\$”, “%”, “*”, and “^”

Additionally, we randomly sample new input-label mappings for each prompt in

order to mitigate any potential effect of priors on the abstract labels from pretraining. The sampled mapping is highly unlikely to have been seen in pre-training; thus, it reflects the task learning ability.

We report the *number* abstract labels in the main results, but compare the three results of using each type of abstract label in Section 5.3.

Chapter 5

Results

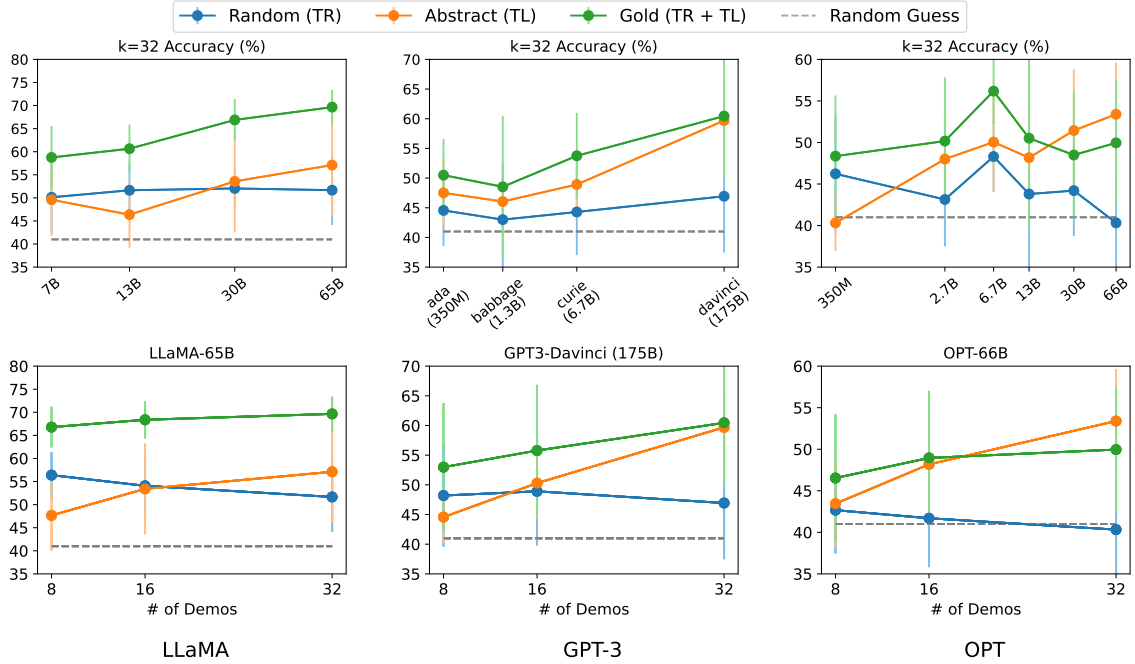


Figure 5.1: Averaged accuracy across 17 datasets for GPT-3 (top), LLaMA (middle), and OPT (bottom). Left graphs plot model sizes from small to large against performance, using 32 examples. Variance is calculated across three prompt templates. Right graphs plots number of demonstrations against performance for GPT3-Davinci and OPT-66B.

Figure 5.2 shows our main results with the three model families (LLaMA, GPT-3, and OPT) and our three settings: GOLD, RANDOM, and ABSTRACT. We first

verify that GOLD, the setting with all the information, performs the best; with *small* model sizes and *few* examples ($k = 8$), there is insignificant gap between RANDOM and GOLD (Min et al., 2022) and ABSTRACT significantly lags behind. Below we summarize the trends of task recognition and task learning across different conditions.

5.1 Task Recognition Is a Broader Capability across Scales

As shown in Figure 5.2, for all three model families, the RANDOM setting show similar performance at all sizes of models and numbers of demonstrations; even the smallest 350M parameter models are able to recognize the task, drawing around 10 points of average performance lead against the random baseline for GPT-3 and 5 points with OPT, using just 8 examples. For LLaMA, the smallest model (7B) performs about 10 points higher than the largest model (65B).

This pattern persists even as more examples are added. With 32 examples, RANDOM does not improve significantly for smaller or larger models. Indeed, some models even experience a downtick in performance; it is possible that the model has picked up on the random pattern or has tried to infer a pattern within the randomness of the labels, thus leading it to output the incorrect answer. This shows that task recognition from in-context examples does not drastically scale with model size or numbers of examples.

5.2 Task Learning Is Enabled with Scale

We observe that task learning is dependent on model sizes: smaller models perform roughly the same across all numbers of demonstrations, suggesting that they struggle to learn the mapping even when many examples are presented. On the other hand,

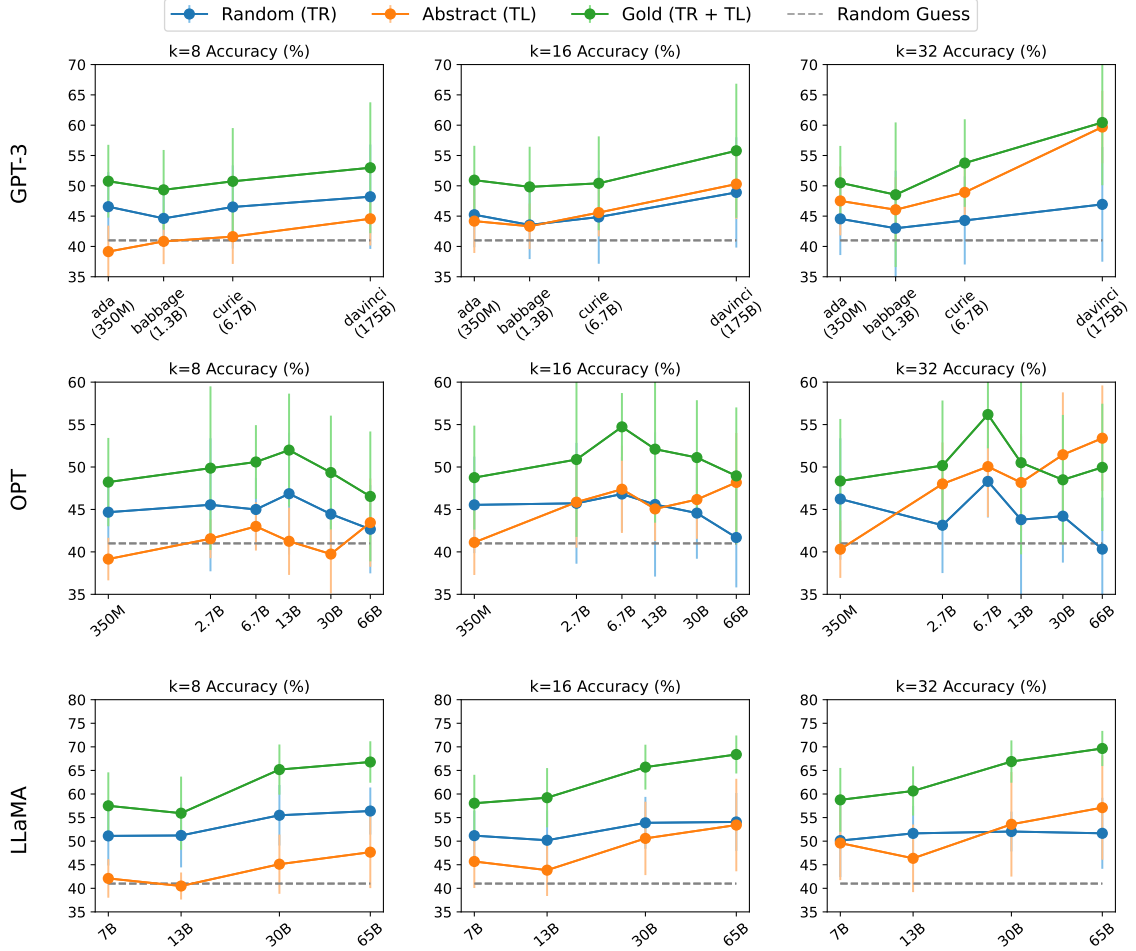


Figure 5.2: Averaged accuracy across 17 datasets for GPT-3 (top), LLaMA (middle), and OPT (bottom). X-axis shows model sizes from small to large. We run experiments with 8 (left), 16 (middle), and 32 (right) demonstrations respectively. Variance is calculated across three prompts.

larger models can utilize the provided mapping information and perform task learning, the performance improves as the number of demonstration increases.

This trend is clearly reflected across plots in Figure 5.1 – with larger K , the slope of ABSTRACT across sizes becomes steeper as larger models have a larger advantage of applying task learning. With only 16 examples, the largest models for GPT-3 and OPT are able to match the performance of GOLD while using a new label mapping. Indeed, with 32 examples, OPT-175B’s ABSTRACT performance even outperform GOLD. While LLaMA-65B’s ABSTRACT performs worse than GOLD, it is clear in

Figure 5.2 that increasing model size is critical to the ability to perform in the ABSTRACT setting. On the other hand, smaller models perform roughly the same across all numbers of prompt examples, suggesting that they struggle to learn the mapping even when many examples are presented in-context.

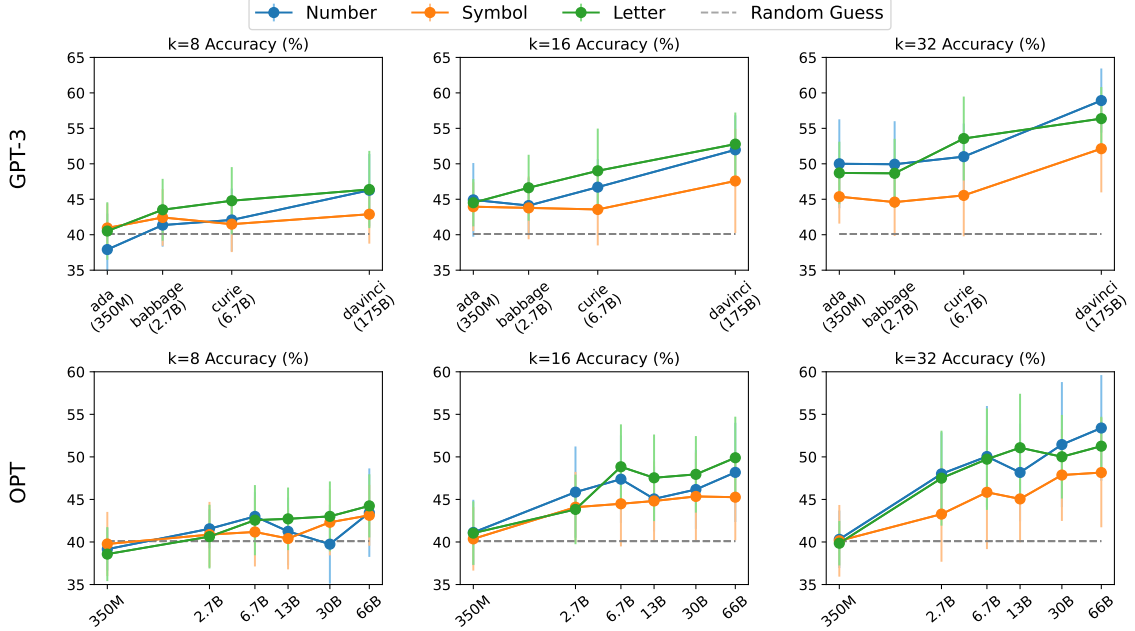


Figure 5.3: Performance of three types of ABSTRACT labels – numbers, letters, and symbols – across all model sizes. X-axis shows the number of demonstrations used.

5.3 Which Abstract Label Form is the Best?

We show in Figure 5.3 the separate ABSTRACT results with the number, letter, and symbol labels. We see that all three versions show a similar trend and coincide with our main results. Numbers and letters perform consistently better than symbols. This highlights the importance of the naturalness of the label space: even for larger models, the more natural the label, the easier it is to perform task learning.

Regardless, the trend is clear; for all types of symbols, larger models succeed at learning the mapping, whereas small models fail to do so even with 32 examples.

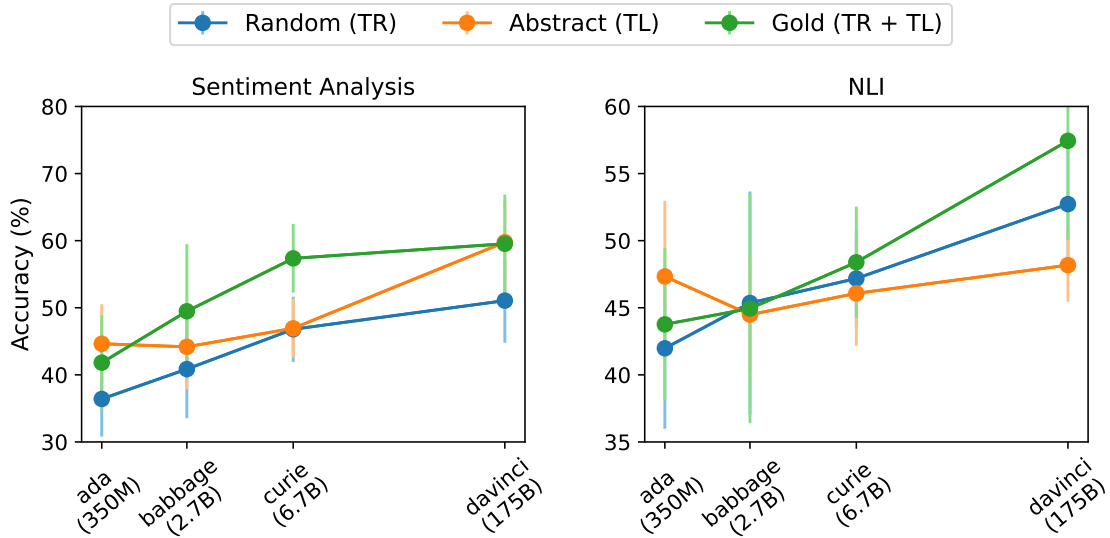


Figure 5.4: Average results of sentiment analysis datasets vs NLI datasets on GPT-3 models.

5.4 Task Difficulty Affects the Trend

We notice that ABSTRACT scales better with sizes and examples when the task is simpler. In Figure 5.4 we compare two types of tasks: sentiment analysis and natural language inference. Sentiment analysis is a simpler task; we observe that the ABSTRACT curve is very sharp, and matches the GOLD performance with the largest model. Since natural language inference is more difficult, we observe that it produces a flatter ABSTRACT curve, suggesting that the model relies more on the natural prompts and pre-training priors. We demonstrate the full task-type breakdown results in §5.5.

5.5 Effect of Number of Demonstrations on Model Performance

We demonstrate average model performance with respect to numbers of demonstrations in Figure B.1. We can see a clear trend that RANDOM (task recognition) does

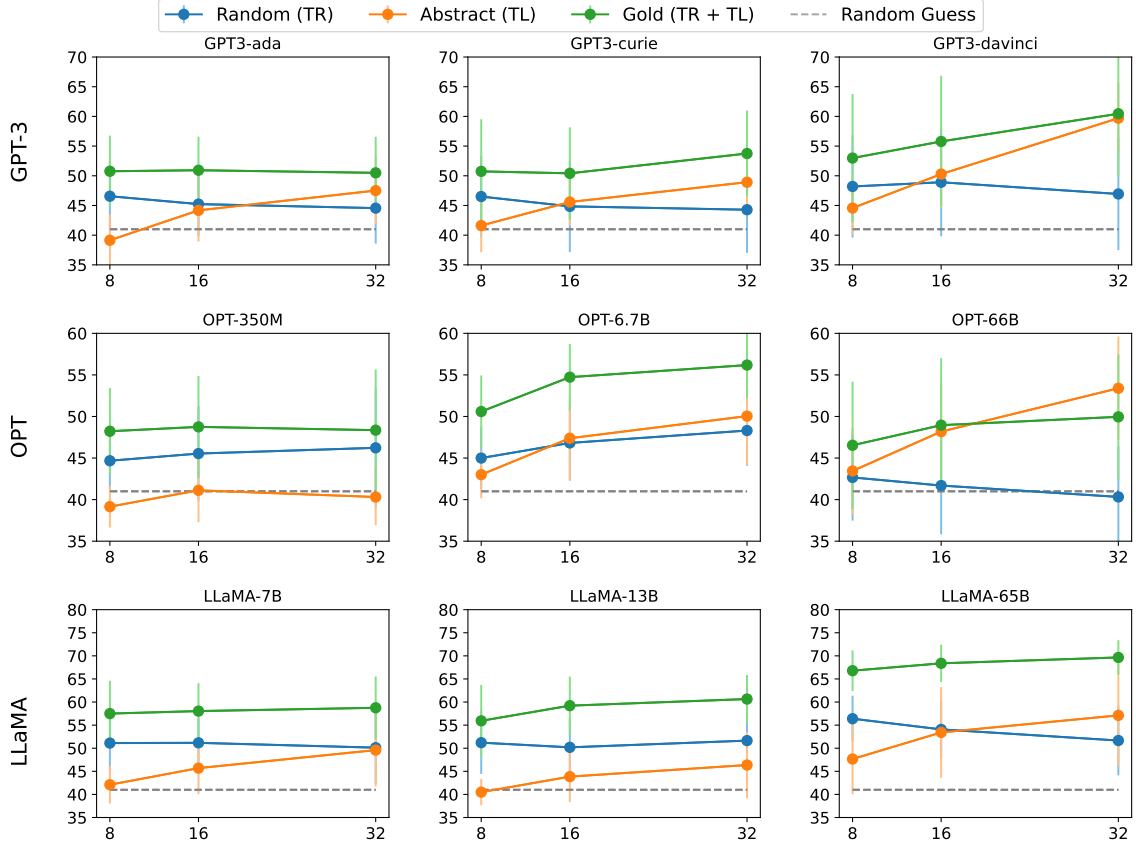


Figure 5.5: Averaged task performance across 17 datasets for OPT (bottom), LLaMA (middle), and 17 for GPT-3 (top). x-axis shows number of demonstrations in the prompt. For each model, we run experiments for RANDOM (left), ABSTRACT (middle), and GOLD (right) demonstrations. Variance is calculated across three templates.

not change much but ABSTRACT improves drastically with more examples, especially for GPT-3 Davinci and OPT-66B. This suggests that larger models actually leverage the examples to learn the function; since the performance improves with more examples, it is clear that the larger models are actually using the additional examples to better learn the mapping demonstrated in the context. Meanwhile, smaller models fail to produce any performance in task learning even with 32 examples, and there is very little improvement in task learning performance as the number of demonstrations increase.

Chapter 6

Conclusion

6.1 Summary and Discussion

While previous work usually studies ICL as an umbrella term, regardless of model size and numbers of examples, we argue that there are two distinct characterizations of ICL – task recognition and task learning. We further demonstrate that they emerge under different conditions. Even small models are capable of performing task recognition, but this ability does not scale; on the other hand, task learning is an emergent ability of large models. We suggest that future work on ICL should distinguish the two phenomena and clearly state the conditions under which the experiments are conducted.

6.2 Limitations

Though LLMs with in-context learning are capable of all kinds of NLP tasks, this work is limited to simple classification tasks because they are easier to be adapted to our RANDOM and ABSTRACT setup. We leave an extension to other types of NLP tasks as future work.

Another limitation of our work lies in the definition and discussion of task learning.

Though we empirically show that large models are capable of acquiring a novel mapping to abstract labels like numbers or letters, how models “learn” mechanistically is still elusive. As suggested in previous work, LLMs may conduct implicit gradient descent over demonstrations, or they may alternatively map the patterns shown in the demonstrations back to concepts learned in pre-training. To some extent, these mechanisms could be considered an advanced form of “task recognition”. This work only designs experiments to better observe and disentangle task recognition and task learning, and we look forward to further studies that reveal more insights about the mechanistic inner-workings of these phenomena in ICL.

6.3 Future Work

These findings could be extended to more model families and more tasks. In addition, more label settings could be explored. It would be interesting to investigate the ability to override label priors. For instance, one could map the original label space to a permutation such that each class is defined by a label whose natural language definition contradicts the class itself (e.g. “positive” to “negative” and vice versa). Thus, in larger models, a stronger ability to learn this novel mapping may compete with stronger label priors; it would be interesting to see the effects of scale and number of demonstrations in this setting.

Many of the potential avenues for future work revolve around better specification of the research question. In particular, it is difficult to define exactly what counts as “task recognition” or “task learning” (as mentioned in Section 6.2), partly because it is difficult to define what counts as a *task*. For instance, while the mapping generated by the ABSTRACT setting is itself unlikely to have been seen in pre-training, it could be argued that the act of identifying such an abstract mapping is itself a task that may have been seen in pre-training. Additionally, it could be argued that some latent

concept of the task needs to be employed for the model to be able to perform in the ABSTRACT setting. After all, even small models *does* have the ability to recognize, say, sentiment analysis; yet it seems that this secondary step of inferring a mapping between latent concepts of "positive" or "negative" to some abstract symbols is where the benefits of model scale come into play. It would be interesting to formalize a definition of these latent task concepts (similar to the document concepts introduced by Xie et al. (2022)) and explain how they are learned in pre-training. It would also be interesting to explore if large models' ability to perform "meta-tasks" (e.g. the task of learning a novel mapping) are enabled in pre-training, and if so, why smaller models are unable to develop the same abilities.

Appendix A

Datasets

A.1 Datasets

We use a total of 17 datasets. **Sentiment analysis** includes SST-2 Socher et al. (2013), financial_phrasebank Malo et al. (2014), emotion Saravia et al. (2018), and poem_sentiment Sheng and Uthus (2020) **Topic and stance classification** includes TREC Voorhees and Tice (2000), tweet_eval_atheist, and tweet_eval_feminist Mohammad et al. (2018); Basile et al. (2019). **Toxicity detection** includes tweet_eval_hate, ethos_race, ethos_gender, ethos_national_origin, and ethos_religion Tsoumakas et al. (2020) **Natural language inference/paraphrase detection** includes SICK Marelli et al. (2014), SNLI Bowman et al. (2015), WNLI Levesque et al. (2012), and MRPC Dolan and Brockett (2005).

We sample from the training set to construct the prompts; following Min et al. (2022), we use the development set for evaluation, using sampled $\max(1350, \text{dataset_size})$ examples.

A.2 Prompt Templates

For each task category (e.g. sentiment classification, topic detection), we manually design three natural language templates. Depending on exact specifications for the dataset, templates may be adjusted to better reflect the task (e.g. "Is this atheist?" for `tweet_eval_atheist`). We apply these templates to the natural language label sets (GOLD and RANDOM). All prompts are presented in Table A.1.

We also design two task-agnostic variations on three minimal templates for ABSTRACT: one for single-sentence tasks and one for multi-sequence tasks (e.g. NLI tasks). We use these minimal templates on the abstract language label sets in order to prevent the model from being exposed to any information regarding the task from the prompt design. All minimal templates are presented in Table A.3

All prompts are designed to be answered with single-token responses (e.g. "Yes/No", "True/False", "positive/negative/neutral", "0/1/2", "A/B/C") so that we can directly check models' last token prediction results instead of applying decoding methods.

Type	Template #	Example
Sentiment Analysis	1	<s> The sentiment is <positive/negative>
	2	<s> Sentiment: <positive/negative>
	3	<s> The sentiment of the text is <positive/negative>
Hate Speech	1	<s> Is this hate speech? <Yes/No>
	2	<s> Is the sentence hateful? <Yes/No>
	3	<s> The sentence contains hate speech. True or False? The answer is <True/False>
Stance Detection	1	<s> The stance is feminist. True or False? The answer is <True/False>
	2	<s> Does the sentence express a feminist view? <Yes/No>
	3	<s> Is the stance feminist? <Yes/No>
Topic Detection	1	<s> The topic is <label>
	2	<s> The sentence is about <label>
	3	<s> Sentence topic: <label>

Table A.1: Natural prompts used as input in GOLD and RANDOM settings for single-sentence datasets. <s> denotes the input sequence; labels are illustrated in red.

Type	Temp. #	Example
Entailment	1	$\langle s1 \rangle$ The question is: $\langle s2 \rangle$? True or False? The answer is $\langle \text{True/False} \rangle$
	2	Hypothesis: $\langle s1 \rangle$ Premise: $\langle s2 \rangle$? Do the sentences show entailment? $\langle \text{Yes/No} \rangle$
	3	The hypothesis is: $\langle s1 \rangle$ The premise is: $\langle s2 \rangle$? Is this entailment? $\langle \text{Yes/No} \rangle$
NLI	1	$\langle s1 \rangle$ The question is: $\langle s2 \rangle$ True, False, or Unknown? The answer is $\langle \text{True/False/Unknown} \rangle$
	2	Hypothesis: $\langle s1 \rangle$ Premise: $\langle s2 \rangle$? Given the premise, is the hypothesis true? Yes, No, or Unknown? The answer is: $\langle \text{Yes/No/Unknown} \rangle$
	3	The hypothesis is: $\langle s1 \rangle$ The premise is: $\langle s2 \rangle$? According to the premise, the hypothesis is true. True, False, or Unknown? The answer is: $\langle \text{True/False/Unknown} \rangle$
Paraphrase Detection	1	$\langle s1 \rangle$ The question is: $\langle s2 \rangle$ True or False? The answer is: $\langle \text{True/False}/\rangle$
	2	Sentence 1: $\langle s1 \rangle$ Sentence 2: $\langle s2 \rangle$ These sentences are paraphrases. True or False? The answer is: $\langle \text{True/False}/\rangle$
	3	Text: $\langle s1 \rangle$ Consider this sentence: $\langle s2 \rangle$ Does it paraphrase the text? $\langle \text{Yes/No} \rangle$

Table A.2: Natural prompts used as input in GOLD and RANDOM settings for multi-sentence datasets. $\langle s1 \rangle$ and $\langle s2 \rangle$ denote the input sequences; labels are illustrated in red.

Type	Template #	Example
Minimal (single sentence)	1	<i><sentence></i> <i><label></i>
	2	<i><sentence></i> Label: <i><label></i>
	3	Sentence: <i><sentence></i> Label: <i><label></i>
Minimal (multiple sentences)	1	<i><sentence1></i> [SEP] <i><sentence2></i> <i><label></i>
	2	<i><sentence1></i> [SEP] <i><sentence2></i> Label: <i><label></i>
	3	Sentence 1: <i><sentence1></i> Sentence 2: <i><sentence2></i> Label: <i><label></i>

Table A.3: Minimal prompts used for ABSTRACT.

Appendix B

Result Tables

B.1 Additional Results

B.1.1 Task-by-Task Analysis

Figure B.2, Figure B.3, Figure B.4, and Figure B.5 show task-type breakdown results. Though individual task-type results are more noisy, we can make a similar observation compared to the main result – ABSTRACT (TL) scales better with sizes and numbers of examples compared to RANDOM (TR).

B.1.2 Task-by-Task Analysis

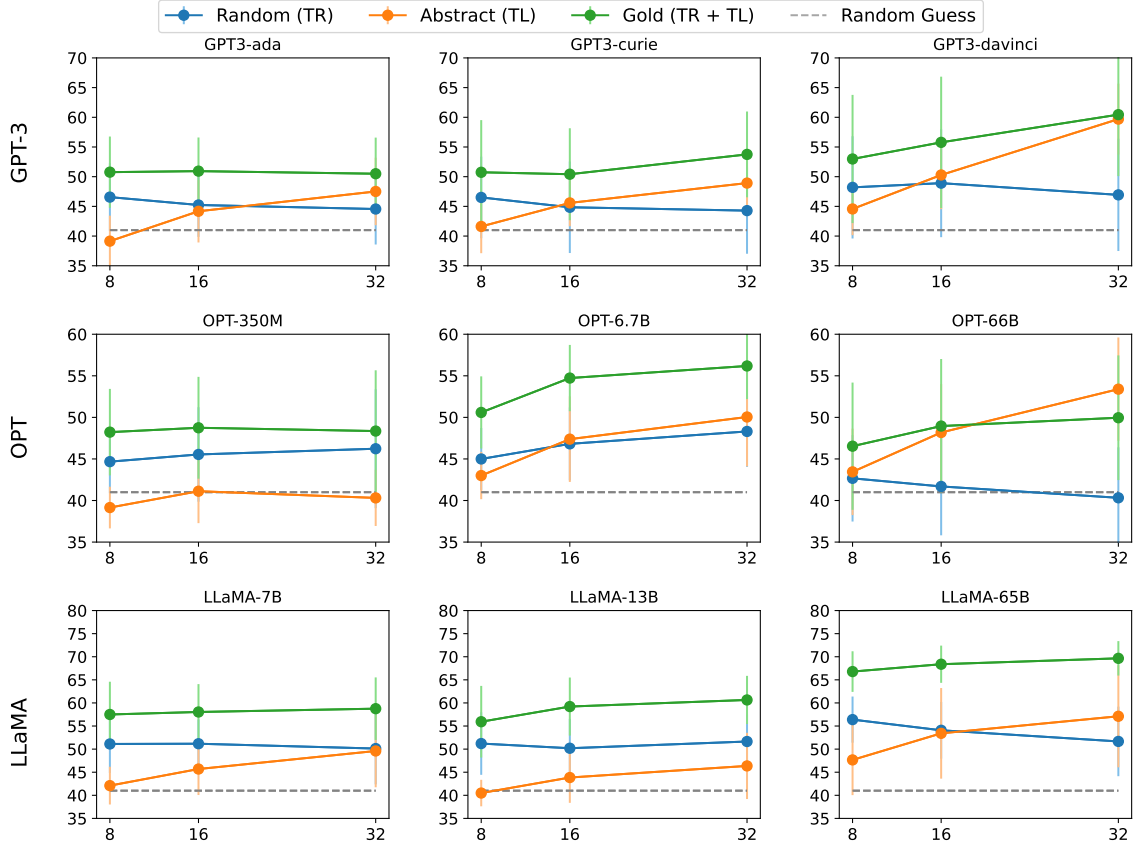


Figure B.1: Averaged task performance across 17 datasets for OPT (bottom), LLaMA (middle), and 17 for GPT-3 (top). x-axis shows number of demonstrations in the prompt. For each model, we run experiments for RANDOM (left), ABSTRACT(middle), and GOLD (right) demonstrations. Variance is calculated across three templates.

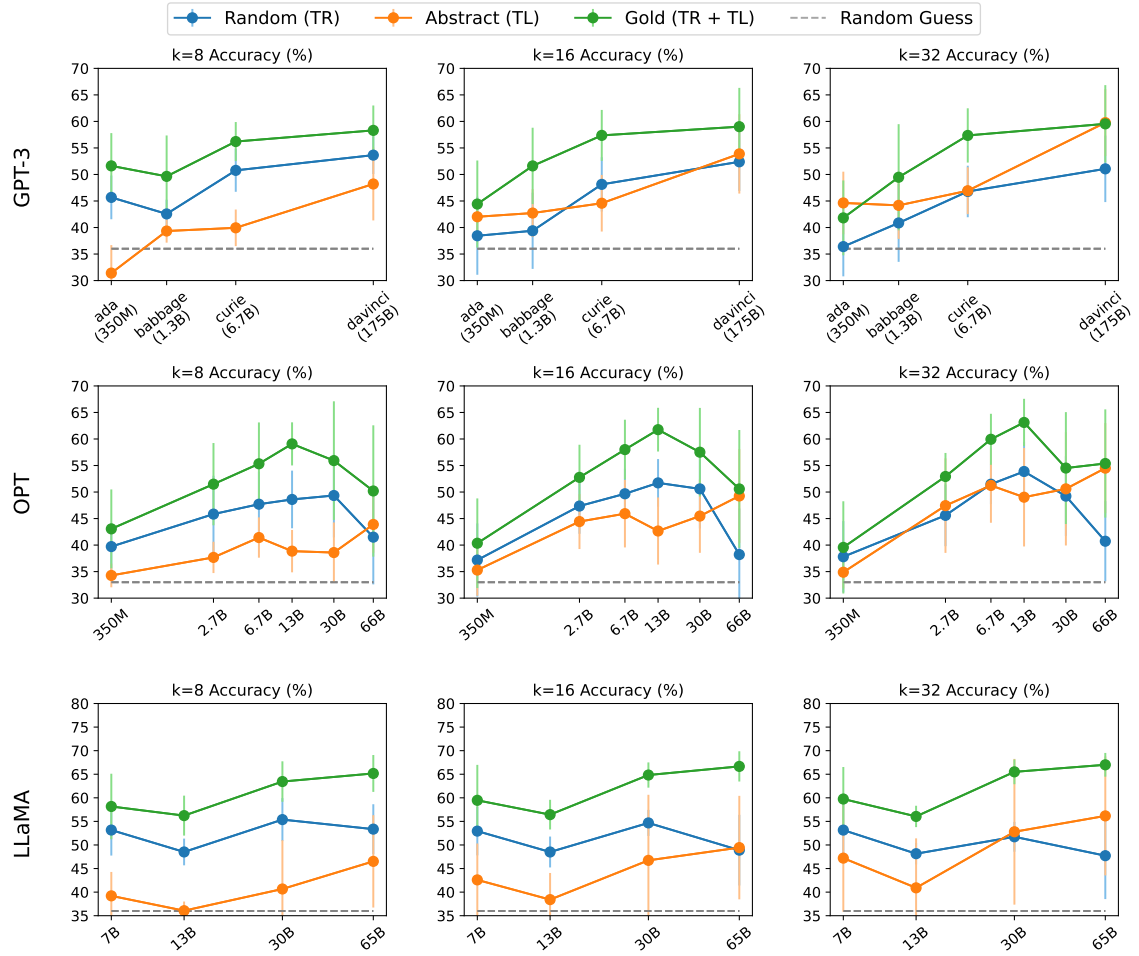


Figure B.2: Average performance of **sentiment analysis** datasets.

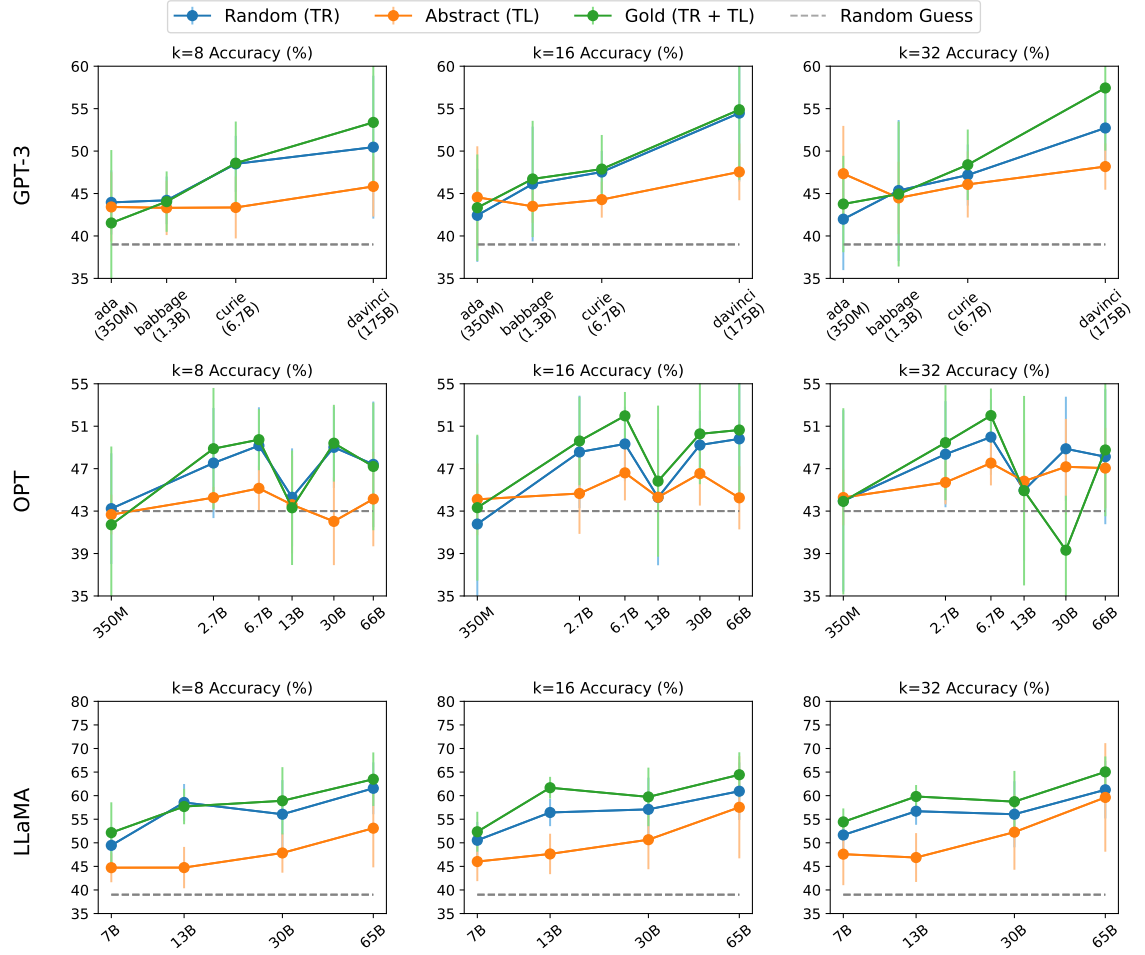


Figure B.3: Average performance of **natural language inference/paraphrase detection** datasets.

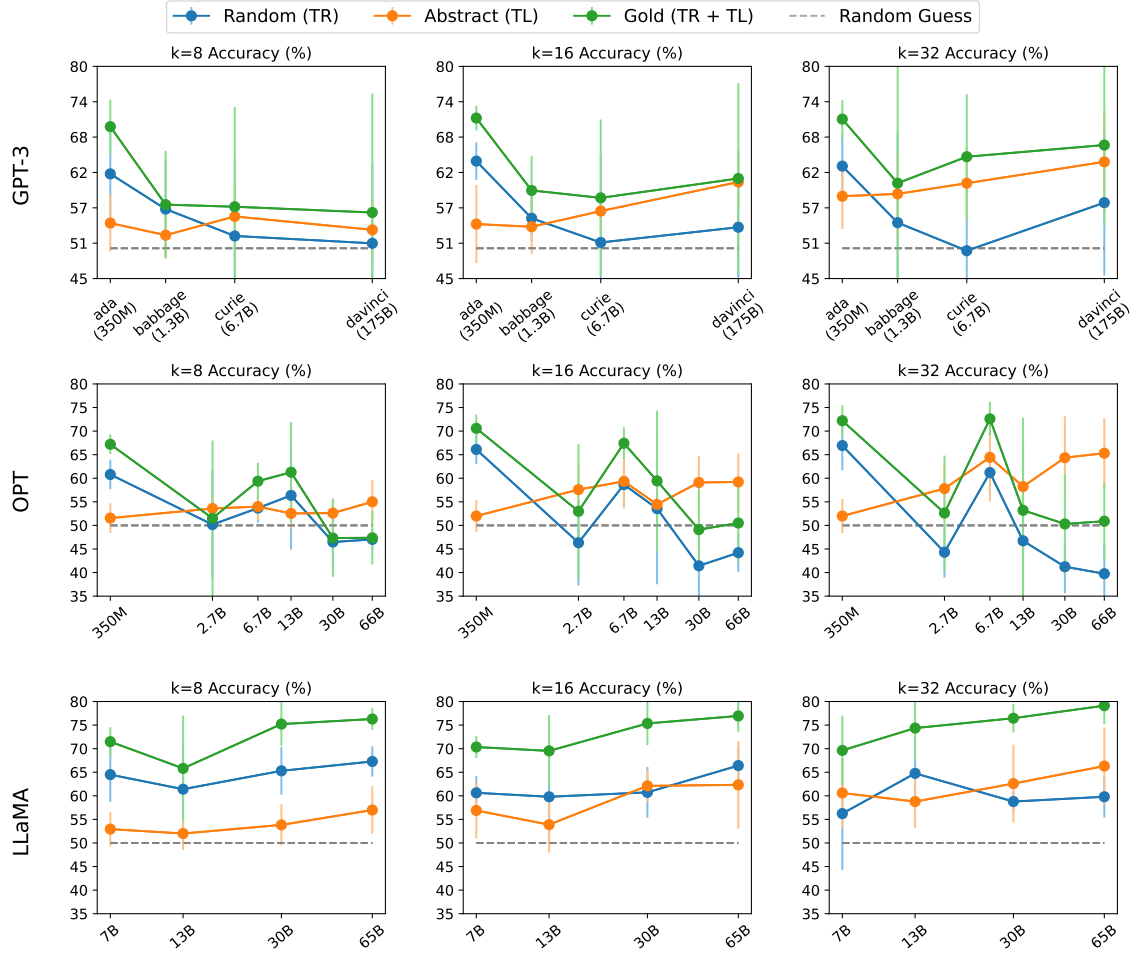


Figure B.4: Average performance of **toxicity detection** datasets.

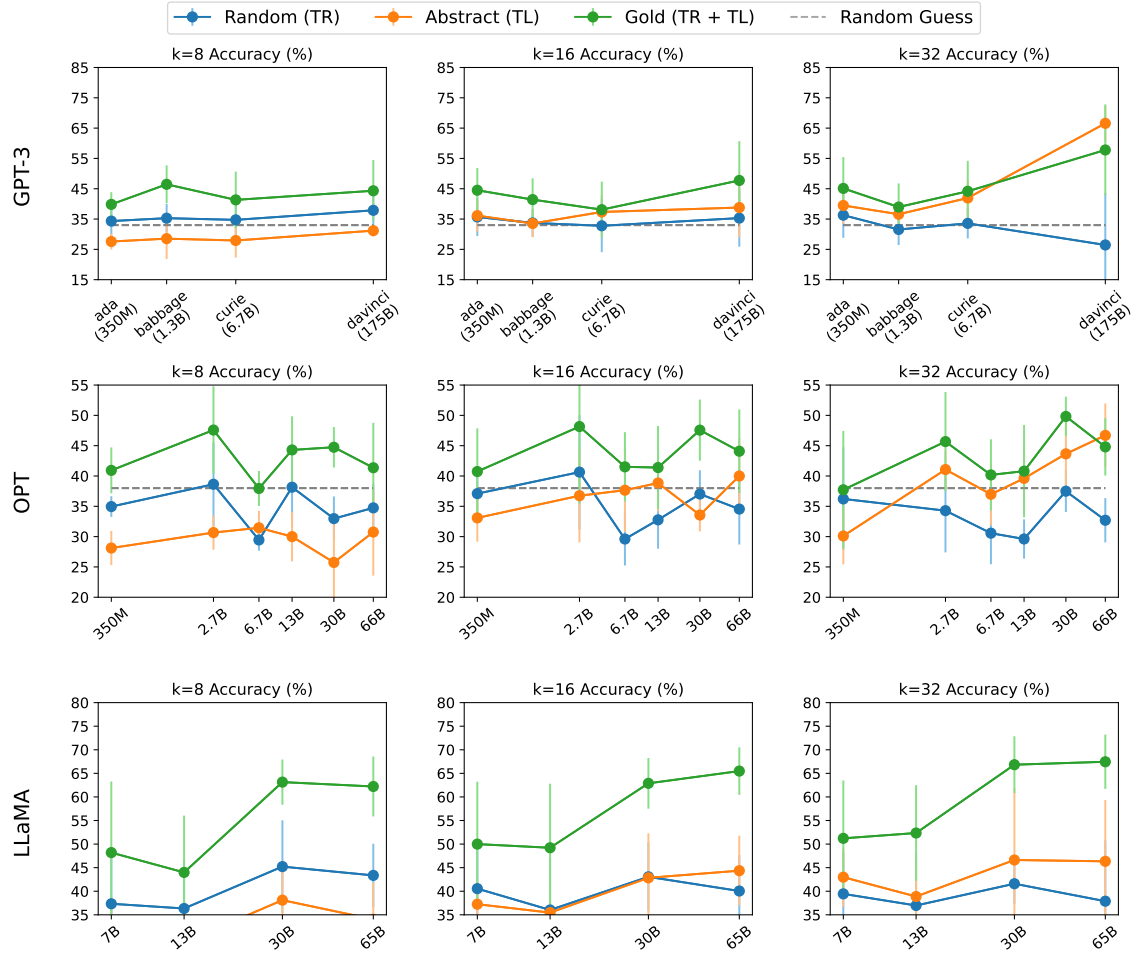


Figure B.5: Average performance of **topic/stance classification** datasets.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.52	0.51	0.54	0.45	0.23	0.4	0.4	0.38	0.41	0.44	0.34	0.43
babbage	0.51	0.52	0.54	0.38	0.37	0.43	0.46	0.29	0.49	0.53	0.34	0.57
curie	0.55	0.54	0.6	0.28	0.33	0.32	0.39	0.32	0.4	0.56	0.36	0.56
davinci	0.56	0.55	0.59	0.34	0.33	0.33	0.4	0.4	0.38	0.4	0.39	0.44
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.23	0.4	0.4	0.38	0.41	0.44	0.34	0.43	0.56	0.39	0.64	0.62
babbage	0.37	0.43	0.46	0.29	0.49	0.53	0.34	0.57	0.45	0.39	0.55	0.54
curie	0.33	0.32	0.39	0.32	0.4	0.56	0.36	0.56	0.54	0.42	0.63	0.53
davinci	0.33	0.33	0.4	0.4	0.38	0.4	0.39	0.44	0.4	0.56	0.44	0.52
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.41	0.44	0.34	0.43	0.56	0.39	0.64	0.62	0.52	0.71	0.62	0.57
babbage	0.49	0.53	0.34	0.57	0.45	0.39	0.55	0.54	0.51	0.52	0.58	0.56
curie	0.4	0.56	0.36	0.56	0.54	0.42	0.63	0.53	0.52	0.6	0.48	0.54
davinci	0.38	0.4	0.39	0.44	0.4	0.56	0.44	0.52	0.51	0.54	0.47	0.52
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.56	0.39	0.64	0.62	0.52	0.71	0.62	0.57	0.76	0.68	0.54	0.74
babbage	0.45	0.39	0.55	0.54	0.51	0.52	0.58	0.56	0.62	0.63	0.51	0.61
curie	0.54	0.42	0.63	0.53	0.52	0.6	0.48	0.54	0.55	0.56	0.6	0.54
davinci	0.4	0.56	0.44	0.52	0.51	0.54	0.47	0.52	0.5	0.48	0.53	0.62

Table B.1: Single dataset accuracies across the GPT-3 model family, using 8 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.51	0.51	0.52	0.44	0.37	0.48	0.4	0.42	0.41	0.37	0.44	0.44
babbage	0.48	0.54	0.55	0.36	0.41	0.31	0.44	0.33	0.48	0.54	0.38	0.54
curie	0.54	0.58	0.62	0.28	0.48	0.3	0.33	0.38	0.32	0.56	0.41	0.56
davinci	0.56	0.6	0.64	0.34	0.42	0.39	0.29	0.44	0.38	0.46	0.49	0.49
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.37	0.48	0.4	0.42	0.41	0.37	0.44	0.44	0.54	0.53	0.67	0.68
babbage	0.41	0.31	0.44	0.33	0.48	0.54	0.38	0.54	0.43	0.53	0.63	0.56
curie	0.48	0.3	0.33	0.38	0.32	0.56	0.41	0.56	0.5	0.55	0.71	0.54
davinci	0.42	0.39	0.29	0.44	0.38	0.46	0.49	0.49	0.38	0.63	0.49	0.51
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.41	0.37	0.44	0.44	0.54	0.53	0.67	0.68	0.52	0.76	0.68	0.52
babbage	0.48	0.54	0.38	0.54	0.43	0.53	0.63	0.56	0.53	0.61	0.58	0.54
curie	0.32	0.56	0.41	0.56	0.5	0.55	0.71	0.54	0.56	0.55	0.49	0.55
davinci	0.38	0.46	0.49	0.49	0.38	0.63	0.49	0.51	0.6	0.56	0.5	0.57
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.54	0.53	0.67	0.68	0.52	0.76	0.68	0.52	0.75	0.69	0.57	0.77
babbage	0.43	0.53	0.63	0.56	0.53	0.61	0.58	0.54	0.61	0.55	0.54	0.62
curie	0.5	0.55	0.71	0.54	0.56	0.55	0.49	0.55	0.59	0.49	0.55	0.59
davinci	0.38	0.63	0.49	0.51	0.6	0.56	0.5	0.57	0.59	0.54	0.67	0.63

Table B.2: Single dataset accuracies across the GPT-3 model family, using 16 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.48	0.52	0.53	0.4	0.37	0.42	0.41	0.38	0.42	0.24	0.45	0.27
babbage	0.53	0.58	0.52	0.32	0.38	0.35	0.42	0.35	0.38	0.44	0.4	0.5
curie	0.54	0.59	0.66	0.26	0.47	0.31	0.38	0.4	0.43	0.57	0.41	0.57
davinci	0.57	0.64	0.66	0.29	0.51	0.37	0.28	0.49	0.37	0.43	0.52	0.49
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.37	0.42	0.41	0.38	0.42	0.24	0.45	0.27	0.55	0.56	0.69	0.66
babbage	0.38	0.35	0.42	0.35	0.38	0.44	0.4	0.5	0.51	0.58	0.65	0.51
curie	0.47	0.31	0.38	0.4	0.43	0.57	0.41	0.57	0.52	0.56	0.71	0.51
davinci	0.51	0.37	0.28	0.49	0.37	0.43	0.52	0.49	0.35	0.68	0.5	0.63
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.42	0.24	0.45	0.27	0.55	0.56	0.69	0.66	0.55	0.73	0.69	0.61
babbage	0.38	0.44	0.4	0.5	0.51	0.58	0.65	0.51	0.57	0.63	0.6	0.59
curie	0.43	0.57	0.41	0.57	0.52	0.56	0.71	0.51	0.59	0.65	0.5	0.61
davinci	0.37	0.43	0.52	0.49	0.35	0.68	0.5	0.63	0.6	0.63	0.51	0.62
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
ada	0.55	0.56	0.69	0.66	0.55	0.73	0.69	0.61	0.73	0.65	0.63	0.77
babbage	0.51	0.58	0.65	0.51	0.57	0.63	0.6	0.59	0.64	0.57	0.56	0.65
curie	0.52	0.56	0.71	0.51	0.59	0.65	0.5	0.61	0.63	0.44	0.61	0.69
davinci	0.35	0.68	0.5	0.63	0.6	0.63	0.51	0.62	0.65	0.6	0.7	0.71

Table B.3: Single dataset accuracies across the GPT-3 model family, using 32 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.49	0.51	0.53	0.43	0.34	0.48	0.41	0.31	0.45	0.33	0.34	0.29
OPT-2.7B	0.52	0.55	0.56	0.43	0.36	0.45	0.47	0.34	0.5	0.52	0.34	0.55
OPT-6.7B	0.53	0.53	0.57	0.26	0.33	0.27	0.33	0.39	0.36	0.46	0.36	0.48
OPT-13B	0.55	0.52	0.61	0.4	0.35	0.4	0.49	0.35	0.47	0.36	0.3	0.37
OPT-30B	0.52	0.54	0.55	0.28	0.24	0.35	0.4	0.34	0.46	0.53	0.31	0.55
OPT-66B	0.52	0.55	0.53	0.29	0.38	0.32	0.44	0.37	0.42	0.44	0.36	0.47
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.34	0.48	0.41	0.31	0.45	0.33	0.34	0.29	0.48	0.36	0.48	0.6
OPT-2.7B	0.36	0.45	0.47	0.34	0.5	0.52	0.34	0.55	0.54	0.42	0.56	0.49
OPT-6.7B	0.33	0.27	0.33	0.39	0.36	0.46	0.36	0.48	0.63	0.44	0.74	0.55
OPT-13B	0.35	0.4	0.49	0.35	0.47	0.36	0.3	0.37	0.59	0.44	0.69	0.62
OPT-30B	0.24	0.35	0.4	0.34	0.46	0.53	0.31	0.55	0.56	0.43	0.61	0.44
OPT-66B	0.38	0.32	0.44	0.37	0.42	0.44	0.36	0.47	0.33	0.44	0.46	0.45
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.45	0.33	0.34	0.29	0.48	0.36	0.48	0.6	0.49	0.66	0.65	0.51
OPT-2.7B	0.5	0.52	0.34	0.55	0.54	0.42	0.56	0.49	0.53	0.49	0.51	0.56
OPT-6.7B	0.36	0.46	0.36	0.48	0.63	0.44	0.74	0.55	0.54	0.59	0.53	0.57
OPT-13B	0.47	0.36	0.3	0.37	0.59	0.44	0.69	0.62	0.53	0.63	0.55	0.53
OPT-30B	0.46	0.53	0.31	0.55	0.56	0.43	0.61	0.44	0.49	0.42	0.46	0.57
OPT-66B	0.42	0.44	0.36	0.47	0.33	0.44	0.46	0.45	0.55	0.53	0.44	0.55
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.48	0.36	0.48	0.6	0.49	0.66	0.65	0.51	0.71	0.66	0.53	0.73
OPT-2.7B	0.54	0.42	0.56	0.49	0.53	0.49	0.51	0.56	0.52	0.48	0.51	0.5
OPT-6.7B	0.63	0.44	0.74	0.55	0.54	0.59	0.53	0.57	0.61	0.53	0.52	0.62
OPT-13B	0.59	0.44	0.69	0.62	0.53	0.63	0.55	0.53	0.61	0.55	0.52	0.62
OPT-30B	0.56	0.43	0.61	0.44	0.49	0.42	0.46	0.57	0.47	0.46	0.51	0.46
OPT-66B	0.33	0.44	0.46	0.45	0.55	0.53	0.44	0.55	0.38	0.49	0.55	0.56

Table B.4: Single dataset accuracies across the OPT model family, using 8 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.52	0.53	0.55	0.47	0.37	0.49	0.42	0.42	0.44	0.33	0.36	0.35
OPT-2.7B	0.52	0.56	0.58	0.44	0.44	0.47	0.51	0.39	0.46	0.55	0.39	0.57
OPT-6.7B	0.52	0.57	0.57	0.22	0.39	0.28	0.39	0.43	0.41	0.48	0.42	0.54
OPT-13B	0.58	0.54	0.62	0.32	0.44	0.38	0.41	0.39	0.41	0.36	0.4	0.36
OPT-30B	0.51	0.57	0.57	0.34	0.4	0.35	0.41	0.32	0.5	0.55	0.45	0.56
OPT-66B	0.5	0.57	0.54	0.25	0.47	0.31	0.47	0.44	0.48	0.49	0.38	0.51
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.37	0.49	0.42	0.42	0.44	0.33	0.36	0.35	0.45	0.4	0.47	0.65
OPT-2.7B	0.44	0.47	0.51	0.39	0.46	0.55	0.39	0.57	0.53	0.5	0.58	0.45
OPT-6.7B	0.39	0.28	0.39	0.43	0.41	0.48	0.42	0.54	0.66	0.53	0.8	0.59
OPT-13B	0.44	0.38	0.41	0.39	0.41	0.36	0.4	0.36	0.6	0.53	0.72	0.54
OPT-30B	0.4	0.35	0.41	0.32	0.5	0.55	0.45	0.56	0.56	0.52	0.64	0.35
OPT-66B	0.47	0.31	0.47	0.44	0.48	0.49	0.38	0.51	0.3	0.57	0.49	0.44
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.44	0.33	0.36	0.35	0.45	0.4	0.47	0.65	0.52	0.71	0.71	0.52
OPT-2.7B	0.46	0.55	0.39	0.57	0.53	0.5	0.58	0.45	0.59	0.47	0.41	0.62
OPT-6.7B	0.41	0.48	0.42	0.54	0.66	0.53	0.8	0.59	0.56	0.71	0.62	0.61
OPT-13B	0.41	0.36	0.4	0.36	0.6	0.53	0.72	0.54	0.53	0.62	0.5	0.55
OPT-30B	0.5	0.55	0.45	0.56	0.56	0.52	0.64	0.35	0.57	0.43	0.38	0.63
OPT-66B	0.48	0.49	0.38	0.51	0.3	0.57	0.49	0.44	0.59	0.51	0.4	0.6
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.45	0.4	0.47	0.65	0.52	0.71	0.71	0.52	0.76	0.73	0.51	0.76
OPT-2.7B	0.53	0.5	0.58	0.45	0.59	0.47	0.41	0.62	0.52	0.45	0.54	0.54
OPT-6.7B	0.66	0.53	0.8	0.59	0.56	0.71	0.62	0.61	0.69	0.64	0.61	0.74
OPT-13B	0.6	0.53	0.72	0.54	0.53	0.62	0.5	0.55	0.58	0.55	0.53	0.58
OPT-30B	0.56	0.52	0.64	0.35	0.57	0.43	0.38	0.63	0.5	0.41	0.59	0.51
OPT-66B	0.3	0.57	0.49	0.44	0.59	0.51	0.4	0.6	0.46	0.46	0.59	0.55

Table B.5: Single dataset accuracies across the OPT model family, using 16 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.53	0.53	0.55	0.42	0.35	0.42	0.43	0.33	0.4	0.36	0.34	0.35
OPT-2.7B	0.51	0.59	0.59	0.31	0.42	0.42	0.43	0.39	0.42	0.53	0.4	0.57
OPT-6.7B	0.55	0.59	0.6	0.26	0.29	0.24	0.4	0.39	0.42	0.49	0.44	0.53
OPT-13B	0.56	0.58	0.59	0.25	0.45	0.36	0.39	0.38	0.42	0.4	0.38	0.37
OPT-30B	0.52	0.59	0.57	0.32	0.47	0.42	0.47	0.42	0.47	0.54	0.45	0.6
OPT-66B	0.48	0.58	0.51	0.27	0.5	0.26	0.4	0.46	0.5	0.45	0.43	0.47
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.35	0.42	0.43	0.33	0.4	0.36	0.34	0.35	0.44	0.38	0.44	0.67
OPT-2.7B	0.42	0.42	0.43	0.39	0.42	0.53	0.4	0.57	0.51	0.56	0.58	0.46
OPT-6.7B	0.29	0.24	0.4	0.39	0.42	0.49	0.44	0.53	0.68	0.61	0.82	0.63
OPT-13B	0.45	0.36	0.39	0.38	0.42	0.4	0.38	0.37	0.61	0.6	0.72	0.48
OPT-30B	0.47	0.42	0.47	0.42	0.47	0.54	0.45	0.6	0.57	0.57	0.7	0.4
OPT-66B	0.5	0.26	0.4	0.46	0.5	0.45	0.43	0.47	0.37	0.64	0.57	0.41
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.4	0.36	0.34	0.35	0.44	0.38	0.44	0.67	0.51	0.73	0.71	0.51
OPT-2.7B	0.42	0.53	0.4	0.57	0.51	0.56	0.58	0.46	0.55	0.49	0.43	0.6
OPT-6.7B	0.42	0.49	0.44	0.53	0.68	0.61	0.82	0.63	0.65	0.74	0.62	0.65
OPT-13B	0.42	0.4	0.38	0.37	0.61	0.6	0.72	0.48	0.56	0.57	0.44	0.64
OPT-30B	0.47	0.54	0.45	0.6	0.57	0.57	0.7	0.4	0.55	0.42	0.36	0.66
OPT-66B	0.5	0.45	0.43	0.47	0.37	0.64	0.57	0.41	0.63	0.52	0.36	0.67
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
OPT-350M	0.44	0.38	0.44	0.67	0.51	0.73	0.71	0.51	0.77	0.74	0.52	0.79
OPT-2.7B	0.51	0.56	0.58	0.46	0.55	0.49	0.43	0.6	0.54	0.41	0.56	0.48
OPT-6.7B	0.68	0.61	0.82	0.63	0.65	0.74	0.62	0.65	0.78	0.65	0.64	0.77
OPT-13B	0.61	0.6	0.72	0.48	0.56	0.57	0.44	0.64	0.5	0.45	0.53	0.5
OPT-30B	0.57	0.57	0.7	0.4	0.55	0.42	0.36	0.66	0.46	0.4	0.71	0.54
OPT-66B	0.37	0.64	0.57	0.41	0.63	0.52	0.36	0.67	0.49	0.4	0.69	0.56

Table B.6: Single dataset accuracies across the OPT model family, using 32 examples.

	tweet_eval_hate			tweet_eval_atheism			tweet_eval_feminist			sick		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.59	0.53	0.64	0.33	0.31	0.37	0.41	0.43	0.45	0.32	0.36	0.38
13B	0.63	0.53	0.65	0.31	0.34	0.28	0.43	0.34	0.44	0.39	0.41	0.41
30B	0.64	0.58	0.72	0.38	0.47	0.52	0.57	0.49	0.65	0.37	0.43	0.41
65B	0.69	0.58	0.72	0.4	0.42	0.58	0.54	0.42	0.58	0.38	0.46	0.41
	financial_phrasebank			ethos_race			ethos_gender			ethos_religion		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.31	0.37	0.41	0.43	0.45	0.32	0.36	0.38	0.64	0.4	0.7	0.65
13B	0.34	0.28	0.43	0.34	0.44	0.39	0.41	0.41	0.42	0.35	0.61	0.61
30B	0.47	0.52	0.57	0.49	0.65	0.37	0.43	0.41	0.65	0.38	0.79	0.69
65B	0.42	0.58	0.54	0.42	0.58	0.38	0.46	0.41	0.6	0.44	0.83	0.69
	ethos_national_origin			snli			sst2			trec		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.45	0.32	0.36	0.38	0.64	0.4	0.7	0.65	0.56	0.73	0.61	0.53
13B	0.44	0.39	0.41	0.41	0.42	0.35	0.61	0.61	0.52	0.66	0.59	0.5
30B	0.65	0.37	0.43	0.41	0.65	0.38	0.79	0.69	0.52	0.76	0.65	0.52
65B	0.58	0.38	0.46	0.41	0.6	0.44	0.83	0.69	0.55	0.75	0.65	0.56
	rte			wnli			mrpc			poem		
	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.64	0.4	0.7	0.65	0.56	0.73	0.61	0.53	0.7	0.71	0.52	0.78
13B	0.42	0.35	0.61	0.61	0.52	0.66	0.59	0.5	0.64	0.71	0.54	0.78
30B	0.65	0.38	0.79	0.69	0.52	0.76	0.65	0.52	0.77	0.67	0.56	0.86
65B	0.6	0.44	0.83	0.69	0.55	0.75	0.65	0.56	0.77	0.73	0.6	0.87

Table B.7: Single dataset accuracies across the LLaMA model family, using 8 examples.

tweet_eval_hate				tweet_eval_atheism			tweet_eval_feminist			sick		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.61	0.58	0.66	0.33	0.49	0.37	0.41	0.35	0.45	0.31	0.43	0.36
13B	0.6	0.58	0.66	0.27	0.5	0.34	0.4	0.34	0.42	0.37	0.42	0.41
30B	0.67	0.67	0.74	0.37	0.54	0.53	0.47	0.5	0.62	0.36	0.51	0.42
65B	0.66	0.62	0.73	0.37	0.56	0.6	0.52	0.53	0.6	0.38	0.55	0.42
financial_phrasebank				ethos_race			ethos_gender			ethos_religion		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.49	0.37	0.41	0.35	0.45	0.31	0.43	0.36	0.65	0.46	0.72	0.6
13B	0.5	0.34	0.4	0.34	0.42	0.37	0.42	0.41	0.41	0.39	0.59	0.56
30B	0.54	0.53	0.47	0.5	0.62	0.36	0.51	0.42	0.64	0.49	0.84	0.6
65B	0.56	0.6	0.52	0.53	0.6	0.38	0.55	0.42	0.56	0.54	0.87	0.62
ethos_national_origin				snli			sst2			trec		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.45	0.31	0.43	0.36	0.65	0.46	0.72	0.6	0.53	0.72	0.57	0.59
13B	0.42	0.37	0.42	0.41	0.41	0.39	0.59	0.56	0.51	0.66	0.59	0.5
30B	0.62	0.36	0.51	0.42	0.64	0.49	0.84	0.6	0.58	0.74	0.6	0.65
65B	0.6	0.38	0.55	0.42	0.56	0.54	0.87	0.62	0.58	0.75	0.66	0.65
rte				wnli			mrpc			poem		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.65	0.46	0.72	0.6	0.53	0.72	0.57	0.59	0.67	0.65	0.59	0.78
13B	0.41	0.39	0.59	0.56	0.51	0.66	0.59	0.5	0.73	0.69	0.54	0.78
30B	0.64	0.49	0.84	0.6	0.58	0.74	0.6	0.65	0.74	0.65	0.64	0.85
65B	0.56	0.54	0.87	0.62	0.58	0.75	0.66	0.65	0.78	0.73	0.64	0.85

Table B.8: Single dataset accuracies across the LLaMA model family, using 16 examples.

tweet_eval_hate				tweet_eval_atheism			tweet_eval_feminist			sick		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.58	0.58	0.64	0.33	0.51	0.35	0.4	0.38	0.47	0.36	0.46	0.4
13B	0.6	0.59	0.68	0.3	0.46	0.37	0.41	0.42	0.46	0.36	0.42	0.42
30B	0.65	0.64	0.73	0.32	0.53	0.6	0.48	0.51	0.63	0.35	0.55	0.42
65B	0.64	0.68	0.78	0.38	0.51	0.6	0.45	0.49	0.63	0.36	0.62	0.43
financial_phrasebank				ethos_race			ethos_gender			ethos_religion		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.51	0.35	0.4	0.38	0.47	0.36	0.46	0.4	0.64	0.5	0.74	0.61
13B	0.46	0.37	0.41	0.42	0.46	0.36	0.42	0.42	0.38	0.38	0.56	0.65
30B	0.53	0.6	0.48	0.51	0.63	0.35	0.55	0.42	0.61	0.61	0.88	0.66
65B	0.51	0.6	0.45	0.49	0.63	0.36	0.62	0.43	0.52	0.66	0.88	0.59
ethos_national_origin				snli			sst2			trec		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.47	0.36	0.46	0.4	0.64	0.5	0.74	0.61	0.59	0.67	0.47	0.62
13B	0.46	0.36	0.42	0.42	0.38	0.38	0.56	0.65	0.53	0.73	0.67	0.57
30B	0.63	0.35	0.55	0.42	0.61	0.61	0.88	0.66	0.6	0.74	0.55	0.6
65B	0.63	0.36	0.62	0.43	0.52	0.66	0.88	0.59	0.63	0.76	0.58	0.66
rte				wnli			mrpc			poem		
Random	Abstract	Gold		Random	Abstract	Gold	Random	Abstract	Gold	Random	Abstract	Gold
7B	0.64	0.5	0.74	0.61	0.59	0.67	0.47	0.62	0.69	0.65	0.64	0.79
13B	0.38	0.38	0.56	0.65	0.53	0.73	0.67	0.57	0.76	0.7	0.62	0.83
30B	0.61	0.61	0.88	0.66	0.6	0.74	0.55	0.6	0.8	0.57	0.65	0.82
65B	0.52	0.66	0.88	0.59	0.63	0.76	0.58	0.66	0.77	0.63	0.73	0.87

Table B.9: Single dataset accuracies across the LLaMA model family, using 32 examples.

Bibliography

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2022. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers.
- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *the Third International Workshop on Paraphrasing (IWP2005)*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723.
- Mitchell A Gordon, Kevin Duh, and Jared Kaplan. 2021. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling laws for autoregressive generative modeling.

- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *CoRR*, abs/2104.08786.
- P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation (LREC)*.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Ha-

- jishirzi. 2021. Reframing instructional prompts to gptk’s language. *CoRR*, abs/2109.07830.
- Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Frieda Rong. 2021. Extrapolating to unnatural language processing with gpt-3’s in-context learning: The good, the bad, and the mysterious.
- Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning.
- Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.
- Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *CoRR*, abs/2010.15980.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubakaran, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Free-

man, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia

Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimeo Xu, Mirac Suzgun, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Milkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramón Risco Delgado, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schus-

ter, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Mishergahi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Timothy Telleen-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ioannis Mollas Tsoumakas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios. 2020. Ethos: an online hate speech detection dataset. In *preprint*.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexan-

- der Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022a. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*.
- Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022b. Transformers learn in-context by gradient descent.
- Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations (ICLR)*.
- Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui

- Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *CoRR*, abs/2102.09690.