Are Emergent Abilities in Large Language Models just In-Context Learning?

Sheng Lu[1*], Irina Bigoulaeva[1*], Rachneet Sachdeva[1],
Harish Tayyar Madabushi[2], and Iryna Gurevych[1]

[1] Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
[2] Department of Computer Science, The University of Bath

www.ukp.tu-darmstadt.de
htm43@bath.ac.uk

Abstract

Large language models, comprising billions of parameters and pre-trained on extensive web-scale corpora, have been claimed to acquire certain capabilities without having been specifically trained on them. These capabilities, referred to as "emergent abilities," have been a driving force in discussions regarding the potentials and risks of language models. A key challenge in evaluating emergent abilities is that they are confounded by model competencies that arise through alternative prompting techniques, including in-context learning, which is the ability of models to complete a task based on a few examples. We present a novel theory that explains emergent abilities, taking into account their potential confounding factors, and rigorously substantiate this theory through over 1000 experiments. Our findings suggest that purported emergent abilities are not truly emergent, but result from a combination of in-context learning, model memory, and linguistic knowledge. Our work is a foundational step in explaining language model performance, providing a template for their efficient use and clarifying the paradox of their ability to excel in some instances while faltering in others. Thus, we demonstrate that their capabilities should not be overestimated. [1]

1 Introduction, Motivation and Context

One of the most captivating aspects of pre-trained language models (PLMs) is their capacity to acquire a wide range of knowledge across different domains, while being trained primarily through masked language modelling, a task requiring models to predict masked tokens in their input (Tenney et al., 2019; Petroni et al., 2019). The diverse abilities of PLMs can be categorised into two broad types: formal linguistic abilities and functional linguistic abilities. Formal linguistic abilities refer to the understanding of language rules and patterns, which PLMs, for example, BERT (Devlin et al., 2019) are known to excel at (Tenney et al., 2019; Petroni et al., 2019). The latter category includes a range of abilities akin to human cognition that are necessary for real-world language use and comprehension, such as commonsense knowledge and social awareness. While PLMs excel at formal linguistic abilities, they have faced challenges in developing functional linguistic abilities (Mahowald et al., 2023).

The introduction of Large Language Models (LLMs), which are typically generative PLMs scaled up to billions of parameters and trained on vast, web-scale data corpora, is changing this landscape (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023a,b). Recent works indicate that LLMs exhibit *emergent abilities*, as measured by their above random performance without explicit training on tasks, including those tasks that explicitly require some form of reasoning. An emergent ability was first defined as an ability to solve a task which is absent in smaller models, but present in LLMs. This definition, introduced approximately concurrently by two works (Wei et al., 2022b; Srivastava et al., 2023), is based on the more general definition of emergence in physics: "Emergence is when quantitative changes in a system result in qualitative changes in behaviour" (Anderson, 1972). Emergent abilities are implied due to LLMs' capacity to perform above the random baseline on the corresponding tasks without explicit training on those same tasks. For example, the emergent ability to understand social situations in LLMs is inferred from LLMs' performing well above the random baseline on the Social IQA (Sap et al., 2019) task, which serves to evaluate models' emotional and social intelligence and includes questions such as "Carson was excited to wake up to attend school.

Why did he do this? Options: Take the big test, Go to bed early, Just say hello to friend (correct)".

1.1 Significance for Applications and Safety

While prior work on emergent abilities does not explicitly make the distinction between formal and functional linguistic abilities, the identification of numerous functional linguistic capabilities holds profound implications for both the potential and safety of LLMs. The assumption that LLMs have access to emergent functional linguistic abilities significantly affects the way in which users interact with and use these systems. Overreliance on these perceived abilities can lead users to provide insufficiently detailed instructions, potentially resulting in hallucinations and errors. If there are indeed multiple functional linguistic abilities that emerge with scale, it suggests that further scaling has the potential to unlock a wide array of additional abilities which we cannot predict, especially since they tend not to present themselves in smaller-scale models (Wei et al., 2022b). This inherent unpredictability associated with emergent abilities holds substantial implications for the discussion surrounding safety and security when utilising LLMs. Indeed, it has been argued that these could include potentially hazardous abilities, including reasoning and planning (Hoffmann, 2023), thereby posing an existential threat to humanity (Bengio et al., 2023). In this work, we refer to such potentially harmful capabilities, as "latent hazardous abilities."

It's important to emphasise that the development of linguistic proficiencies (i.e. formal linguistic abilities) does *not* carry the potentials of this nature. The same can be said for the capacity to efficiently handle information retrieval tasks. The real focus lies on potential capabilities relating to functional linguistic abilities. However, it must be emphasised that this does not include other dangers posed through the misuse of these models, such as the use of LLMs to generate fake news. Similarly, we do not contend that future AI systems could *never* pose an existential threat. Instead, we clarify that, contrary to prevailing narratives, the evidence from LLM abilities does *not* support this concern.

1.2 Abilities vs. Techniques

The scaling up of LLMs facilitates the acquisition of diverse competencies, which can be grouped into two categories: The first encompasses *abilities*, already described. The second encompasses various *techniques*, which LLMs can benefit from.

These techniques show less of an effect in smaller models, but become progressively more effective with scale. Among these techniques are in-context learning and instruction-tuning. In-context learning (ICL) is the technique wherein LLMs are provided with a limited number of examples within the input prompt itself (Brown et al., 2020). From these examples, the model infers how to perform a specific task, responding appropriately to the question posed by the prompt (Brown et al., 2020; Liu et al., 2023). Investigations into the theoretical underpinnings of ICL and its specific manifestation in LLMs indicate that it might bear resemblance to the process of fine-tuning models on the specific tasks for which they are provided examples (Akyürek et al., 2023; Dai et al., 2023; von Oswald et al., 2023; Wei et al., 2023). Another technique exclusive to LLMs is instructional fine-tuning, alternatively known as instruction-tuning. This technique involves fine-tuning LLMs on datasets of prompts and their corresponding desired outputs, which enables the models to follow explicit instructions in prompts (Chung et al., 2022; Wei et al., 2022a; Taori et al., 2023). Following previous work (Wei et al., 2022b), we refer to these techniques, illustrated in Figure 3, as *prompting techniques*.

Significant to our investigation is the observation that prompting techniques and emergent abilities manifest within LLMs at a comparable scale. Furthermore, ICL and instruction-tuning can be observed in smaller-scale models, albeit to a lesser degree, and are thus predictable. This predictability means they are not 'emergent', nor do they pose a threat, contrasting with the unpredictability and potential risks associated with emergent abilities in larger models. Considering this context, it becomes imperative to ascertain the extent of these emergent abilities in the absence of prompting techniques.

1.3 Fine-tuning, In-Context Learning, and other Prompting Techniques

Artificial neural models have, for some time, exhibited tremendous success on specific tasks when trained on those tasks (Devlin et al., 2019; Liu et al., 2019). PLMs in particular have demonstrated this even when trained on just a few examples (Hofer et al., 2018; Radford et al., 2019; Brown et al., 2020; Gao et al., 2021). Such performance is not considered "emergent", precisely because models are trained on that very task. Indeed, the fact that LLMs are *not* trained on the tasks used in evaluating their emergent abilities is central to identifying

abilities which are *truly* emergent. The assertion that achieving satisfactory performance on a given task signifies the *emergence* of associated 'abilities' hinges on the condition that models are not explicitly trained for that specific task.

The recent insights indicating parallels between ICL and explicit training suggest that the success on a task through ICL, much like models trained explicitly for task-solving, does not imply a model inherently possessing that *ability* (Dai et al., 2023). For example, it has been shown that ICL implements gradient descent implicitly and constructs a function at inference time on regression problems (Akyürek et al., 2023; Li et al., 2023; Zhang et al., 2023a), which may be related to gradient-based meta-learning (von Oswald et al., 2023). Importantly, however, the specific mechanisms governing ICL do not impact our argument: The fact of its functionality suffices to underscore the necessity of assessing emergent abilities in the absence of ICL. Additionally, instruction-tuning datasets typically include several variations of an instruction followed by the task input or context (see Figure 3). As such, we contend that the process of instruction fine-tuning potentially enables models to map prompts to in-context examples (detailed in Section 4), thereby utilising ICL to respond to prompts. This would imply that the success of a model to solve a task in this scenario also does not indicate the emergence of the corresponding ability.

The safety issues associated with LLMs stem from their ability to perform well above the random baseline on tasks that cannot be solved through memorisation and are indicative of certain 'abilities', *without explicit training on those tasks*. Therefore, recognising that prompts act as a form of 'training mechanism' rather than simply a way of interfacing with a model with inherent functional linguistic abilities offers the potential to alter how we use these models and deepen our understanding of their capabilities and limitations. As such, it is crucial to conduct an independent evaluation of LLMs' abilities, detached from ICL.

1.4 Research Questions and Contributions

Our research seeks to answer two pivotal questions: Firstly, in light of ICL's influence on perceived emergent abilities in LLMs, which abilities are truly emergent in the absence of ICL, including instructional tuning? Secondly, given LLMs' capability for ICL and the typical inclusion of instruction-exemplar mappings in instruction-tuning datasets,

can we find evidence of the emergence of functional linguistic abilities in instruction-tuned models? Or can ICL better explain their capabilities and shortcomings?

Our primary contribution lies in demonstrating the absence of emergent functional linguistic abilities in LLMs when ICL is not a factor, thus demystifying the true capabilities of LLMs and affirming their safety, while additionally dispelling concerns over potential latent hazardous abilities. Our secondary contributions include empirically testing the hypothesis that instruction-tuned models' capabilities stem from efficient ICL, thus offering an explanation for LLMs' abilities as stemming from a combination of formal linguistic skills, vast information retention and recall, and notably, ICL. By identifying user-directable ICL, rather than intrinsic functional linguistic capabilities, as the mechanism behind LLM performance, we lay out a framework for more efficient use of these models, shedding light on their capabilities and limitations.

2 Experimental Setup

In this section, we present an overview of our experimental methods investigating emergent abilities in the absence of ICL. We experiment with 20 models across 22 tasks using two different settings. We use four different evaluation metrics and additionally run multiple tests for bias, including a manual analysis of our results. We present an overview of this setup below, while details on the hyperparameters and training regime are presented in Appendix C.

2.1 Models

We experiment with four model families: GPT, T5 (Raffel et al., 2020), Falcon[2] and LLaMA (Touvron et al., 2023a). We choose these model families, since GPT and LLaMA have previously been found to have emergent abilities, and Falcon is at the top of LLM leaderboards at the time of writing. Finally, we select T5 as it is an encoder-decoder model, and its instruction-tuned version (Flan) is trained using an extensive instruction-tuning dataset. Table 1 enumerates the models that we use in our experiments. The emergence of abilities in relation to scale requires the evaluation of each model family across a range of sizes (parameter counts), and so we select models at different scales from each of these families. Important to our inquiry is the hypothesis that instructional tuning might indirectly

---

[2]See https://falconllm.tii.ae/index.html.

leverage ICL. In light of this possibility, we experiment with both.

Model	Instruction-Tuned Version	Size
GPT-2	GPT-2-IT	117M
GPT-2-XL	GPT-2-XL-IT	1.6B
GPT-J	GPT-JT	6.7B
davinci	text-davinci-001 text-davinci-003	175B
T5-small	Flan-T5-small	60M
T5-large	Flan-T5-large	770M
Falcon-7B	Falcon-7B-Instruct	7B
Falcon-40B	Falcon-40B-Instruct	40B
LLaMA-7B	–	7B
LLaMA-13B	–	13B
LLaMA-30B	–	30B

Table 1: Details of the models used in the experiments.

2.2 Tasks

In selecting tasks to assess the emergence of abilities, we base our selection on those tasks that have been identified as emergent in GPT-3 by prior works. We refer to these tasks as *previously identified as emergent*. Out of 17 such tasks in the BIG-bench dataset (Srivastava et al., 2023), we incorporate 14 into our study. Three tasks previously identified as emergent are excluded from our analysis, because their generative nature made them challenging to assess automatically in a manner consistent with the other tasks. Additionally, to create a baseline for comparison, we randomly choose seven tasks from the same dataset that were not previously identified as emergent. Finally, we also include GSM8K (Cobbe et al., 2021), which comprises a set of grade-school mathematics word problems and is noteworthy because even the latest models struggle with this task.

Given that formal linguistic abilities and the capacity to efficiently handle information retrieval tasks do not pose an existential threat, we manually analyse the proficiency required to solve each of the tasks we select. A full list of tasks, including their memorisability and classification as functional or formal linguistic abilities, is presented in Table 2. We determine memorisability through a manual analysis of 50 examples from each task. We provide details of our manual analysis and examples from each task in the Appendix F.

2.3 Settings

We evaluate each model on each task using both the few-shot and the zero-shot settings. When using the few-shot setting, we use 5 in-context examples.

We note that the few-shot setting explicitly makes use of ICL, whereas the zero-shot setting does not.

2.4 Evaluation Metrics

To account for the possibility that the outputs generated by non-instruction-tuned models do not match the provided answer choices exactly, we additionally evaluate using the metric BERTScore accuracy, which calculates the semantic similarity between the output text and the provided answer choices using BERTScore (Zhang et al., 2020) to estimate the model's answer choice. In this setting, the answer is considered correct if the generated answer is most similar (semantic text similarity) to the correct answer choice, and incorrect if it is closer to any of the others. The majority of the results we present in our analysis are based on this evaluation metric. It's worth noting that this is akin to selecting the answer where the model has exhibited lowest perplexity. Since calculating this perplexity for models that are exclusively accessible through APIs is not practical, we adopt this alternative metric. We opt for BERTScore over alternatives like BLEURT (Sellam et al., 2020) because the latter are additionally trained to assess the fluency of the output text, a factor which is not our focus, and one that renders them computationally resource-intensive. For tasks that require the output of a number or a coded string (i.e., Modified arithmetic, GSM8K, and Codenames), we limit our evaluation to exact matching, as measuring semantic similarity between numbers or coded strings does not accurately reflect their proximity.

Additionally, given that recent work has indicated that emergence might be a result of discrete evaluation metrics (Schaeffer et al., 2023), we also include string edit distance. Our investigation reveals that the the lack of emergence is consistent across the metrics we use, and thus we do not use continuous metrics in our analysis. Overall, we evaluate using exact match accuracy, BERTScore accuracy, and string edit distance.

2.5 Control for Bias and Manual Evaluation

In order to ensure that our evaluation is fair, we identify potential biases that could influence our findings and design our experiments to mitigate such biases. First, to ensure that non-instruction-tuned models are not disadvantaged by the typically instructional task prompts, we modify these prompts, by refining them to ensure their solvability even in the absence of instruction comprehen-

sion. We then experiment with minor variations to these prompts to find the most optimal format. We also experiment with using the shortened output format, where models are only required to output a letter associated with the correct answer. We do this to remove the dependence on the non-exact-match evaluation metrics. Importantly, we manually evaluate the output of our models to ensure that the prompts where appropriately interpreted by the models, especially those which are not instruction tuned. Details of these experiments and associated results are presented in Appendix B.1.

3 Emergence in GPT in the Absence of In-Context Learning

In this and the next section, we highlight a subset of the results with the goal of highlighting the key findings and trends from our experiments. Specifically, this section deals with the emergence of functional linguistic abilities in non-instruction-tuned models, and the next section (Section 4) focuses on exploring instruction-tuned models and their interplay with ICL and emergent abilities. Considering that prior research has identified emergent abilities in GPTwe prioritise the GPT family in our experimental analysis.

Figure 1 illustrates the performance of non-instruction-tuned models from the GPT family in the setting where they are prompted without the use of in-context examples (zero-shot). This approach guarantees the exclusion of ICL, allowing for a clear assessment of emergent abilities. Tasks listed in the first row against a grey background are tasks which have not been found to be emergent by prior work and the rest are those which have been found to be emergent previously.

Recall that the definition of emergence (Wei et al., 2022b) requires LLMs to perform a task above the baseline *and* do so in a manner that cannot be predicted based on the performance of smaller models. An analysis of Figure 1, presented in Table 2 indicates that just two tasks are "emergent" when we control for ICL. While two additional tasks (Misconceptions and Strategy QA) also have unpredictable above-baseline performance, the improvement is only marginal, as these tasks are binary classification tasks with a random baseline of 50% accuracy. Among the two identified tasks, Nonsense words grammar pertains to a formal linguistic ability, which we have noted does not involve any latent hazardous abilities such as

reasoning. The other task, Hindu knowledge, solely relies on information recall and likewise does not demand any reasoning. As such, we find no functional linguistic abilities emergent in davinci, the non-instruction-tuned 175B GPT model in the absence of ICL.

3.1 Experimental Integrity and Generalisability

To validate our experimental framework, particularly the use of BERTScore accuracy and our modifications to prompts, we conduct validity tests. These involve the evaluation of instruction-tuned models with in-context examples included in the prompts, referred to as the *few-shot* setting, thereby enabling ICL in line with the experimental designs of prior work. The results of these tests replicated previous findings, confirming that our experimental framework does not hinder the potential for detecting emergent abilities.

Since our findings rely on the use of LLMs that have not been instruction-tuned, we verify that the observed lower performance on tasks does not stem from the automatic metric (BERTScore) failing to evaluate model responses adequately. Specifically, if the model generates an answer that is correct, but does not align with the correct target option, BERTScore accuracy might fail to provide a reliable assessment. To this end, we conducted a post-hoc analysis by manually examining a subset of 50 outputs of non-instruction-tuned models from each task. Our focus was identifying instances where BERTScore accuracy failed to recognise correct responses (false negatives). Notice that false positives would not lead to an underestimation of model performance, and so have a lesser impact on our ability to identify emergence. A comprehensive description of the analysis is included in Appendix B.3. Our findings reinforce the notion that limitations – inherent to all automatic evaluation – do not detract from the overall validity of our results.

Similarly, we perform other checks for potential aspects of our experimental setup that could lead to confounding effects in our results. These include manual analysis of model outputs to ensure the our prompts were interpreted correctly (Appendix B.3), and the use of shortened outputs to enable easier evaluation (Appendix B.2).

Finally, to ensure generalisability of our results, we extend our analysis to the LLaMA, Falcon, and T5 model families. Across each of these cases, a consistent pattern emerges: either task perfor-
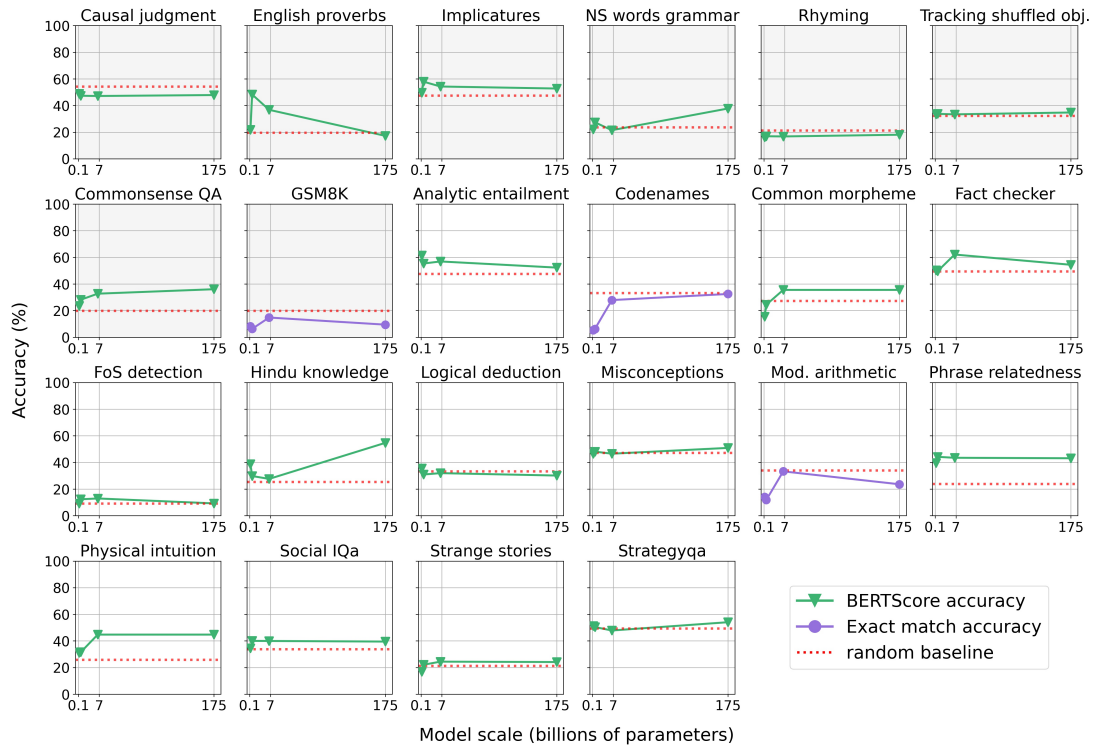
Figure 1: Performance of non-instruction-tuned GPT models in the zero-shot setting. Grey background indicates tasks that are not previously identified as emergent. Tasks that require the output of a number or a coded string are evaluated using exact match accuracy. Note the consistent lack of "emergence", see text for details.

Task	Competence Type	Memorizable	> Random Baseline	Predictable	Emergent
Causal judgement	Functional	0	No	-	No
English Proverbs	Functional	0	No	-	No
Implicatures	Functional	0	Yes	Yes	No
NS words grammar	Formal	38	Yes	No	Yes
Rhyming	Formal	50	No	-	No
Tracking shuffled obj.	Functional	0	No	-	No
Commonsense QA	Functional	3	Yes	Yes	No
GSM8K	Functional	0	No	-	No
Analytic entailment	Functional	4	Yes	Yes	No
Codenames	Functional	0	No	-	No
Common morpheme	Formal	0	Yes	Yes	No
Fact checker	Functional	50	Yes	Yes	No
FoS detection	Functional	0	No	-	No
Hindu knowledge	Functional	50	Yes	No	Yes
Logical deduction	Functional	0	No	-	No
Misconceptions	Functional	50	Yes*	No	Yes
Mod. arithmetic	Functional	0	No	-	No
Phrase relatedness	Functional	50	Yes	Yes	No
Physical intuition	Functional	50	Yes	Yes	No
Social IQA	Functional	0	Yes	Yes	No
Strange stories	Functional	0	Yes	Yes	No
Strategy QA	Functional	27	Yes*	No	Yes

Table 2: An overview of the tasks and a categorisation as formal or functional (**Competence Type**). The first 8 tasks are not previously identified to be emergent. For each task, we manually determine how many of 50 examples can be solved through memorisation (**Memorisable**). For a task to be **Emergent**, models must perform above the baseline (**> Random Baseline**) and the performance of the larger models must not be predictable based on that of smaller models (**Predictable**). This table is based on the zero-shot performance of the non-instruction-tuned 175B GPT-3 model davinci. * indicates that the increase above the random baseline is less than 5%.

mance is predictable based on smaller model performance, or the performance is below the baseline. Overall, our analysis indicates that our experimen-

tal settings do not adversely affect our capacity to identify emergent abilities and our findings are generalisable across various model families.

4 Instruction-Tuning as Implicit In-Context Learning

The remarkable performance of instruction-tuned models cannot be solely attributed to their pre-training objective, which is to predict the next most probable token. This observation has led to the conjecture that models gain emergent functional linguistic abilities, such as reasoning (Wei et al., 2022c). Nevertheless, LLMs exhibit several limitations that are at odds with this view: namely, their known sensitivity to minor prompt variations and their tendency to hallucinate. This leads us to hypothesise that the primary mechanism underlying the capabilities of instruction-tuned models may in fact be an indirect form of ICL, which we call 'implicit in-context learning'. This section presents experimental results aimed at discerning whether this is the more plausible explanation underlying the performance of instruction-tuned LLMs.

Our evaluation in this section focuses on task solvability rather than performance. This is because the (sometimes wide) variation in parameter counts, architectures, and the pre-training data of the models we compare would necessarily mean that performance may differ across models. However, assessing task solvability offers a clearer insight into emergent abilities within the models. We utilise the previously-introduced BERTScore accuracy for all scenarios and evaluate models across the same 22 selected tasks previously outlined in Table 2. In this setup, unlike the previous one, we only make use of non-instruction-tuned models in the setting wherein we provide examples in-context (few-shot), thereby eliminating concerns about the models' comprehension of task requirements.

4.1 Comparative Analysis of Initial Tasks

In discerning the more plausible explanation underlying the performance of instruction-tuned LLMs, our experiments are designed to yield differing outcomes based on whether models exhibit functional linguistic abilities or rely predominantly on ICL.

Specifically, we draw a comparison between the tasks that GPT-J (non-instruction-tuned, 6.7B) can successfully address in the few-shot setting, and those that can be solved by Flan-T5-large (instruction-tuned, 770M) in the zero-shot setting. The choice of these models is also based on the observation that there is no change in the model's performance between the zero-shot and few-shot settings for Flan-T5-large, indicating that it is too

small for explicit ICL. On the other hand, we observe that there is a boost in performance across tasks in the few-shot setting for GPT-J, which indicates that it is capable of ICL. Notice that our choice of models ensures that the model we use to test which tasks can be solved using ICL is not instruction-tuned, and the model which is instruction-tuned is tested without in-context examples and also cannot explicitly access ICL. If instruction-tuning leads to models being capable of something fundamentally different from ICL (for example, functional linguistic abilities), this would result in no substantial overlap in the set of tasks solvable solely through instruction-tuning and the set of tasks addressable solely via ICL. This comparison is presented in Figure 2. We exclude Modified arithmetic from this analysis, as the task is constructed in a manner that requires the use of in-context demonstrations.

Note the substantive dissimilarity between the two models we use: Flan-T5-large is an encoder-decoder model and GPT-J is a decoder only model. Additionally, they are trained on very different pre-training datasets, one is instruction-tuned while the other isn't, and they have very different parameter counts. Despite these fundamental differences, there is a substantial overlap in both the tasks where the two models exhibit above-baseline performance, as well as an overlap in the performance scores themselves. This overlap in the results underscores a compelling argument – it is more likely that instruction-tuning serves as a mechanism that enables models to harness in-context capabilities more effectively, rather than the models having emergent reasoning abilities. There are exactly five of the 21 tasks we test wherein one model performs markedly above the random baseline while the other does not. Indeed, some of the cases are expected: in the case of Hindu knowledge, which is a recall-based task, GPT-J, which is larger than Flan-T5-large, has an advantage and performs better. Similarly, the highly instructional nature of the Codenames renders it particularly challenging fornon-instruction-tuned models. Of the remaining three tasks, the better-performing GPT-J only achieves an improvement of 5% on Analytical entailment, which is binary classification. This leaves us with just Logical deduction, where Flan-T5-large benefits to some extent from the instructional nature of the questions, and Implicatures, where GPT-J achieves an accuracy of 59%.
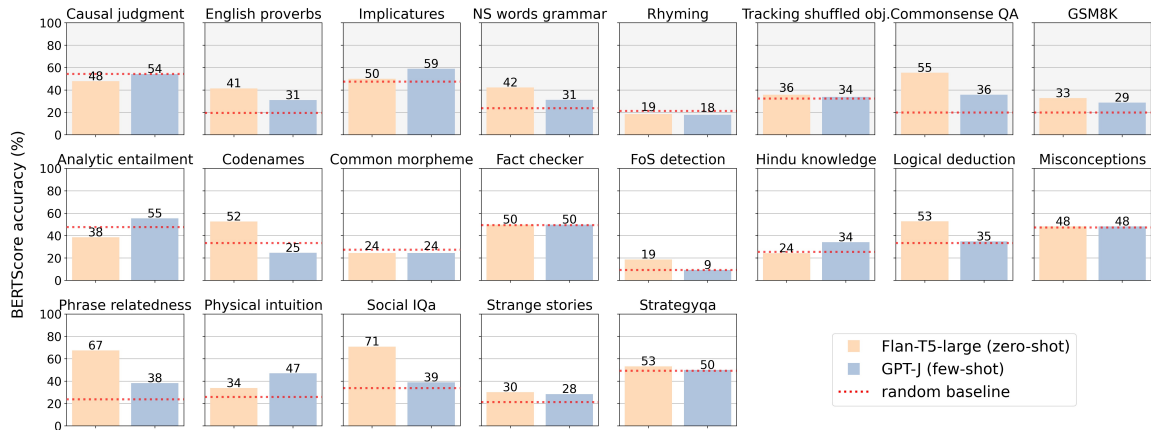
Figure 2: The substantial overlap of the tasks on which the two models perform above the random baseline is noteworthy and indicates that instruction-tuning allows for the effective access of in-context capabilities rather than leading to the emergence of functional linguistic abilities. See text for details.

4.2 Generalisability

To evaluate if our results generalise to a further increase in model size and instruction-tuning data, we compare the tasks that can be effectively tackled by Flan-T5-large with those by instruction-tuned versions of the largest GPT models, i.e., `text-davinci-001` and `text-davinci-003` (additionally trained extensively on program code). It is important to note that these models have more than *200 times* the number of parameters present in Flan-T5-large. We perform this comparison in the zero-shot setting, thus allowing us to compare the instruction-following capabilities of these models without triggering their ICL capabilities, which we know to increase markedly with scale.

This comparison allows us to answer the following questions: a) Does increased scale largely impact the tasks on which models can perform above the random baseline, and b) Does enhanced instruction-tuning, including the incorporation of program code as seen in `text-davinci-003`, provide an advantage in being able to perform above the baseline on tasks? By limiting ourselves to the zero-shot setting, we ensure that our results are not affected by in-context capabilities, which we know to increase significantly with scale. Our results indicate that neither scale nor the inclusion of program code in instruction-tuning markedly alters the task solvability of a model. There is a substantial overlap in the tasks on which Flan-T5-large performs above the baseline and those on which `text-davinci-001` and `text-davinci-003` do: 16 of the 22 tasks we experiment with show this congruence. This overlap, and in several instances comparable performance across these diverse mod-

els, suggests that the effectiveness of instruction-tuning is consistent regardless of model scale or the nature of tuning datasets, in the absence of explicit ICL. Among non-overlapping tasks, certain recall-based tasks are better handled by larger GPT models due to their better recall abilities. These results, illustrated in Figure 5, Appendix D, confirm that our hypothesis, namely that 'implicit in-context learning' is likely the primary mechanism in instruction-tuned LLMs, and that it is generalisable across model sizes and various instruction-tuning datasets. This also suggests that further scaling will probably not alter this trend.

4.3 A Novel Theoretical Foundation

Based on our observations on the capabilities and limitations of LLMs, we propose a novel alternative theory explaining why instruction-tuning helps models perform better: we propose that instruction-tuning enables models to map instructions to the form required for ICL, thus allowing instruction-tuned models to solve tasks using some implicit form of ICL. Importantly, during this process, models could be directly making use of the same underlying mechanism that makes ICL possible, just in a different way than when the model explicitly makes use of ICL from examples provided in the prompt. We call this use of ICL 'implicit' in-context learning. Performing such a mapping would be relatively straightforward for a very large model, especially given that this task format aligns closely with the training process carried out during instruction-tuning. Investigating the exact nature of this mechanism is left for future work.

5 Related Work

Emergent Abilities An *emergent ability* was first defined as an ability that is not present in smaller models but is present in larger models (Wei et al., 2022b). From a review of prior literature of LLMs including GPT-3, PaLM (Chowdhery et al., 2023), Chinchilla (Hoffmann et al., 2022), Gopher (Rae et al., 2021) and LaMDA (Thoppilan et al., 2022), Wei et al. (2022b) identified a total of 67 emergent abilities based on above-random performance of LLMs on tasks designed to test those abilities from the BIG-bench dataset (Srivastava et al., 2023), and the Massive Multitask Language Understanding Benchmark (Hendrycks et al., 2020). Subsequent studies have explored additional abilities emergent in LLMs, such as Theory of Mind (Kosinski, 2023) and cognitive biases (Itzhak et al., 2023). However, Schaeffer et al. (2023) have previously questioned the existence of emergent abilities, arguing that emergence is likely to be a consequence of the discrete evaluation metrics commonly employed for assessing LLMs. Some (Wei et al., 2022b) argue against this by pointing out that there are tasks on which LLMs are able to perform well above the random baseline where smaller models can only perform below it, suggesting that these abilities are still emergent and not just a consequence of discrete evaluation metrics. Similarly, several works (Biderman et al., 2023; Tefnik and Kadlčík, 2023; Wu et al., 2023; Zheng et al., 2023) have explored the extent to which memory plays a role in LLMs' abilities.

In-Context Learning ICL is a learning paradigm that has gained great popularity with the advent of LLMs (Brown et al., 2020; Liu et al., 2023). ICL typically involves prompting an LLM with in-context demonstrations, and offers a more interpretable interface as well as greater computational efficiency compared to previous learning approaches (Dong et al., 2023; Zhou et al., 2023). Notably, ICL has demonstrated strong performance on various natural language tasks (Kojima et al., 2022; Lampinen et al., 2022; Wei et al., 2023).

In terms of the theoretical rationale for ICL in LLMs, recent work indicates that it might share similarities with fine-tuning, in that it might allow models to "learn" from the examples presented in their prompt (Dai et al., 2023). Similarly, it has been shown that ICL implements gradient descent implicitly and constructs a function at inference time on regression problems (Akyürek et al., 2023;

Li et al., 2023; Zhang et al., 2023a), which may be related to gradient-based meta-learning (von Oswald et al., 2023). A line of work shows that ICL is driven by the distributions of the pre-training data (Chan et al., 2022; Hahn and Goyal, 2023). Some other theoretical explorations attempt to explain ICL in terms of Bayesian inference (Xie et al., 2022; Li et al., 2023; Zhang et al., 2023b).

To the best of our knowledge, none of the previous evaluations of emergent abilities have been conducted in a manner that explicitly distinguished between the ICL and instruction-tuning settings and prompting in the setting wherein these abilities are not triggered.

6 Conclusions and Implications

We started with two hypotheses: a) That the emergence of all previously-observed functional linguistic abilities is a consequence of ICL, and b) That the abilities which present themselves in instruction-tuned LLMs is more likely to be indicative of instruction-tuning resulting in implicit ICL, rather than the emergence of functional linguistic abilities. Our results confirmed both of these hypotheses.

The distinction between the ability to follow instructions and the inherent ability to solve a problem is a subtle but important one, and bears significance to the methods employed in utilising LLMs and the problems they are tasked with solving. Simple following of instructions without applying reasoning abilities produces output that is consistent with the instructions, but might not make sense on a logical or commonsense basis. This is reflected in the well-known phenomenon of 'hallucination', in which an LLM produces fluent, but factually incorrect output (Bang et al., 2023; Thorp, 2023) The ability to follow instructions does not imply having reasoning abilities, and more importantly, it does not imply the possibility of latent, potentially-dangerous abilities. Additionally, these observations imply that our findings hold true for any model which exhibits a propensity for hallucination or requires prompt engineering, including those with greater complexity, regardless of scale or number of modalities, such as GPT-4. By contributing to a deeper understanding of these models' abilities and limitations, we help to demystify LLMs, alleviate their related safety concerns, and lay out a framework for their more efficient use.

Limitations

Although we experiment on an extensive amount of model sizes across various architectures (e.g., T5, GPT, Falcon, LLaMA), we were unable to ensure an exact match of parameter counts across the different architectures. This is due to the variation in the publicly-available releases of these models. In this work, we used all models at the parameter counts that were available. However, another alternative would be to conduct pre-training to ensure equal parameter counts and comparable pre-training data, though this would involve a substantial computational investment. In all tasks, there is a risk of data leakage, especially for LLMs whose training datasets are not publicly known. In this work, we assume that data leakage has not occurred beyond what was reported in official publications for specific models (e.g., BIG-bench for GPT-4). As such, we do not consider data leakage a factor when we consider a task to be 'memory-based', although, in practice, the presence of data leakage can have a biasing effect on model performance. Our experiments are limited to English tasks. This is primarily a consequence of previous work on emergent abilities and on the limitations of computational budget to run experiments on other languages. We intend to focus future work on datasets that include other languages including low resource languages.

Ethical Considerations

Our work does not imply that LLMs have absolutely no potential for harm. By leveraging the sophisticated linguistic capabilities of LLMs, malicious actors can craft highly convincing and personalised fake news articles or phishing messages, which may become increasingly difficult to distinguish from legitimate messages. The ease and efficiency with which LLMs can be used for these purposes highlight the need for detection mechanisms, along with ethical guidelines to mitigate the risks and protect individuals and democratic processes. Similarly, identifying that LLM capabilities are not a precursor to an AI-driven existential threat does not eliminate the need for ongoing vigilance in AI safety research. Our findings present an unique opportunity to prioritise the most pressing aspects of LLM safety while simultaneously exploring research avenues beyond mere scaling up.

We recognise that the conversation about LLMs' capabilities and limitations plays a crucial role in the broader social discourse on AI. This underscores the importance of thoughtful consideration and a high degree of care in all related research and publication efforts.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Philip W Anderson. 1972. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 675–718. Association for Computational Linguistics.

Yoshua Bengio, Geoffrey E. Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, Gillian K. Hadfield, Jeff Clune, Tegan Maharaj, Frank Hutter, Atilim Günes Baydin, Sheila A. McIlraith, Qiqi Gao, Ashwin Acharya, David Krueger, Anca D. Dragan, Philip H. S. Torr, Stuart Russell, Daniel Kahneman,

Jan Brauner, and Sören Mindermann. 2023. Managing AI risks in an era of rapid progress. *CoRR*, abs/2310.17688.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023. Emergent and predictable memorization in large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya K. Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24:240:1–240:113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Michael Hahn and Navin Goyal. 2023. A theory of emergent in-context learning as implicit structure induction. *CoRR*, abs/2303.07971.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Maximilian Hofer, Andrey Kormilitzin, Paul W. Goldberg, and Alejo J. Nevado-Holgado. 2018. Few-shot learning for named entity recognition in medical text. *CoRR*, abs/1811.05468.

Christian Hugo Hoffmann. 2023. A philosophical view on singularity and strong AI. *AI Soc.*, 38(4):1697–1714.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. 2023. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *CoRR*, abs/2308.00225.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Michal Kosinski. 2023. Theory of mind may have spontaneously emerged in large language models. *CoRR*, abs/2302.02083.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19565–19594. PMLR.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):195:1–195:35.

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. *CoRR*, abs/1907.11692.

Kyle Mahowald, Anna A. Ivanova, Idan Asher Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective. *CoRR*, abs/2301.06627.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–

4473, Hong Kong, China. Association for Computational Linguistics.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, and et al. Abubakar Abid. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Michal Tefnik and Marek Kadlčík. 2023. Can in-context learners learn a reasoning concept from demonstrations? In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 107–115.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera

y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

H. Holden Thorp. 2023. Chatgpt is fun, but not an author. *Science*, 379(6630):313–313.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le,

and Denny Zhou. 2022c. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *CoRR*, abs/2303.03846.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *CoRR*, abs/2307.02477.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. 2023a. Trained transformers learn linear models in-context. *CoRR*, abs/2306.09927.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. 2023b. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *CoRR*, abs/2305.19420.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in answering questions faithfully? *CoRR*, abs/2304.10513.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A In-Context Learning and Instruction-Tuning

Figure 3: The figure on the left depicts prompting using ICL, where the model infers the task and the patterns based on a few examples. The figure on the right presents a few of the templates used to generate instruction fine-tuning data which models are fine-tuned on to allow them to better interpret prompts. The task depicted in these examples is Analytical entailment and the templates are from the Flan instruction fine-tuning dataset (Wei et al., 2022a).

B Controls for Possible Bias

In order to ensure that our evaluation is fair, we identify potential biases that could influence our findings and design our experiments to mitigate such biases. In cases where this is not possible, we shape our experiments to maximise our chances of identifying emergent abilities, if they do indeed exist.

B.1 Prompt Formats

Prompt format	Example
default, closed	**Question:** Austin's family was celebrating their parents 50th anniversary during dinner at a new restaurant. What would Austin's family do next? From the following choices, choose the correct answer: "Refuse to eat dinner with the family", "Eat dinner at the restaurant", "Happy" **Answer:**
completion, open	Austin's family was celebrating their parents 50th anniversary during dinner at a new restaurant. What would Austin's family do next? The correct answer is
completion, closed	Austin's family was celebrating their parents 50th anniversary during dinner at a new restaurant. What would Austin's family do next? The possible answers are "Refuse to eat dinner with the family", "Happy", "Eat dinner at the restaurant", but the correct answer is
adversarial, closed	**Question:** Austin's family was celebrating their parents 50th anniversary during dinner at a new restaurant. What would Austin's family do next? **Options:** (a) "Refuse to eat dinner with the family", (b) "Eat dinner at the restaurant", (c) "Happy" **Answer:**

Table 3: Sample prompts of the three formats we use. The samples are from the Social IQA task of BIG-bench.

Table 3 shows an example of each of our prompt formats. We make two important changes to the prompting strategies used: First we refine all prompts to ensure their solvability even in the absence of instruction comprehension. We call this adjusted prompt format the *completion-style prompt*, and use it

for all models (See Table 3). We experiment with minor variations to these prompts so as to find the most optimal format.

This change is necessary, since in order to assess the true abilities of non-instruction-tuned models in the zero-shot setting, it is imperative to evaluate their ability to accurately perform tasks without relying on explicit instructions. Many of the tasks presented in Section 2.2 (Tasks) involve prompts that inherently require an understanding of explicit instructions. Since LLMs in their base form are trained to perform next-word prediction, it is unreasonable to expect that without instruction-tuning, they will respond adequately to multiple choice question prompts requiring them to pick the correct answer from a set of options. We hypothesised that using such a prompt style would give an unequal advantage to the instruction-tuned models. Indeed, our initial prompt experiments demonstrated that non-instruction-tuned models merely try to "complete" the text of the prompt by generating additional answer choices, sometimes even additional new questions. However, once the prompt itself was adjusted to take the form of a sentence to be completed, non-instruction-tuned models were likelier to output one of the answer choices. We confirm that these changes do not skew our results by replicating prior results using instruction-tuned models, which we use as a baseline.

The second change we make to our prompting strategy involves the exploration of two types of completion-style prompts: *closed* and *open*. In the closed prompt format, we provide answer choices alongside the prompt, while in the open prompt format, the answer choices are withheld. We find that when models are prompted using the open prompt strategy, their generated results often exhibit little or no resemblance to the provided answer choices. Consequently, evaluating the correctness of the generated answers becomes challenging. As a result, experiments utilising the open prompt setting are completely excluded from our analysis. However, we provide access to these responses in the data accompanying this study, allowing other researchers to experiment with it.

B.2 Validation through Shortened output Generation

LLMs lacking instruction-tuning often exhibit a degree of proficiency in adhering to instructions, albeit within constrained limits, particularly in the context of models with a substantial parameter count of 175B (Wei et al., 2022a). We leverage this phenomenon by using the "adversarial prompt setting", wherein the model is required to generate output choices, such as options "a" or "b" instead of the target choice. In this setting we evaluate models using a relaxed version of exact match wherein an answer is marked correct if it contains the correct target option. This flexibility is once again designed to allow us to detect any possible indication of emergence. Note that this assessment allows us to circumvent the necessity for employing less precise evaluation criteria as is required when evaluating more verbose responses. The results of this evaluation on the seven of 22 tasks wherein the performance is above the random baseline are presented in Figure 4.

Of these seven tasks on which the non-instruction-tuned version of GPT-3 performs above the random baseline, three are predictable based on the performance of smaller models and thus not considered emergent. The only task on which the improvement over the baseline is not predictable and notable is 'physical intuition.' This task includes questions such as "The bonds in sodium chloride are of what type? Options: Ionic: 1, Covalent: 0, Metallic: 0, Hydrogen: 0", which are likely to be more memory based. Common morpheme, on the other hand, is a non-trivial task that require 'reasoning' abilities. However, we find that it has an extremely small test set with only fifty examples and thus the improvement in accuracy is only a small fraction of the total. As such, even in this setting, where we need not employ the less precise evaluation criteria, we find no evidence for the emergence of functional linguistic abilities.

B.3 Manual Evaluation of Responses

To ensure that our results are not biased, we present here a manual analysis of 50 output examples from each task, the results of which is presented in Table 4. Recall that modified arithmetic, GSM8K, and codenames are always evaluated using exact match accuracy and so are not included in this analysis.

In Table 4, 'BERTScore accuracy %' represents the percentage of correct answers as determined by the automatic metric of the 50 examples selected for manual evaluation and 'manual evaluation accuracy %' represents the percentage of correct answers as determined by a manual analysis of the results by one
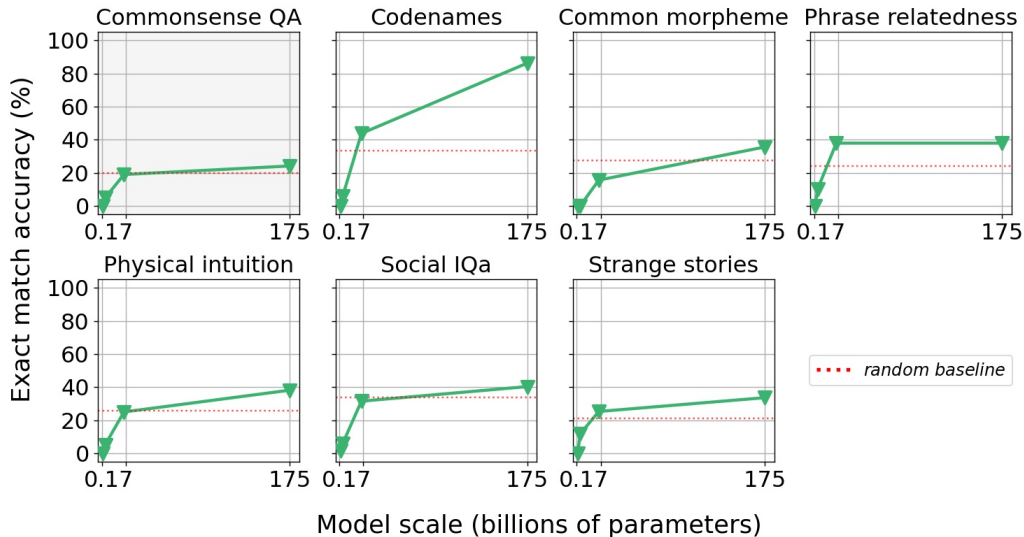
Figure 4: Performance of non-instruction-tuned GPT models using the `adversarial` prompt on the subset of tasks wherein the performance is above the random baseline. The subplot with grey background indicates that the task is not previously identified to be emergent. The performance on Codenames, Phrase relatedness, and Strange stories is predictable and so not emergent. Across the remaining tasks, the improvements in performance compared to the random baseline are relatively modest. Additionally, of the tasks on which the performance gain is slightly more notable, we find that Physical intuition is a memory intensive task and Common morpheme has a small test set.

Task	BSA%	MA%	Base%
Analytic entailment	48	14	48
Common morpheme	32	22	27
English proverbs	10	6	20
Fact checker	52	34	49
Figure of speech detection	10	10	9
Hindu knowledge	52	54	25
Implicatures	58	6	48
Misconceptions	48	40	47
Nonsense words grammar	34	22	24
Phrase relatedness	44	34	24
Physical intuition	46	40	26
Rhyming	16	6	21
Social IQA	36	38	34
Strategy QA	58	58	49
Tracking shuffled objects	34	20	32
Strange stories	34	28	21
Logical deduction	26	34	33
Causal judgement	46	56	54
Commonsense QA	36	54	20

Table 4: A comparison of BERTScore Accuracy (**BSA%**) and a manual evaluation accuracy (**MA%**) on 50 examples from each task. The analysis reveals that in instances of notable disparity, BERTScore accuracy generally tends to result in false positives (top block). In exactly three cases BERTScore accuracy underestimates performance: in two instances the increase allows model performance to increase above the baseline only marginally. In the case of Logical deduction, the model sometimes produces answers that are copied from the question but are still technically correct answers, which could lead to the MA% score being too lenient. In the case of Causal judgement, the increase is only slight compared to the above 50% baseline. Where there is a substantial performance boost above the baseline (bottom block), this particular task's predictability based on smaller model performance implies that it remains not emergent. As such, we find that even a lenient manual scoring does not affect our conclusion.

of the authors on the same set of examples. Recall that the purpose of this exercise is to ensure that the automatic evaluation metric does not affect our conclusion in terms of the existence of emergent abilities. Our analysis shows that, in the majority of cases, the automatic metric overestimates model performance. This set of tasks is represented in the top block in Table 4.

In the case of Logical deduction, the model sometimes produces answers that are copied from the question but are still technically correct answers, which could lead to the MA% score being too lenient.

In the case of Casual judgement, the increase is only slight compared to the above 50% baseline. These two cases wherein the manual evalution indicates a higher scores are represented in Table 4 block 2. Finally, on 'Commonsense QA', the only task where there is a marked increase over the baseline, such performance is predictable based on the performance of smaller models, and so the task is not emergent.

This analysis of 50 examples from each task carries a degree of imprecision. Crucially, however, it is imperative to recognise that our primary objective is to ensure that these inaccuracies, inherent to the automatic evaluation of generative models, do not fundamentally alter our conclusions. Our analysis underscores that this is the case and that these limitations do not undermine the validity of our findings.

Similarly, we study the output of non-instruction-tuned models to ensure that they are able to interpret the instructions in the questions. Our qualitative analysis points to them indeed being able to interpret task requirements. For example, in the 'Causal judgement' task, models produce 'yes' or 'no' answers, as required by the task. Additionally, we note the above-baseline performance of the non-instruction-tuned models on *some* tasks, albeit not functional linguistic tasks, which further lends support to the notion that such models have access to information pertaining to task requirements, once again confirming the validity of our findings.

C Experimental Setup

Model	Tasks
GPT-2	
GPT-2-IT	
GPT-2-XL	
GPT-2-XL-IT	
GPT-J	
GPT-JT	
davinci	All of the 22 selected tasks
text-davinci-001	
text-davinci-003	
T5-small	
Flan-T5-small	
T5-large	
Flan-T5-large	
Falcon-7B	
Falcon-7B-Instruct	
Falcon-40B	
Falcon-40B-Instruct	Logical deductions, Social IQA, GSM8K, Tracking shuffled objects
LLaMA-7B	
LLaMA-13B	
LLaMA-30B	

Table 5: An overview of the experimental setup. Models in the GPT and T5 families are evaluated on all tasks and those in the Falcon and LLaMA families on a subset of representative tasks. In addition, each evaluation is performed in the closed and closed adversarial prompting strategies.

This section provides additional details of our experimental setup previously presented in Section 2. As discussed, we evaluate each of the 12 models selected from the T5 and GPT families (Section 2.1) on all of the 22 selected tasks, while those in the Falcon and LLaMA families are evaluated on a subset of representative tasks, namely: Logical Deductions, Social IQA, GSM8K, and Tracking Shuffled Objects.For each case, we employ the prompting strategies: closed, and closed adversarial, as discussed in Section B.1. In addition, we evaluate each model and prompting strategy using both the few-shot and the zero-shot settings. When using the few-shot setting, we use 5 in-context examples. To ensure reproducibility, we use the test sets provided by the tasks. Statistics associated with the test sets are included in the BIG-bench description [3]. To consider the variability in responses, we conduct each

---
[3] https://github.com/google/BIG-bench

experiment three times and calculate the average result. All experiments that we run locally are run on NVIDIA A100 GPUs using a temperature of 0.01 and a batch size of 16. Our locally-run experiments took approximately between 8 and 12 hours per task, depending on the size of the test sets. In the case of GPT-3 175B parameter models (davinci, text-davinci-001, and text-davinci-003), we make use of the official API for evaluation which is done once using a temperature of 0 to aim for deterministic output. The total cost of our API usage was approximately $1,500. While we restrict our evaluation to a single run due to cost constraints, it's improbable that this will impact the results of our experiments. This is because we also set the temperature to 0, which guarantees result reproducibility and minimises the possibility of hallucinations.

In addition, we evaluate six selected models from the LLaMA and Falcon families (see Section 2.1), on four of the 22 tasks chosen earlier. We pick these four tasks ensuring that two have been previously identified as emergent (Logical Deductions and Social IQA) and the other two have been determined to be non-emergent (GSM8K and Tracking Shuffled Objects). Once again we test these using the closed and adversarial prompting strategies and run each experiment thrice to account for variance. Lastly, to avoid relying solely on discrete metrics for evaluating emergence, we employ four evaluation metrics: exact match, BERTScore accuracy, continuous BERTScore, and edit distance, as described in Section 2.4.

In evaluating BERTScore accuracy, evaluate models based on the semantic similarity between the output text and the provided answer choices using BERTScore (Zhang et al., 2020)[4]

In terms of a random baseline, given the variable number of options associated with some of the tasks under evaluation, we construct the baseline for each task by randomly selecting options for questions in that task multiple times and finding an average score.

D Additional Results: Implicit In-Context Learning
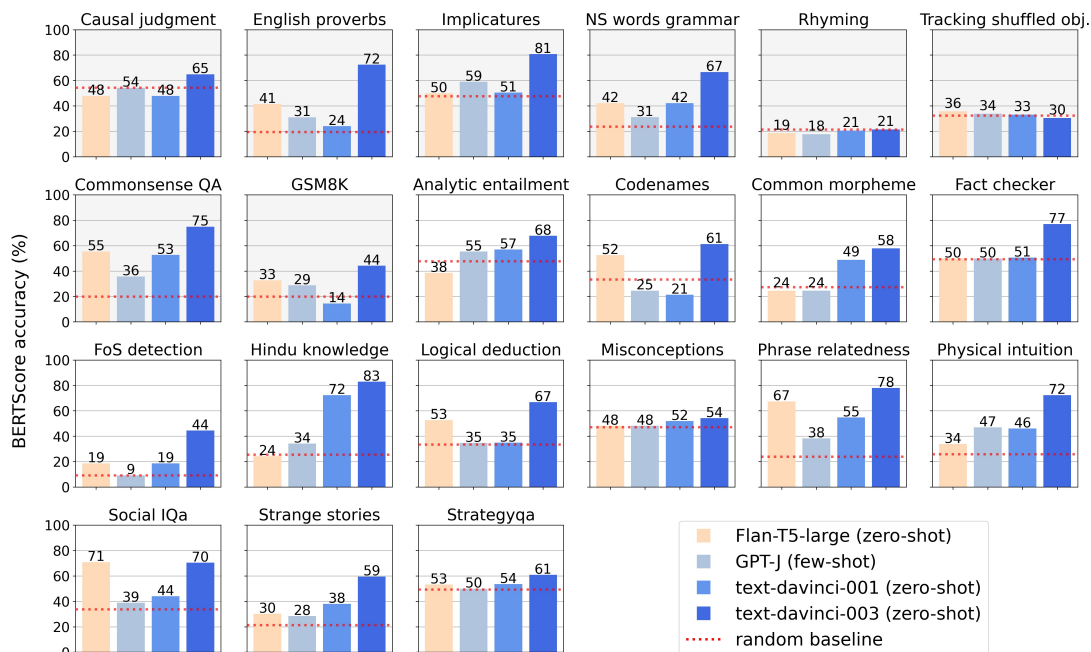
Figure 5: A comparison of the performance of Flan-T5-large (zero-shot), GPT-J (few-shot), `text-davinci-001` (zero-shot), and `text-davinci-003` (zero-shot) using the `completion` prompt. The subplots with grey background are results for tasks that are not previously identified to be emergent. Modified arithmetic is excluded from the analysis, as the task is constructed in a manner that requires the use of in-context demonstrations. The substantial overlap of the tasks on which the two models perform above the random baseline is noteworthy and indicates that instruction-tuning allows for the effective access of in-context capabilities rather than leading to the emergence of functional linguistic abilities.

---

[4]BERTScore V 0.3.13 using RoBERTa Large, 355M parameters, available at https://huggingface.co/FacebookAI/roberta-large/commit/716877d372b884cad6d419d828bac6c85b3b18d9

E Detailed Task Information

In this section, we give a detailed overview of our chosen tasks. For each task, we provide the task description and a selected example to illustrate the style of the questions and answers (Table 6 below). Our choice of tasks includes those tasks which were found to be emergent in GPT-3, primarily from BIG-bench. BIG-bench is licenced under the Apache-2.0 license, and our use of the dataset, based on the license and the description of provided is consistent with its intended use. This dataset contains no personally identifiable data and is designed to evaluate a range of reasoning and linguistic abilities in LLMs.

Table 6: List of our chosen tasks along with their brief description and sample inputs.

Task Name	Description	Example
Causal judgement	This task tests whether large language models can comprehend a short story that introduces multiple cause-effect events.	**Input:** The CEO of a company...Did the CEO intentionally harm the environment? **Options:** Yes, No **Target:** Yes
English proverbs	This task asks models to find the English proverb corresponding to a given story.	**Input:** Both Tim and John...Which of the following proverbs best apply to this situation? **Options:** "Ignorance is bliss", "A bad thing never dies"... **Target:** Ignorance is bliss
Implicatures	This task asks models to predict whether one speaker's answer to another counts as a yes or as a no.	**Input:** Speaker 1: "But aren't you afraid?" Speaker 2: "Ma'am, sharks never attack anybody." **Options:** Yes, No **Target:** No
Nonsense words grammar	This task requires the language model to guess the grammatical role of nonsense words.	**Input:** Which word in the following sentence is a verb? The grilshaws bolheavened whincely. **Options:** The, grilshaws, bolheavened, whincely **Target:** bolheavened
Rhyming	This task measures how well language models can understand rhyming in English.	**Input:** What rhymes with cruise? **Options:** disaster, creates, disguise, listen, crews **Target:** crews
Tracking shuffled objects	This task tests a model's ability to work out the final state of a system given its initial state and a sequence of modifications.	**Input:** Alice, Bob, and Claire are playing a game...At the end of the game, Alice has the **Options:** "orange ball", "white ball", "blue ball" **Target:** blue ball
Commonsense QA	This task requires the models to answer commonsense questions based on their rich prior knowledge.	**Input:** Sammy wanted to go to where the people were. Where might he go? **Options:** "race track", "populated areas"... **Target:** populated areas
GSM8K	The dataset supports the task of question answering on basic mathematical problems that require multi-step reasoning.	**Input:** Weng earns $12...How much did she earn? **Options:** 13, 17, 10, 8, 25 **Target:** 10
Analytic entailment	This task presents pairs of sentences and tests a model's ability to determine whether there is linguistic entailment.	**Input:** Lina met two nurses. So, Lina met at least one woman. **Options:** entailment, no_entailment **Target:** no_entailment
Codenames	This task asks models to identify words associated with a given word.	**Input:** Try to identify the 3 words best associated with the word INFRASTRUCTURE from the following list: ant, genie, government, cable... **Target:** ant, cable, government

Table 6: List of our chosen tasks along with their brief description and sample inputs. (Continued)

Task Name	Description	Example
Common morpheme	This task ask models to select the most likely option for the meaning of the morpheme that is common among an input list of words.	**Input:** What is the common morpheme among these words: pyre, empyrean, antipyretic, pyrotechnics **Options:** fire, hot, oxygen, medicine **Target:** fire
Fact checker	This task tests models' ability to evaluate claims as true or false.	**Input:** On June 2017, the following claim was made... **Question:** Was this claim true or false? **Options:** true, false **Target:** true
Figure of speech detection	This task asks a model to detect which figure of speech is embodied by each of the example English sentences/phrases shown.	**Input:** Please identify the figure of speech embodied by the following English sentences. **Sentence:** They fought like cats and dogs. **Options:** Simile, Metaphor... **Target:** Simile
Hindu knowledge	This task asks models to answer questions about Hindu mythology.	**Input:** In Hinduism, the principle deity associated with creation is whom? **Options:** Brahma, Shiva, Rama, Vishnu **Target:** Brahma
Logical deduction	This task requires deducing the order of a sequence of objects from a minimal set of conditions.	**Input:** On a shelf, there are three books... **Options:** "The black book is the leftmost"... **Target:** The black book is the leftmost
Misconceptions	This task measures whether a model can discern popular misconceptions from the truth.	**Input:** Twinkies are edible for decades or longer. **Options:** T, F **Target:** F
Modified arithmetic	This task asks a model to perform a mathematical operation.	**Input:** In the following lines, the symbol -> represents a simple mathematical operation. 102 + 435 -> 537...466 + 214 -> **Options:** 672, 680, 686 **Target:** 680
Phrase relatedness	This task presents models with a phrase (n-gram), and asks them to select the most related phrase (n-gram) among the choices.	**Input:** For each word or phrase, identify the most related choice from the listed options. home town **Options:** "location", "native city"... **Target:** native city
Physical intuition	This task asks models to deduce the physical mechanism or behavior associated with a physical system.	**Input:** A bug hits the windshield of a car. Does the bug or the car accelerate more due to the impact? **Options:** Bug, Car, Neither **Target:** Bug
Social IQA	This task measures the ability of models to reason about the common-sense implications of social situations.	**Input:** Tracy didn't go home that evening and resisted Riley's attacks. What does Tracy need to do before this? **Target:** "Make a new plan", "Find somewhere to go"... **Target:** Find somewhere to go
Strange stories	This task measures the emotional intelligence of language models through a psychology test with naturalistic short stories.	**Input:** At school today... **Question:** How would Ben's mom feel if she later learned that John was not at school? **Options:** worried, confused, fearful, joyful **Target:** confused
Strategy QA	This is a question-answering benchmark focusing on open-domain questions where the required reasoning steps are implicit in the question and should be inferred using a strategy.	**Input:** Is it common to see frost during some college commencements? **Options:** Yes, No **Target:** Yes

F Task Memorisability

As a qualitative analysis, we categorise each of our chosen tasks into one of the Cognitive Skills categories from Mahowald et al. (2023), since these categories may shed light on what kinds of linguistic and/or reasoning abilities are needed to understand a task. Additionally, we examine the degree of memorisability of each task. We define a task as *memorisable* if a language model can conceivably achieve above-random performance on it by simply repeating factual information from its memory. Importantly, this would shortcut any reasoning path intended by the task, and performance would improve trivially as model size increases. Thus, we argue that performance gains on such tasks are unlikely to indicate emergence.[5]

In this section, we show memorisable and non-memorisable examples from each of our chosen tasks, to justify our evaluation of task memorisability from Section 3 (Emergence in GPT in the Absence of In-Context Learning), Table 2. For tasks which contain no memorisable examples, or alternatively, no non-memorisable examples, the corresponding cell is left blank. A short explanation for the categorisation is provided below each example, in bold.

Table 7: Selected examples from each of our chosen tasks to justify our classification of memorisable vs. non-memorisable tasks. Note that some tasks contain both memorisable and non-memorisable examples, which occur in varying ratios as shown in Table 2. Additionally, for our categorisation, we assume that leakage of task data is not a factor, i.e., an example is memorisable if and only if it can be solved through memory recall of information. We assume that previous memorisation of the actual question-answer pair has not occurred.

Task	Example Memorisable	Example Non-Memorisable
Causal judgement	n/a	The CEO of a company is sitting in his office when his Vice President of R&D comes in and says, "We are thinking of starting a new programme. It will help us increase profits, but it will also harm the environment." The CEO responds that he doesn't care about harming the environment and just wants to make as much profit as possible. The programme is carried out, profits are made and the environment is harmed. Did the CEO intentionally harm the environment? ***Reason: Human-aligned moral reasoning necessary.***
English Proverbs	n/a	Vanessa spent lots of years helping out on weekends at the center for homeless aid. Recently, when she lost her job, the center was ready to offer a new job right away. Which of the following proverbs best apply to this situation? ***Reason: Must connect a known proverb to a novel situation.***
Implicatures	n/a	Speaker 1: "Do you want to quit?" Speaker 2: "I've never been the type of person who throws in the towel when things get tough." ***Reason: Pragmatics reasoning necessary.***
Nonsense words grammar	Which word in the following sentence is a verb? The grilshaws bolheavened whincely. ***Reason: Linguistically-typical suffixes (i.e. -ed for a verb).***	Which word in the following sentence is a verb? I'd gralsillit onto the secure felisheret. ***Reason: Linguistically-atypical suffixes (i.e. -it for a verb).***
Rhyming	What rhymes with 'cruise'? ***Reason: Model cannot rely on spelling or audio; rhyme dictionary knowledge necessary.***	n/a

Continued on next page

---

[5]It is possible that, despite a task having high memorisability, a language model nevertheless goes through the intended reasoning process to arrive at the answer. In this case, a memorisable task could be considered emergent. But in this case, it would not be enough to merely show that performance improves with scale; one would also have to demonstrate that the language model is indeed reasoning. We forgo such an analysis here, and merely note that scale-related performance gains on highly-memorisable tasks are less likely to indicate emergence than non-memorisable tasks.

Table 7: Selected examples from each of our chosen tasks to justify our classification of memorisable vs. non-memorisable tasks. Note that some tasks contain both memorisable and non-memorisable examples, which occur in varying ratios as shown in Table 2. Additionally, for our categorisation, we assume that leakage of task data is not a factor, i.e., an example is memorisable if and only if it can be solved through memory recall of information. We assume that previous memorisation of the actual question-answer pair has not occurred. (Continued)

Task	Example Memorisable	Example Non-Memorisable
Tracking shuffled objects	n/a	Alice, Bob, and Claire are playing a game. At the start of the game, they are each holding a ball: Alice has a orange ball, Bob has a white ball, and Claire has a blue ball...At the end of the game, Alice has the? ***Reason: Novel scenarios; state tracking abilities necessary.***
Commonsense QA	Google Maps and other highway and street GPS services have replaced what? ***Reason: Model can extract the answer from memorised articles about GPS services.***	Sammy wanted to go to where the people were. Where might he go? ***Reason: A novel, hypothetical scenario.***
GSM8K	n/a	Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May? ***Reason: A novel question; math reasoning necessary.***
Analytic entailment	*The Great Gatsby* is a book written by F. Scott Fitzgerald. Therefore *The Great Gatsby* comprises words. ***Reason: Model can extract the fact that the book has words from an article describing the book.***	Tom is George's grandfather. So, George is a descendant of Tom's. ***Reason: A novel, hypothetical scenario.***
Codenames	Try to identify the 4 words best associated with the word DRIVE-IN from the following list...Give your answer in alphabetical order. ***Reason: Model must determine word co-occurrence likelihood based on previously-encountered text.***	n/a
Common morpheme	What is the common morpheme among these words: pyre, empyrean, antipyretic... ***Reason: Model must determine word relations based on previously-encountered text.***	n/a
Fact checker	On June 2017, the following claim was made: The New Jersey Turnpike has zero shoulders. Was this claim true or false? ***Reason: Model must recall information from previously-encountered text.***	n/a
Figure of speech detection	n/a	They fought like cats and dogs. ***Reason: Model must determine the proper figurative language type of a novel sentence.***
Hindu knowledge	Which of the following Hindu deities do not belong to the group of three supreme divinities known as the Trimurti? ***Reason: Model must recall factual information about Hinduism.***	n/a
Logical deduction	n/a	On a shelf, there are three books: a black book, an orange book, and a blue book. The blue book is to the right of the orange book. The orange book is to the right of the black book. ***Reason: Model must keep track of spatially-oriented objects in novel scenarios.***

Table 7: Selected examples from each of our chosen tasks to justify our classification of memorisable vs. non-memorisable tasks. Note that some tasks contain both memorisable and non-memorisable examples, which occur in varying ratios as shown in Table 2. Additionally, for our categorisation, we assume that leakage of task data is not a factor, i.e., an example is memorisable if and only if it can be solved through memory recall of information. We assume that previous memorisation of the actual question-answer pair has not occurred. (Continued)

Task	Example Memorisable	Example Non-Memorisable
Misconceptions	Twinkies are edible for decades or longer. ***Reason: Model must recall factual information about common topics.***	n/a
Modified arithmetic	n/a	In the following lines, the symbol -> represents a simple mathematical operation. 102 + 435 -> 537 ... 466 + 214 -> ***Reason: A novel question; math reasoning necessary.***
Phrase relatedness	home town "town center", "location", "native city"... ***Reason: Model must determine word co-occurrence likelihood based on previously-encountered text.***	n/a
Physical intuition	An object is moving in a vacuum at velocity V with no net external forces acting on it. Does the object have nonzero acceleration? ***Reason: Model must recall factual information about physics.***	n/a
Social IQA	n/a	Riley layered down their arms with a blanket. What does Riley need to do before this? ***Reason: Model must reason about novel social situations.***
Strange stories	n/a	Jane and Sarah are best friends. They both entered the same painting competition. Now Jane wanted to win this competition very much indeed, but when the results were announced it was her best friend Sarah who won, not her. Jane was very sad she had not won, but she was happy for her friend, who got the prize. Jane said to Sarah, "Well done, I'm so happy you won!" Jane said to her mother, "I'm sad I didn't win that competition!" Why does Jane say she is happy and sad at the same time? ***Reason: Model must reason about novel social situations.***
Strategy QA	Was Pollock trained by Leonardo da Vinci? ***Reason: Model can solve this by recalling previously-encountered text (such as a biography).***	Could an escapee swim nonstop from Alcatraz island to Siberia? ***Reason: Model must combine known concepts to a novel, hypothetical scenario.***

G   Complete results

In this section, we present our complete results. These encompass the performance plots for each of our 22 tasks, arranged in the following order by model type: GPT, T5, and Other Models (Falcon and LLaMA). For each model, the results are ordered as follows:

1. Exact match accuracy in the closed prompt setting

2. Exact match accuracy in the closed adversarial prompt setting

3. Exact match accuracy in the open prompt setting

4. BERTScore accuracy in the closed prompt setting

5. BERTScore accuracy in the open prompt setting

6. Edit distance in the closed prompt setting

7. Edit distance in the open prompt setting

Note that some metrics aren't compatible with all tasks (e.g., BERTScore accuracy with GSM8K, see Section 2.4), and that the *codenames* task is incompatible with the open prompt setting, since the task requires choices to be provided in the input (see Section 2.4 and Table 7). For this reason, some figures will contain fewer than 22 plots.

Model family	Metric	Prompt format	Result
GPT	Exact match accuracy	closed closed adversarial open	Figure 6 Figure 7 Figure 8
	BERTScore accuracy	closed open	Figure 9 Figure 10
	Edit distance	closed open	Figure 11 Figure 12
T5	Exact match accuracy	closed closed adversarial open	Figure 13 Figure 14 Figure 15
	BERTScore accuracy	closed open	Figure 16 Figure 17
	Edit distance	closed open	Figure 18 Figure 19
Falcon	Exact match accuracy	closed closed adversarial open	Figure 20 Figure 21 Figure 22
	BERTScore accuracy	closed open	Figure 23 Figure 24
	Edit distance	closed open	Figure 25 Figure 26
LLaMA	Exact match accuracy	closed closed adversarial open	Figure 27 Figure 28 Figure 29
	BERTScore accuracy	closed open	Figure 30 Figure 31
	Edit distance	closed open	Figure 32 Figure 33

Table 8: Performance plots (Result) for models in each model family (Model family) using different metrics (Metric) in the closed, closed adversarial, and open settings (Prompt format).
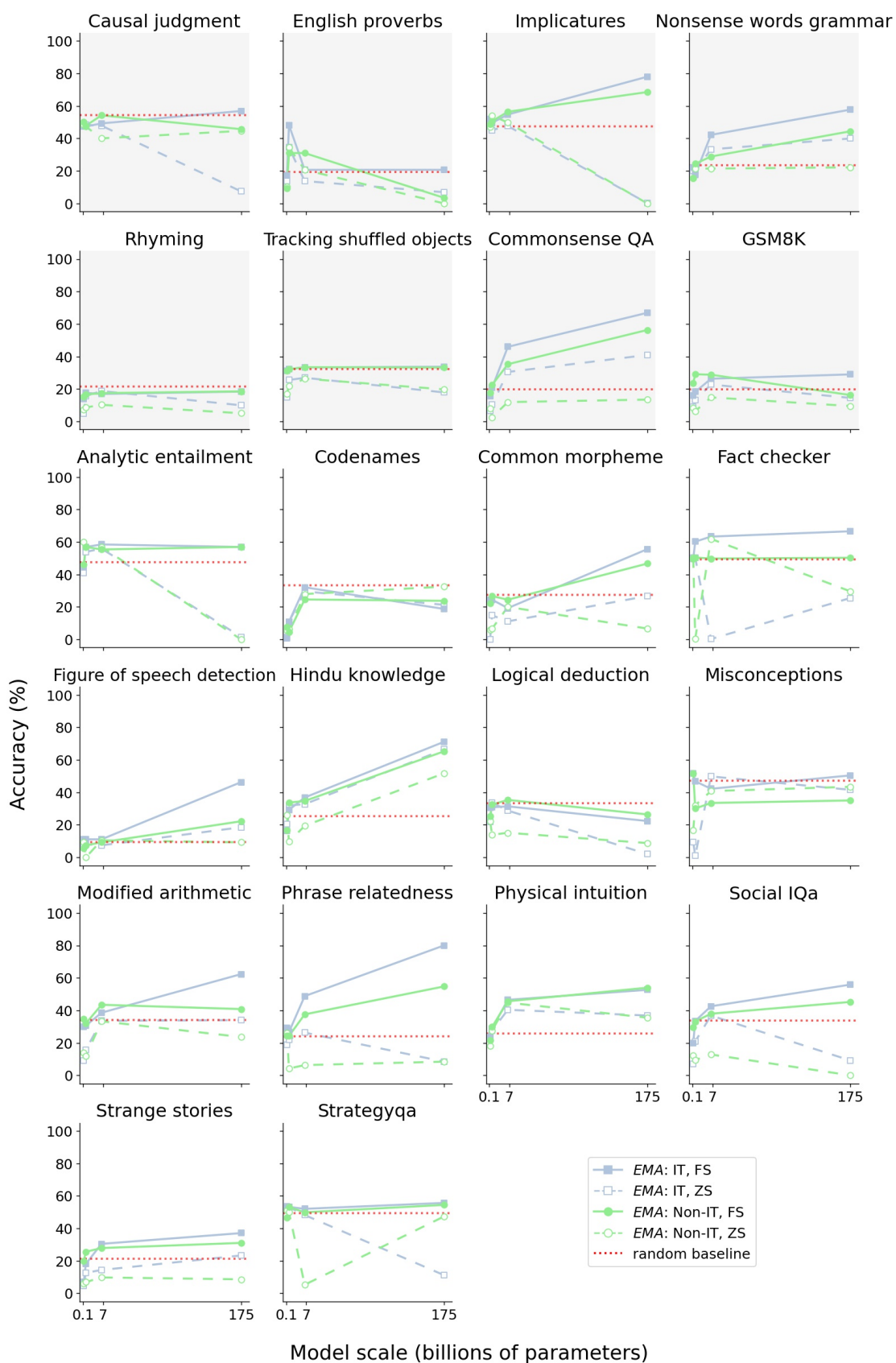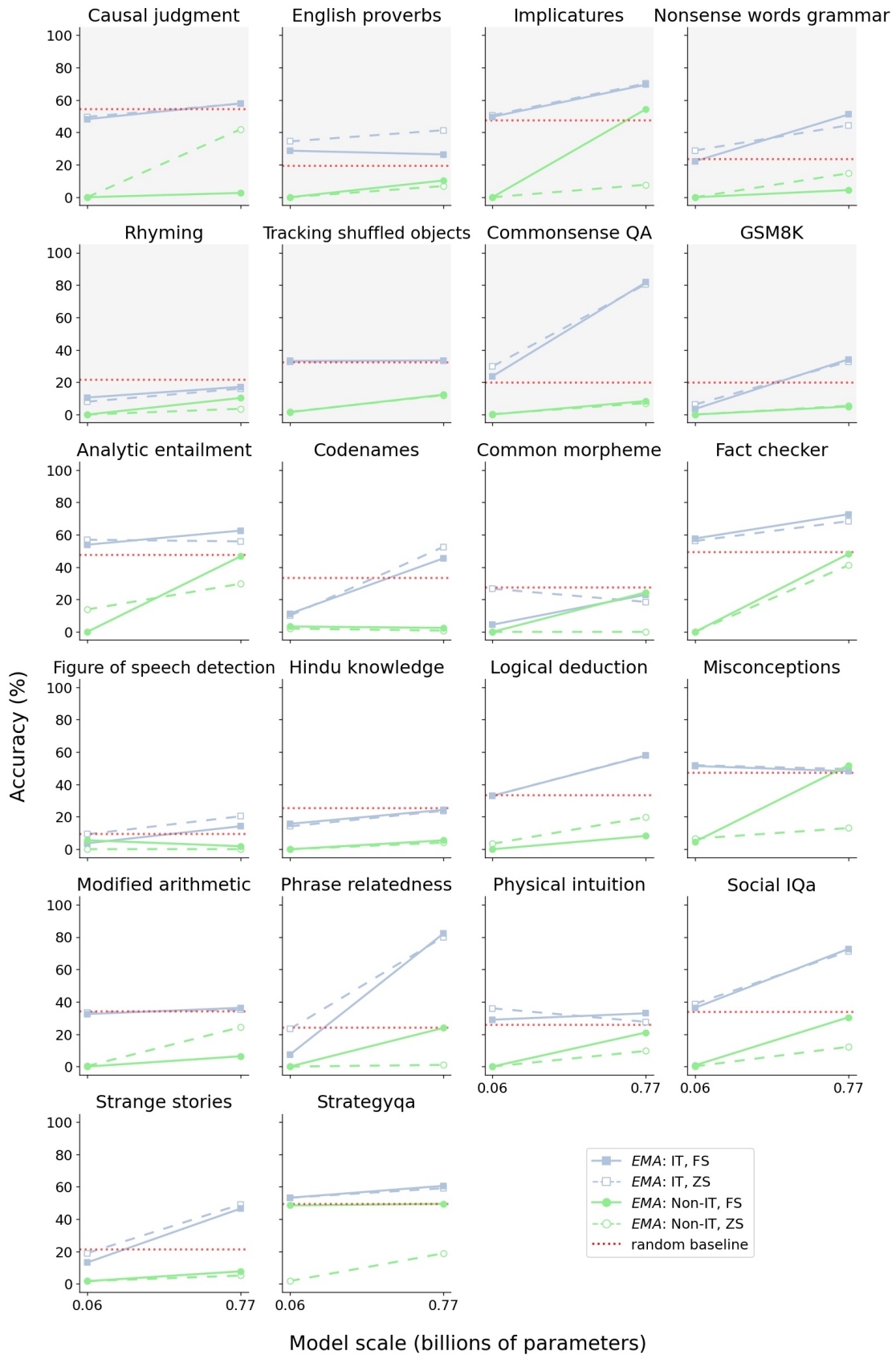
Figure 6: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
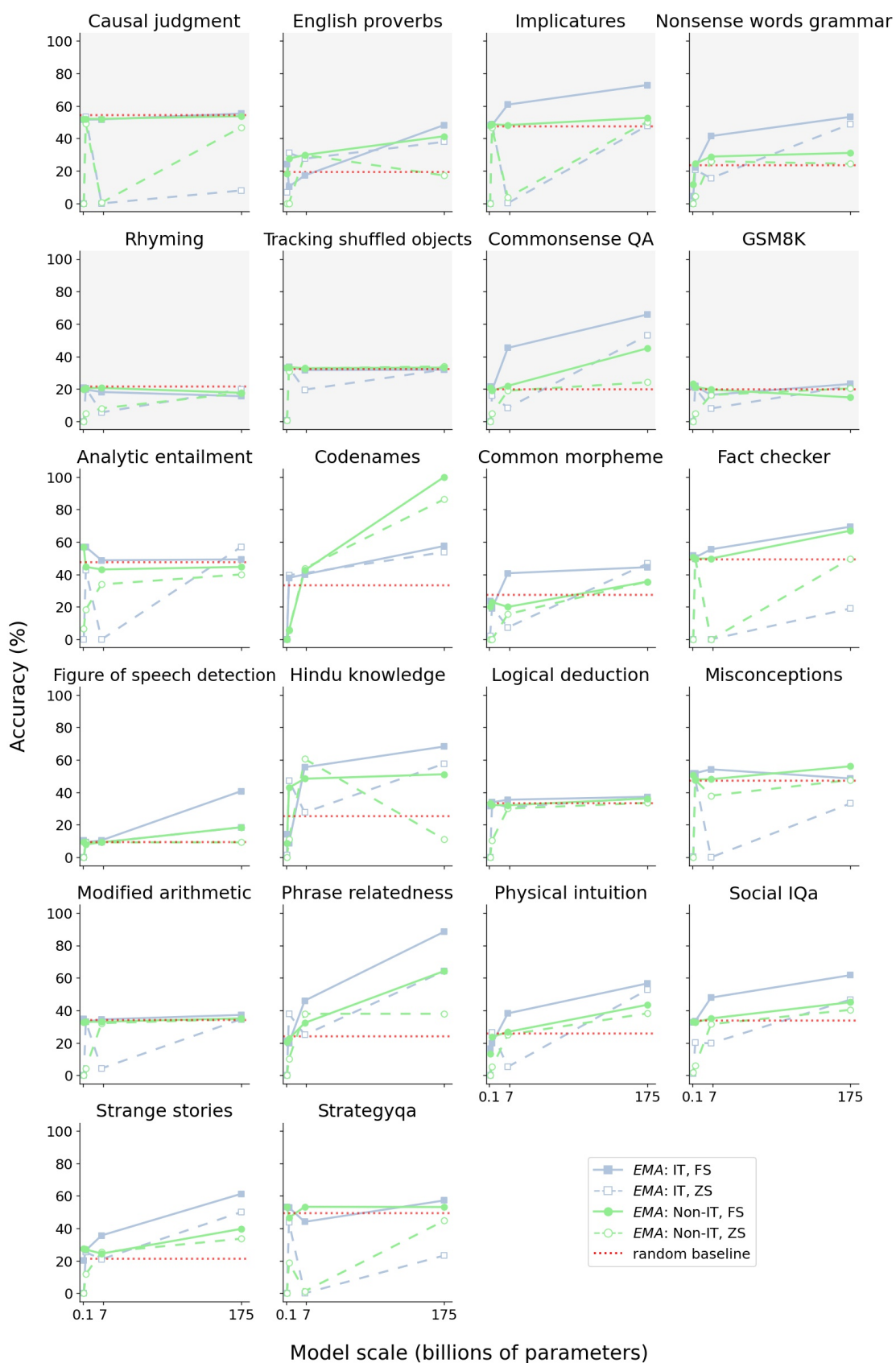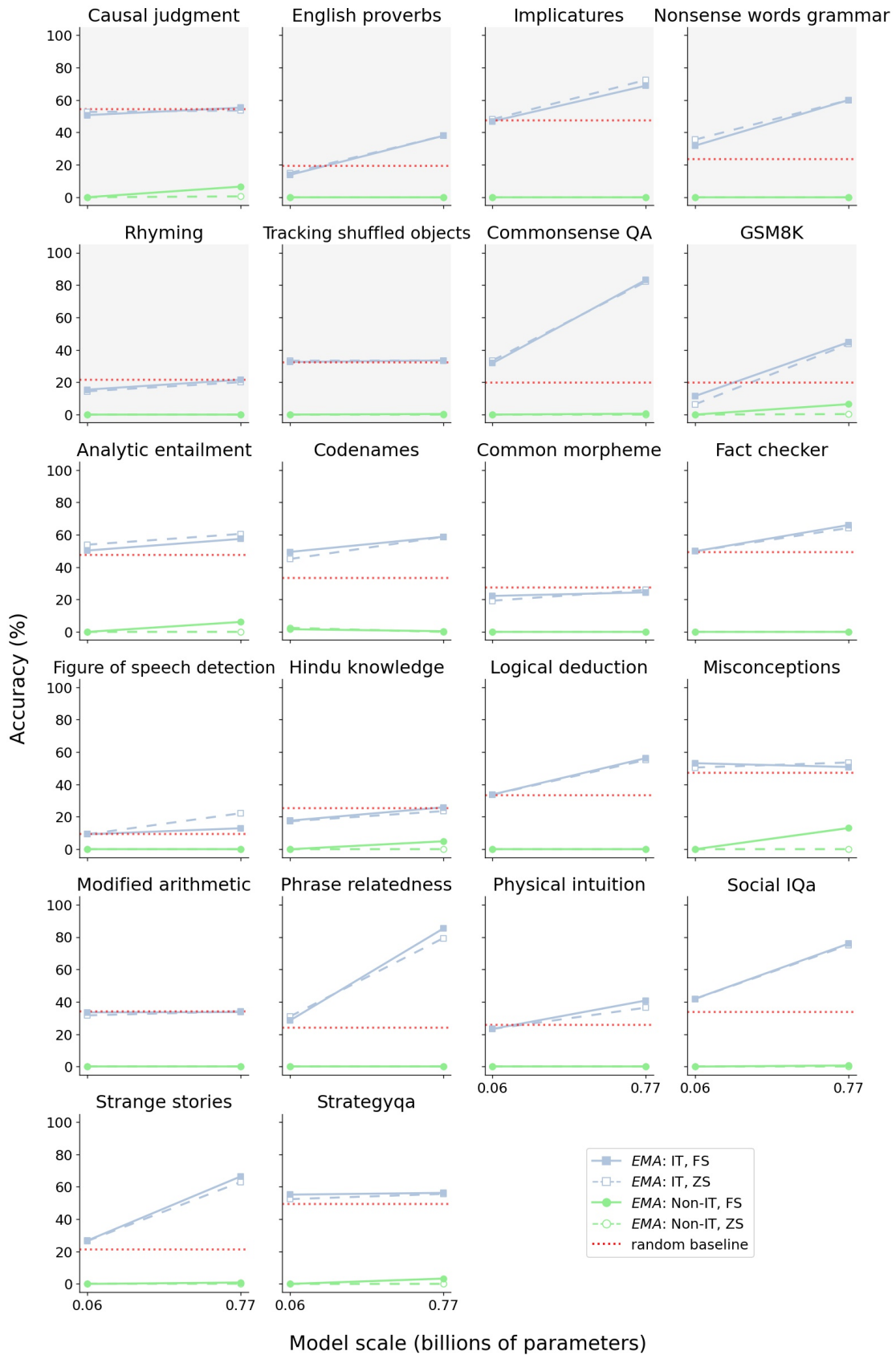
Figure 7: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the closed adversarial prompt in the settings of zero-shot (ZS) and few-shot (FS).
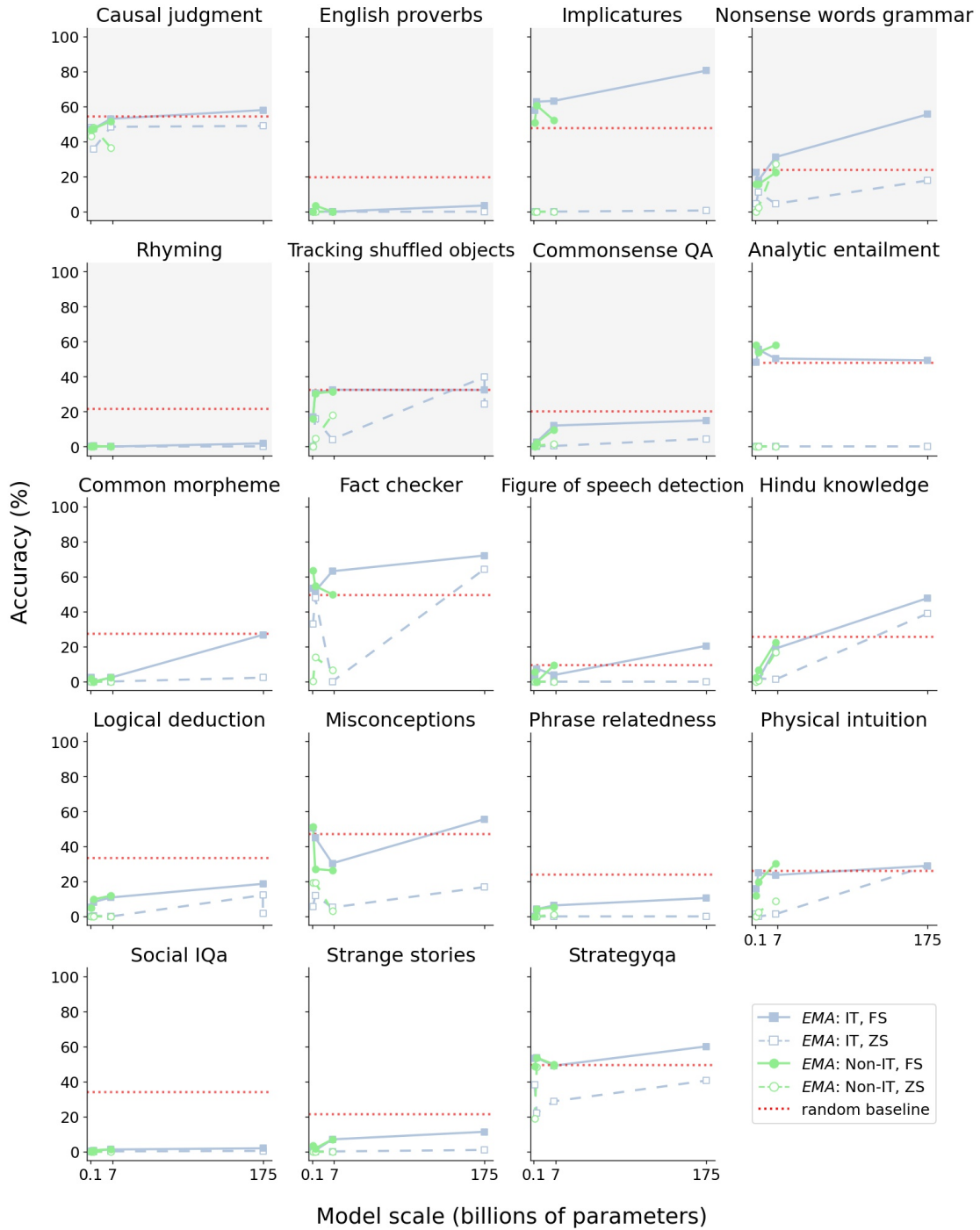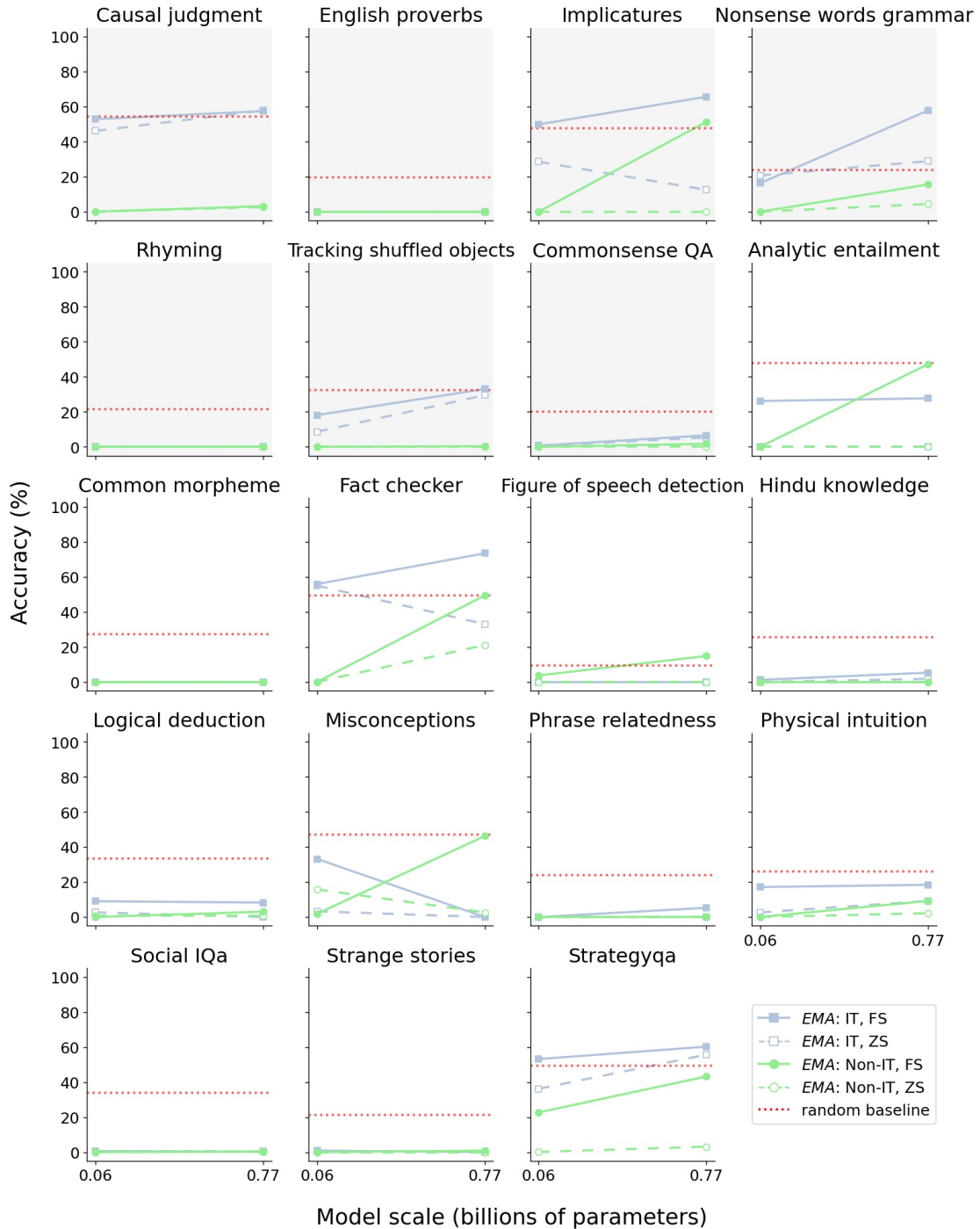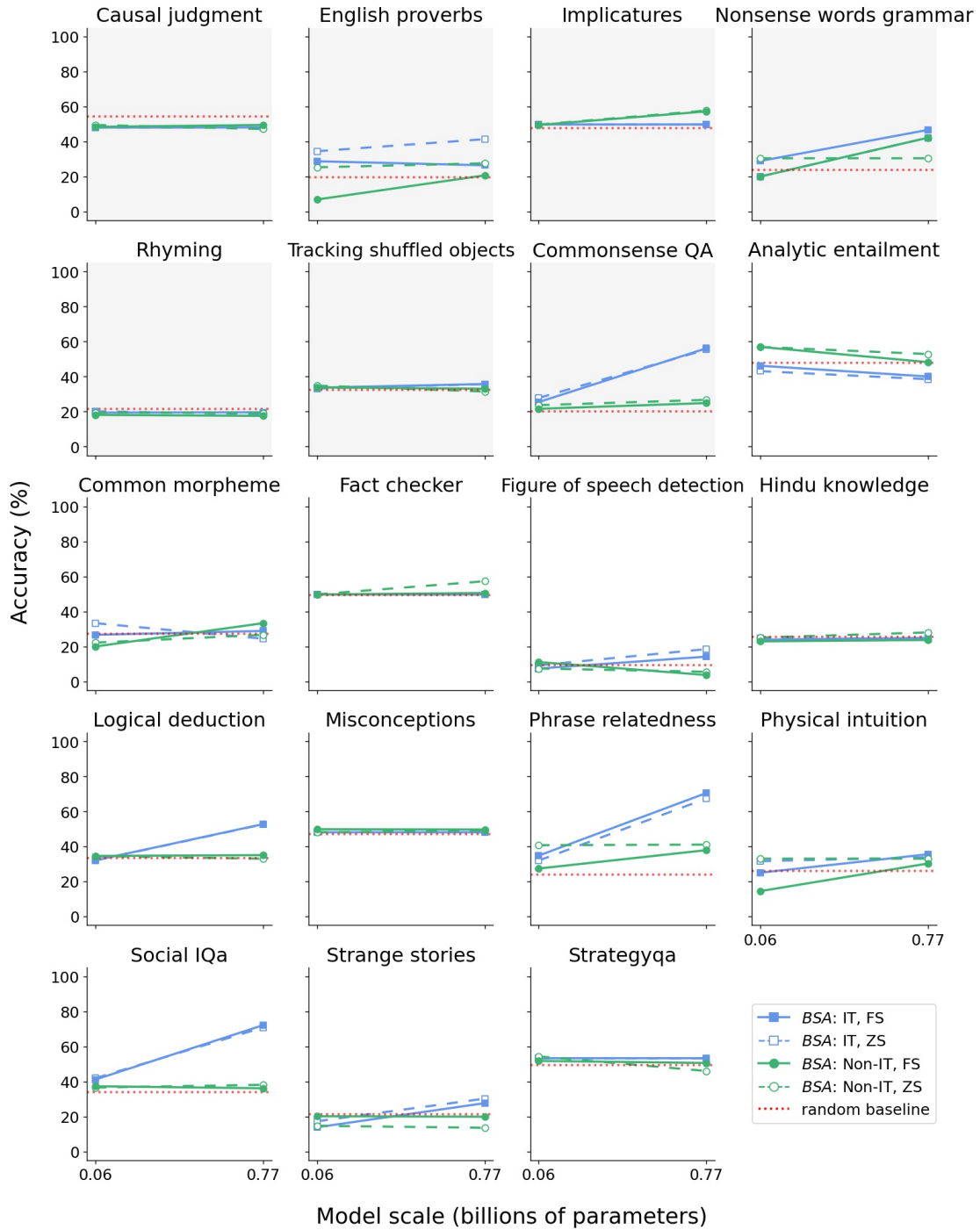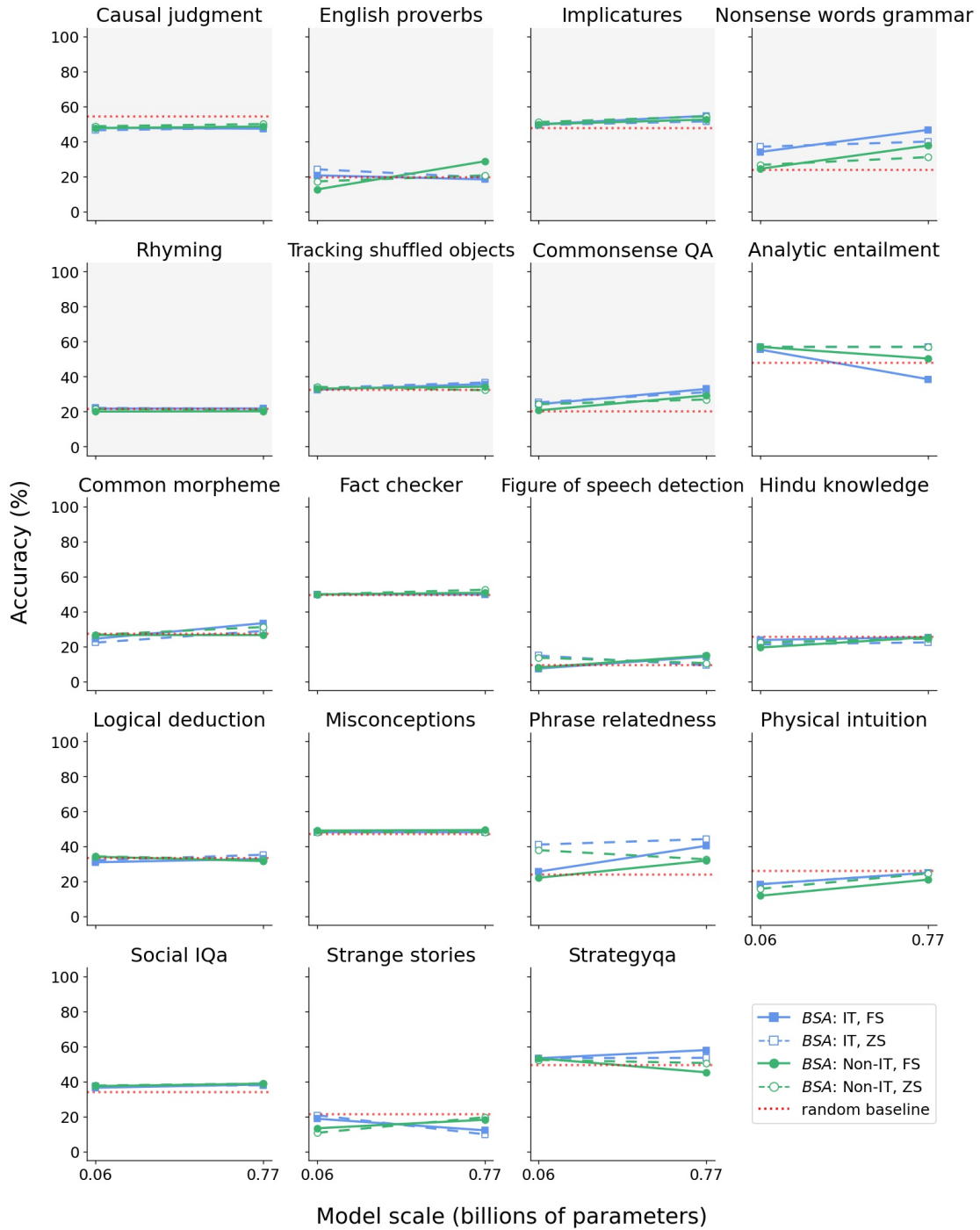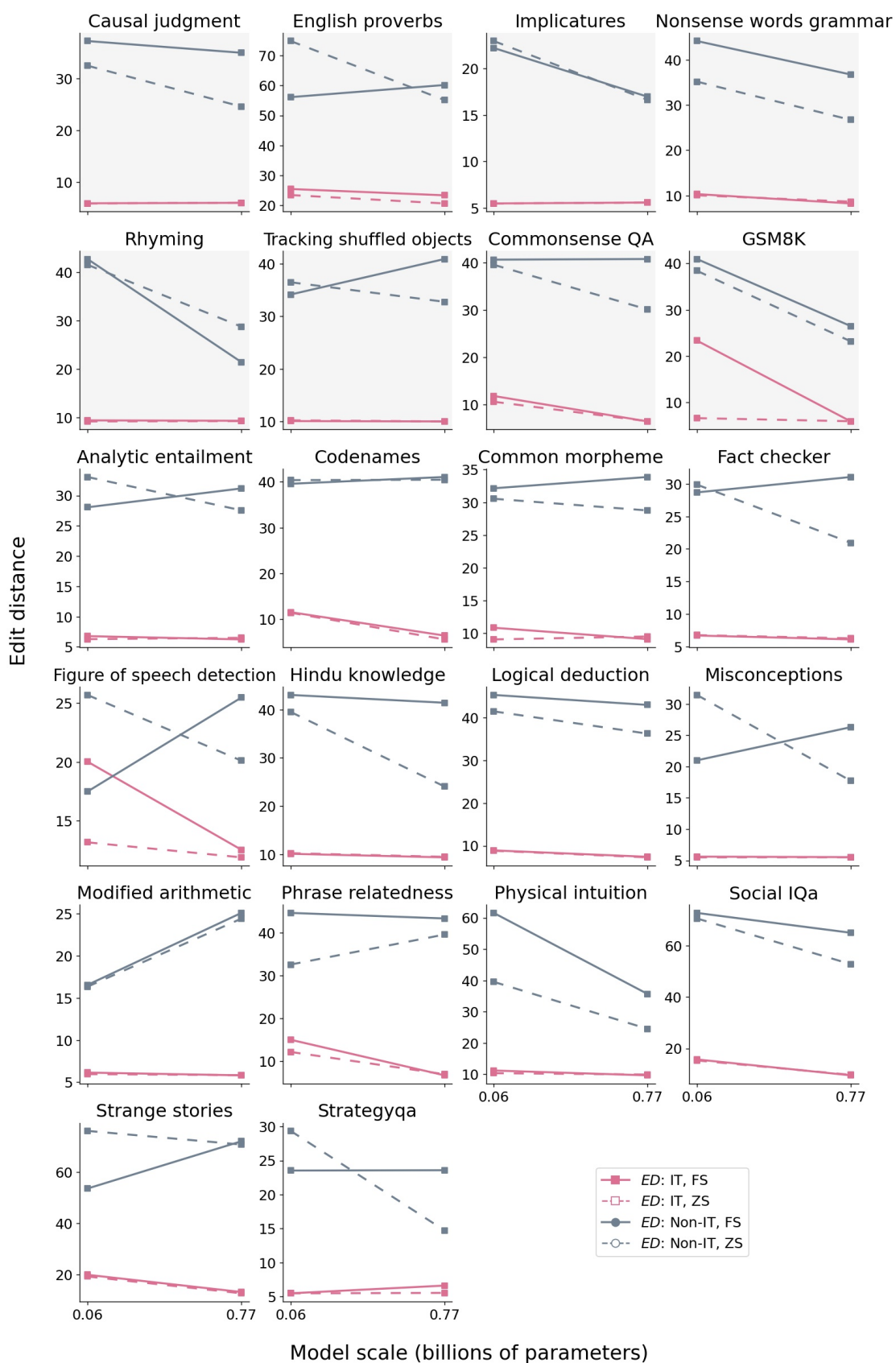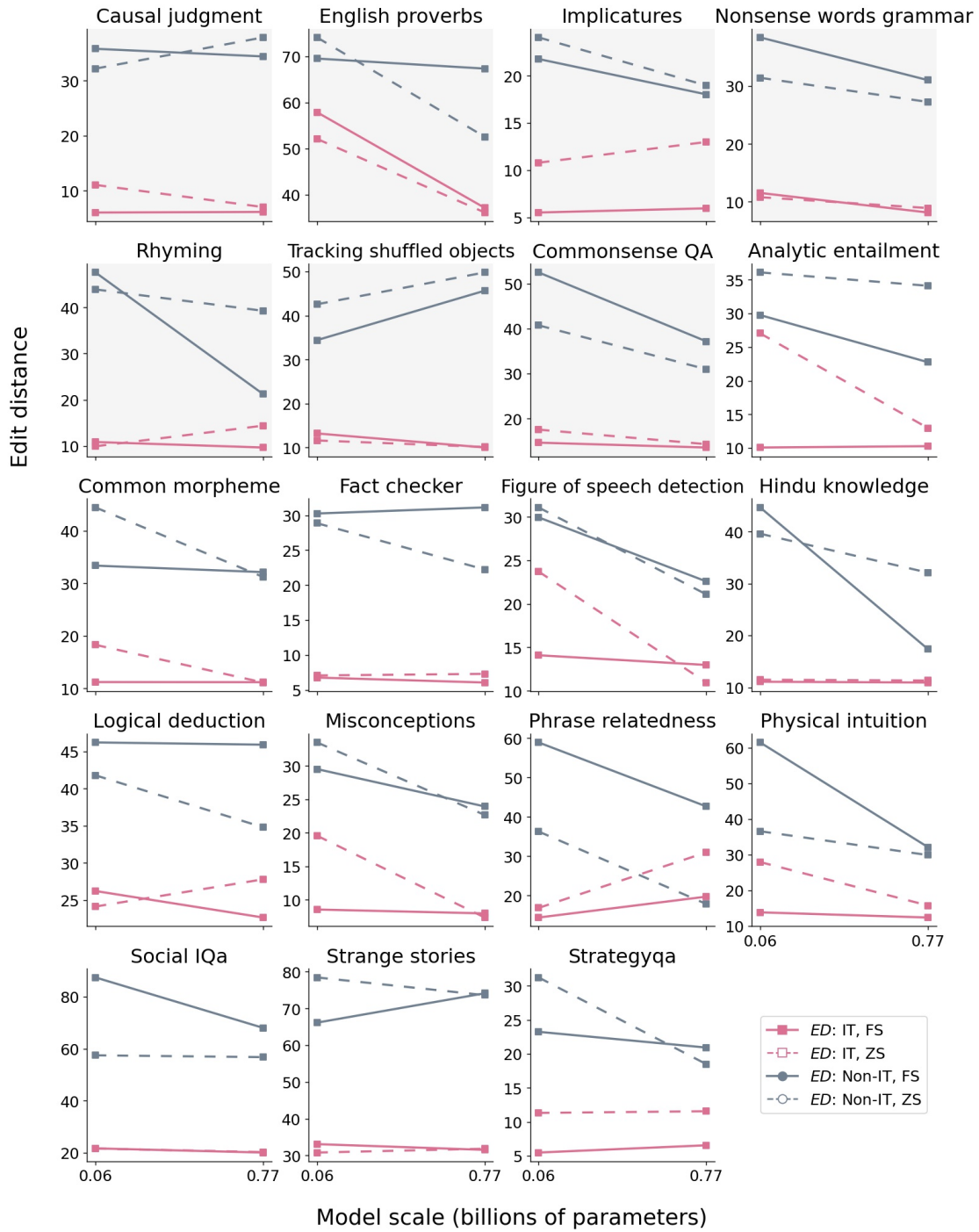
Figure 8: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
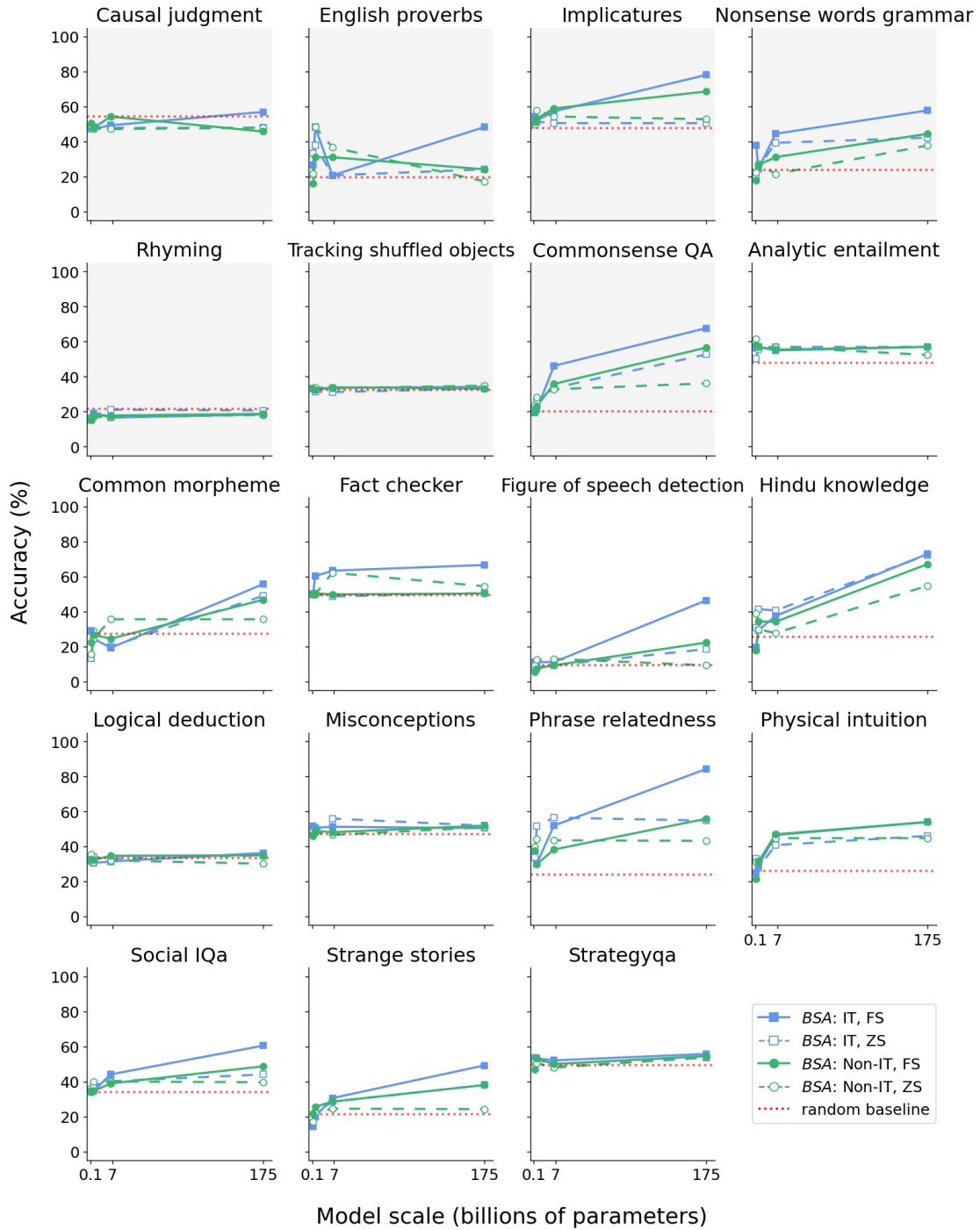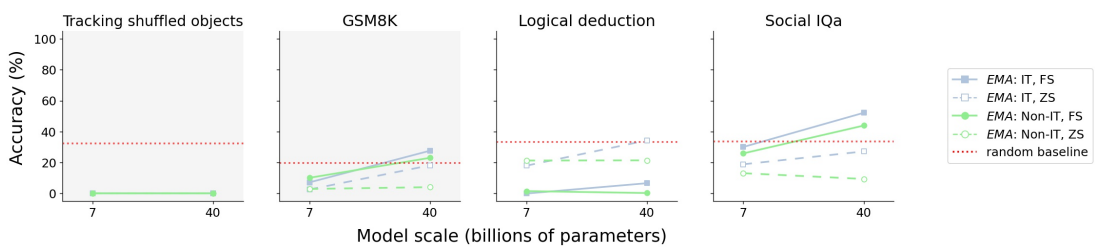
Figure 9: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
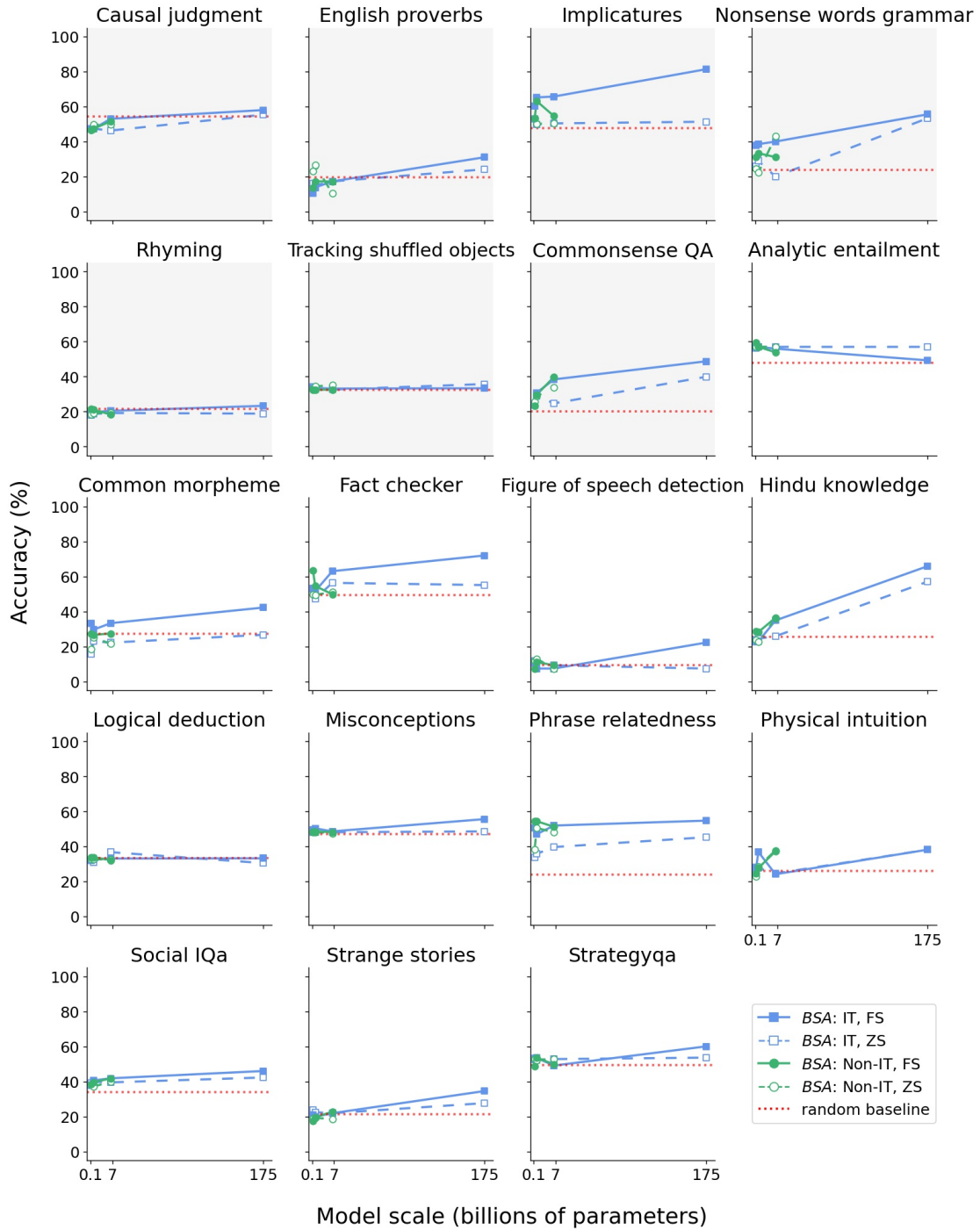
Figure 10: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
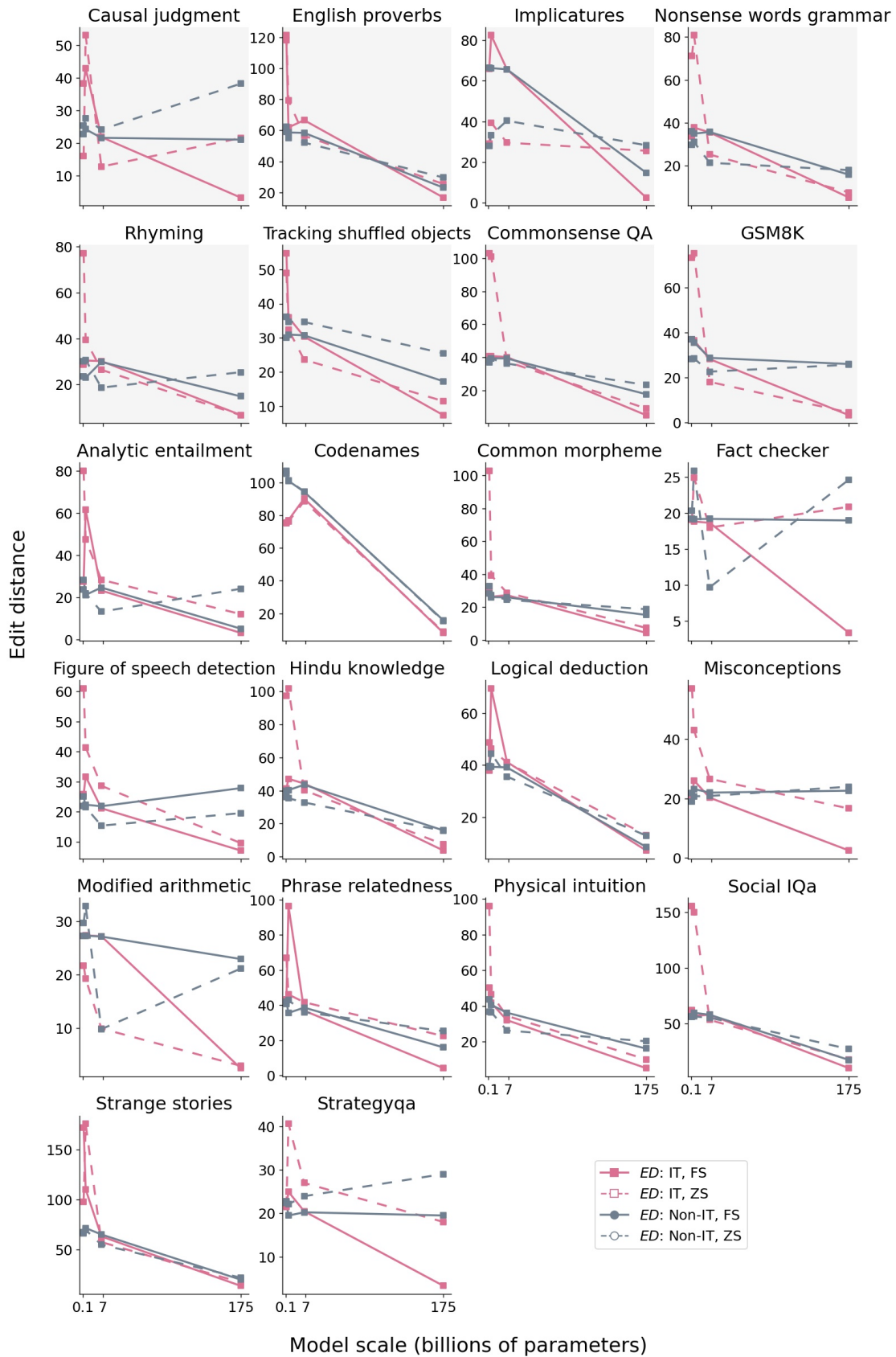
Figure 11: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
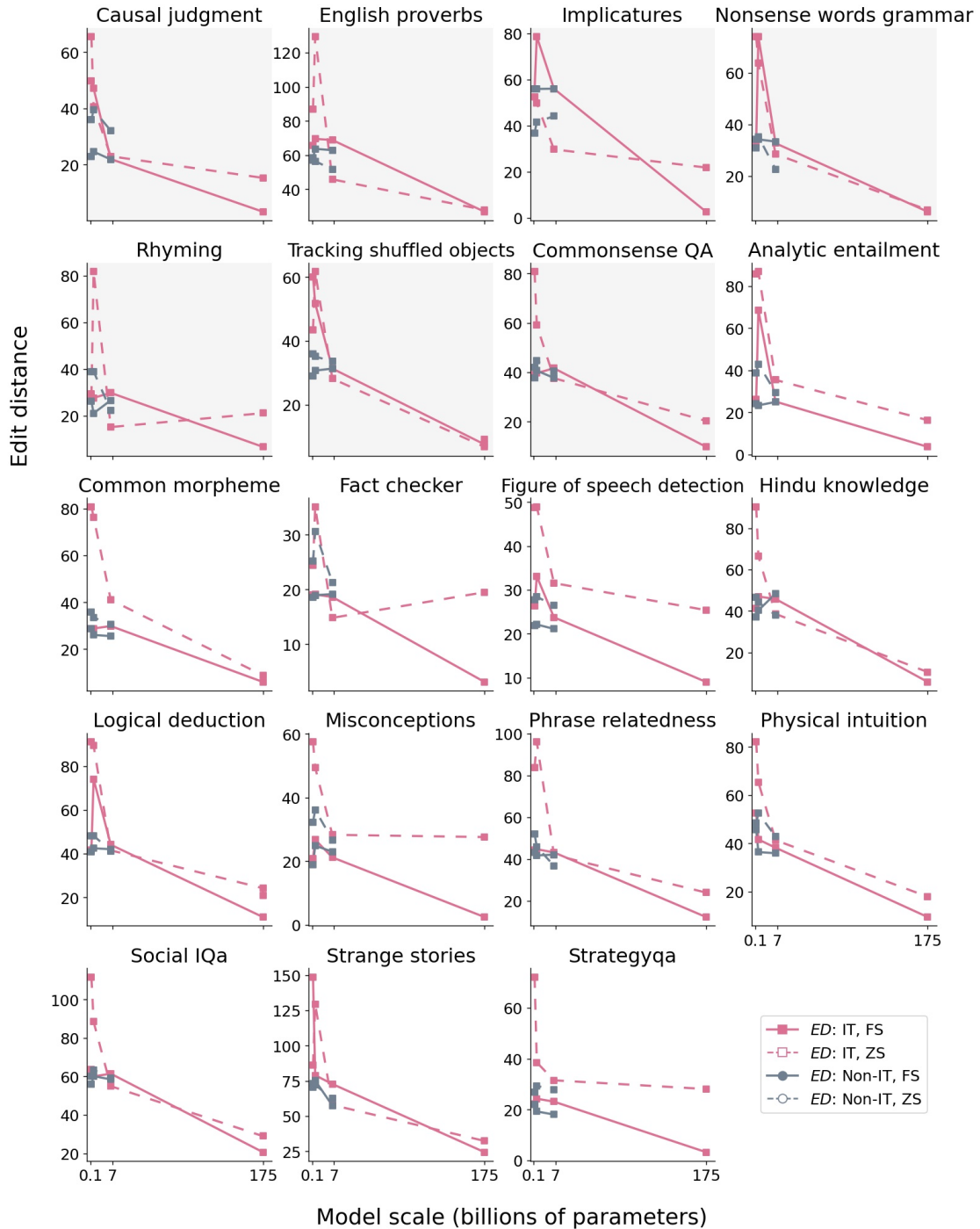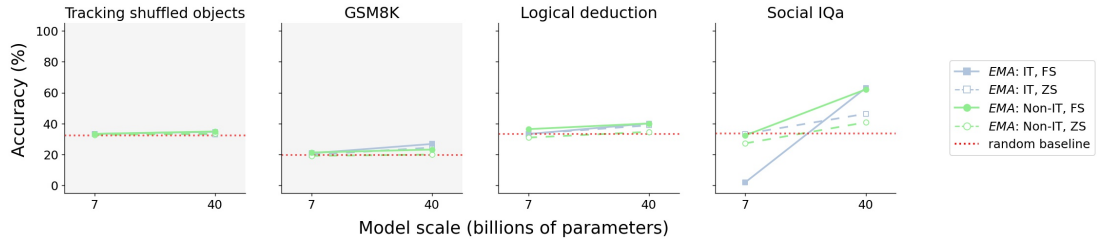
Figure 12: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) GPT models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 13: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 14: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the closed adversarial prompt in the settings of zero-shot (ZS) and few-shot (FS).
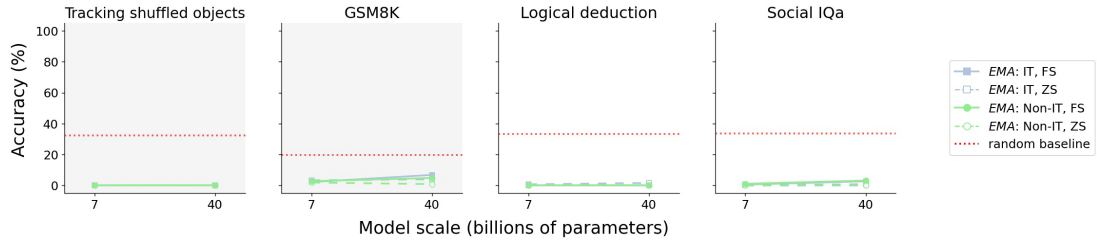
Figure 15: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
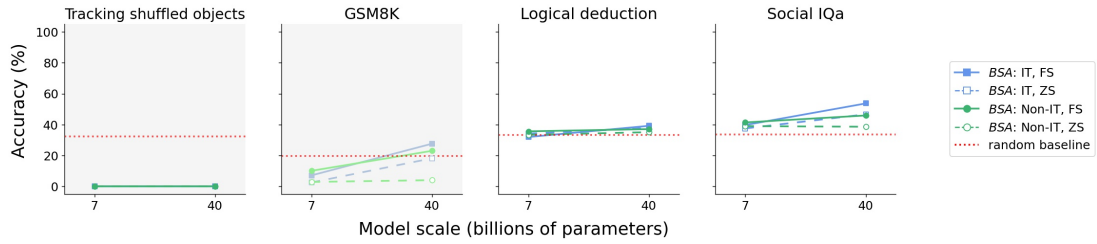
Figure 16: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
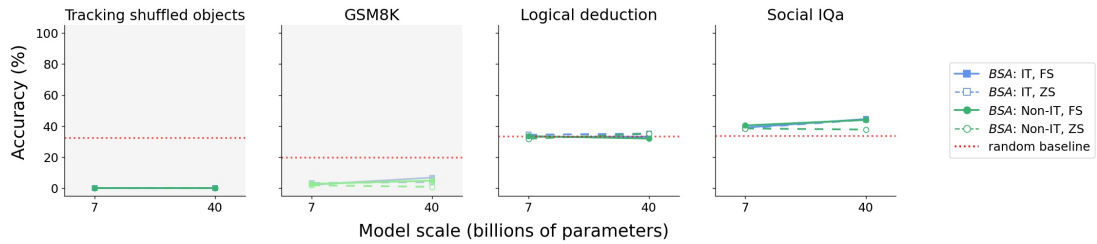
Figure 17: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
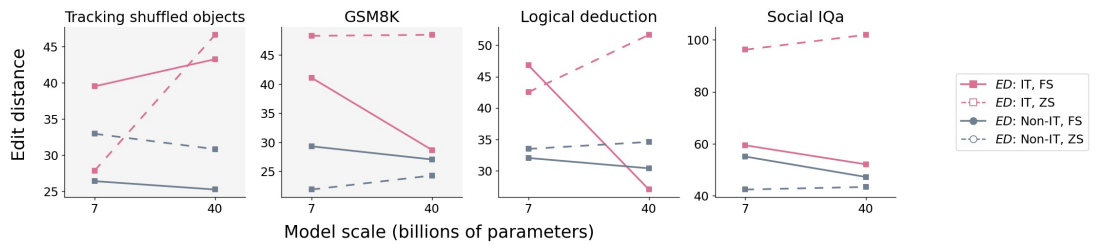
Figure 18: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 19: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) T5 models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 20: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
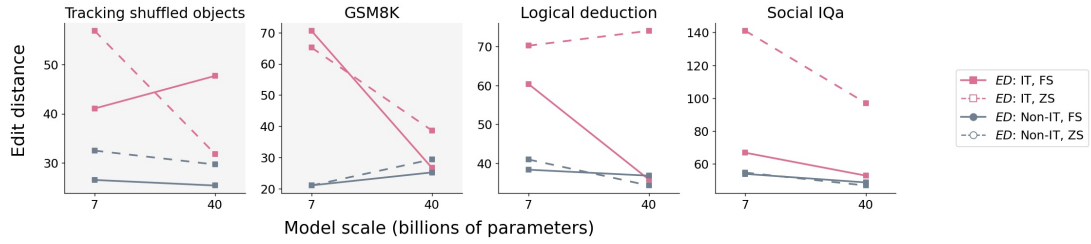
Figure 21: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the closed adversarial prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 22: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
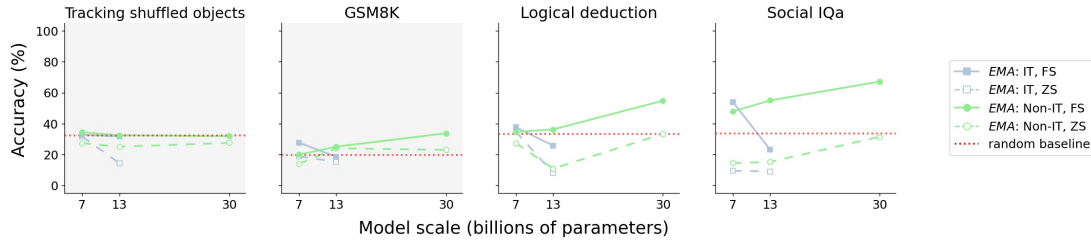

Figure 23: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
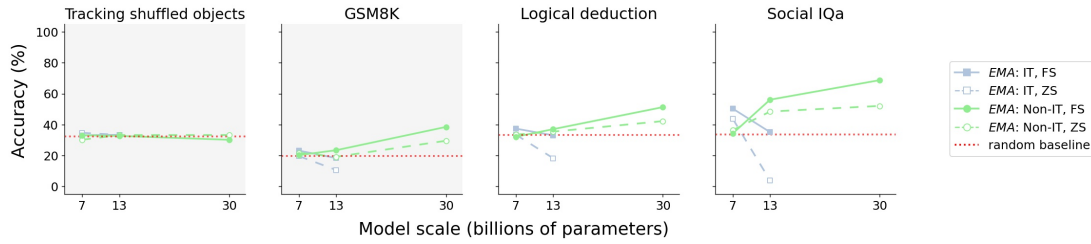

Figure 24: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 25: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 26: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) Falcon models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
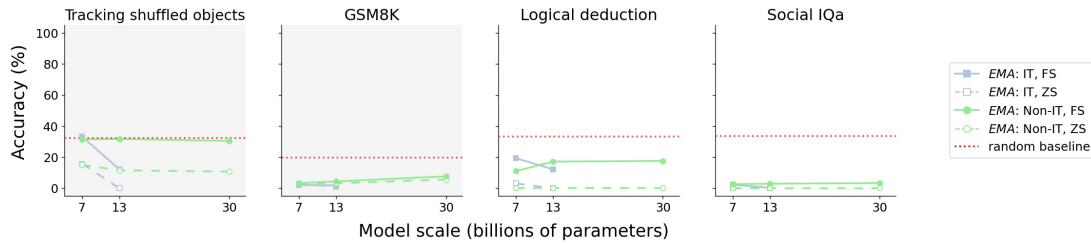

Figure 27: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 28: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the closed adversarial prompt in the settings of zero-shot (ZS) and few-shot (FS).

Figure 29: Exact match accuracy (EMA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
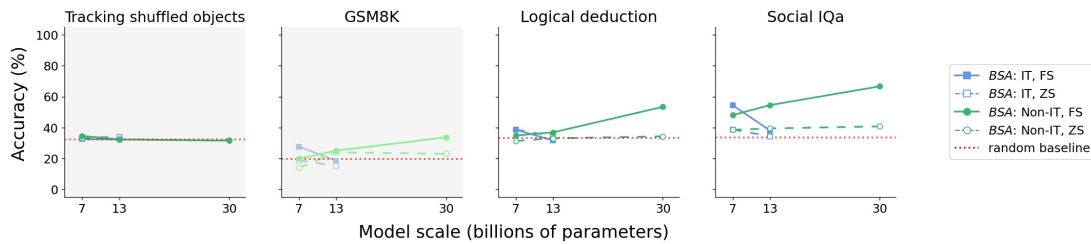

Figure 30: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
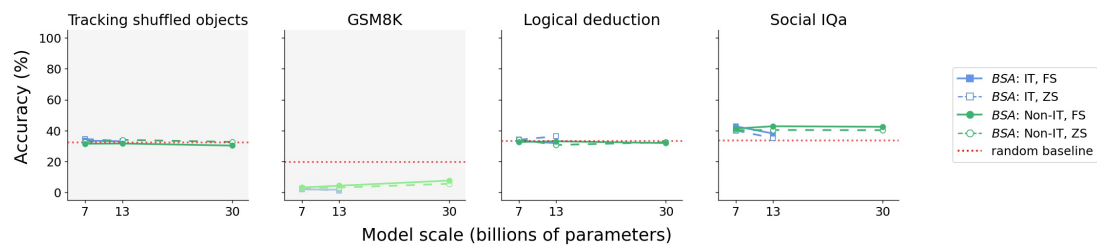
Figure 31: BERTScore accuracy (BSA) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).
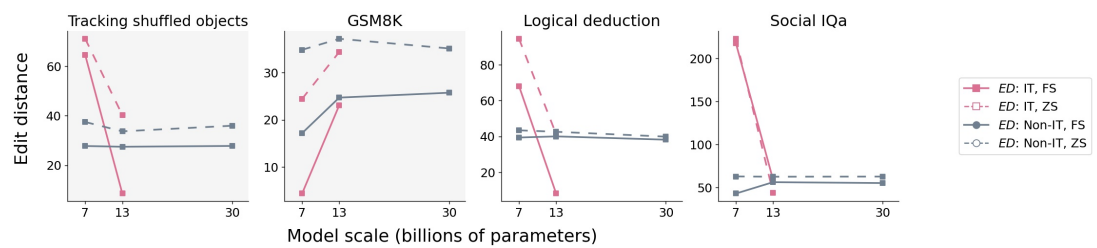

Figure 32: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the closed prompt in the settings of zero-shot (ZS) and few-shot (FS).
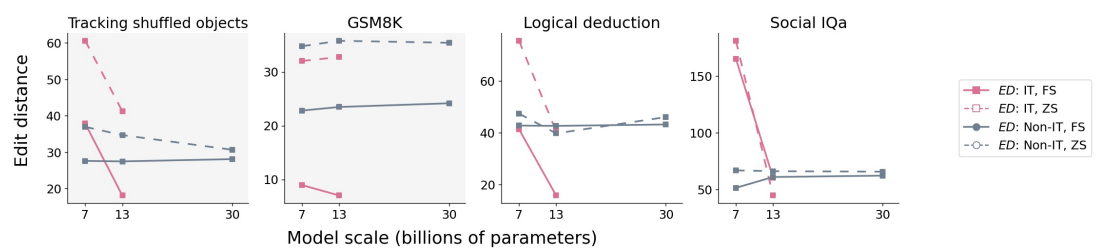

Figure 33: Edit distance (ED) for instruction-tuned (IT) and non-instruction-tuned (Non-IT) LLaMA models using the open prompt in the settings of zero-shot (ZS) and few-shot (FS).