Experimental Contexts *Can* Facilitate Robust Semantic Property Inference in Language Models, but Inconsistently

Kanishka Misra[τ] **Allyson Ettinger**[α] **Kyle Mahowald**[τ]
[τ]The University of Texas at Austin [α]Allen Institute for Artificial Intelligence
kmisra@utexas.edu allysone@allenai.org kyle@utexas.edu

Abstract

Recent zero-shot evaluations have highlighted important limitations in the abilities of language models (LMs) to perform meaning extraction. However, it is now well known that LMs can demonstrate radical improvements in the presence of experimental contexts such as in-context examples and instructions. How well does this translate to previously studied meaning-sensitive tasks? We present a case-study on the extent to which experimental contexts can improve LMs' robustness in performing property inheritance—predicting semantic properties of novel concepts, a task that they have been previously shown to fail on. Upon carefully controlling the nature of the in-context examples and the instructions, our work reveals that they can indeed lead to non-trivial property inheritance behavior in LMs. However, this ability is inconsistent: with a minimal reformulation of the task, some LMs were found to pick up on shallow, non-semantic heuristics from their inputs, suggesting that the computational principles of semantic property inference are yet to be mastered by LMs.

1 Introduction

Carefully controlled behavioral analyses on meaning-sensitive tasks have revealed holes in the ability of language models (LMs) to demonstrate robust meaning extraction and use (Pandia and Ettinger, 2021; Elazar et al., 2021; Schuster and Linzen, 2022; Misra et al., 2023; Kim and Schuster, 2023, *i.a*). However, since a large subset of these investigations uses zero-shot evaluation as the primary methodology, there are growing concerns that they do not paint a complete picture of LMs' abilities (Lampinen, 2022; Sinclair et al., 2022; Sinha et al., 2023). Conclusions that LMs lack a particular ability may be overhasty if it turns out the ability is easily accessed through in-context learning, different question formulations, or particular instructions (Lampinen, 2022; Wei et al., 2022).

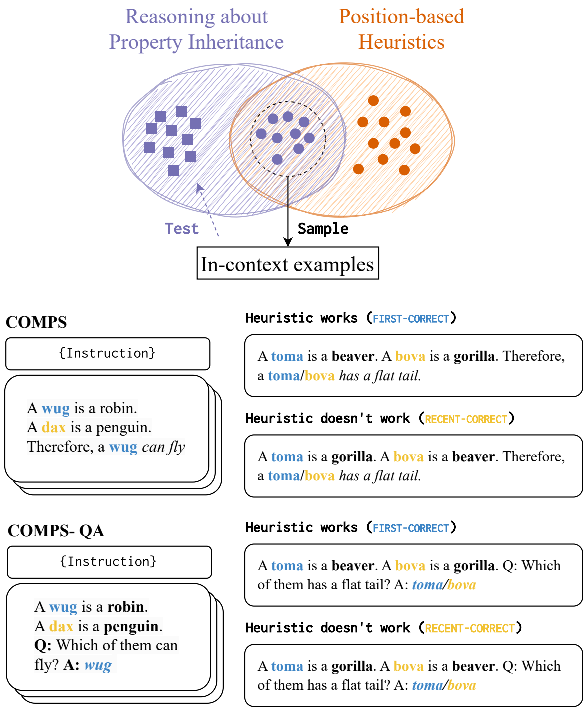


Figure 1: We prompt LMs with in-context examples that are compatible with both, robust property inheritance, as well as position-based heuristics. At test time, we evaluate the model on cases where the heuristics support desirable behavior *and* on cases where they do not. We use stimuli from COMPS and its reformulation as a QA task (COMPS-QA).

Our focus[1] in this paper is a particularly challenging data set for meaning-sensitive behavior: COMPS (Misra et al., 2023), a dataset of minimal pair sentences that tests the ability of LMs on property knowledge of everyday concepts (*a* ***beaver****/gorilla has a flat tail*) and their inheritance for novel concepts (*a wug is a* ***beaver****/gorilla. therefore a wug has a flat tail*). Contemporary LMs failed miserably on the hardest subset of the COMPS stimuli, the examples of which contain two novel concepts (WUG VS. DAX), where only one of them inherits the target property (*has a flat tail*):

[1]Our code can be found in this link

(1) A **wug** is a beaver. A **dax** is a gorilla. Therefore, a **wug**/**dax** has a flat tail.

Given the success of LMs on a wide variety of complicated tasks, their utter failure on this seemingly straightforward task remains puzzling. Here, we systematically explore COMPS on five modern LMs ranging from 1.5–13B parameters, varying (a) whether models are evaluated zero-shot or with multiple examples and (b) whether or not instructions are present.

Unlike other minimal-pair datasets, using COMPS in an in-context learning setting is non-trivial (and thus potentially informative). This is because the task can be solved using a position-based heuristic. For example, in one subset of COMPS, the target property is always attached to the **first** novel concept—like in (1). Importantly, models' failures on COMPS were shown to be in part a result of models' tendencies towards heuristic behavior: the performance of LMs is particularly bad when the distractor (*a dax is a gorilla*) is *recent*—i.e., autoregressive LMs show a recency bias in attributing properties to novel concepts. In that sense, COMPS follows a rich body of work in which tasks are set up in a manner that two types of generalization mechanisms can lead to the same prediction, but only one of which is desirable (McCoy et al., 2019, 2020; Warstadt et al., 2020b; Mueller et al., 2022; Si et al., 2023).

We find that experimental contexts, as operationalized using in-context examples and instructions, can in fact demonstrate robust improvements in LMs' property inheritance behavior as measured by the stimuli in COMPS. However, this improvement comes with a caveat: With a minimal reformulation of COMPS into a QA task, where there is a direct link between the LMs' output space and the features of the input that control the heuristic, LMs show a strong preference towards the heuristic, and are therefore at chance. This discrepancy suggests that the improvements on the original task do not necessarily indicate that the models have successfully mastered the reasoning ability required to perform property inheritance, which remains a key challenge for them.

2 Methodology

Dataset We use the most difficult subset of the COMPS dataset (Misra et al., 2023)—COMPS-WUGS-DIST—for our experiments. This dataset contains 13,828 sentence pairs of the form similar to (1), constructed using 152 animal concepts and 991 properties.

Stimuli re-design We take a number of steps to minimize noise from other (likely uninterpretable) heuristics beyond the ones we have set out to target. First, we enforce that the concepts and properties that appear in the in-context examples are disjoint from ones that are used in tests. To this end, we sample 50 concepts and their relevant properties and reserve it for our in-context examples, leaving the rest to be sampled for our test set. We also enforce this constraint for our novel concepts—i.e., all in-context examples contain different nonce words, and the collection of nonce words for the in-context examples and the test set is disjoint. Furthermore, we counterbalance the nonce words in the test set in a manner that having a bias towards one of them would lead to chance performance. We additionally also use multiple different sets of in-context examples, to add variability and to ensure that the results are not only due to one particular choice of in-context examples. In total, we use 10 different in-context learning example sets, each containing 6 different COMPS stimuli. For our test set, we use a constant set of 256 unique pairs sampled from our pool of stimuli containing unused concepts and properties.

Heuristics Our most important design decision is to consider two distinct sets of stimuli—each separately making available the two types of heuristics that the LMs could rely upon: FIRST-CORRECT and RECENT-CORRECT, where the property is inherited by the first and the most recent novel concept, respectively. That is, for the same set of in-context examples, we have a version where the first concept is correct like in (1), and one where the most recent concept is correct:

(2) A **wug** is a gorilla. A **dax** is a beaver. Therefore, a **wug**/**dax** has a flat tail.

For each type of in-context stimuli, we similarly have two versions of test stimuli: one that is consistent with the target heuristic, and one that is not. That is, a test example that is consistent with the FIRST-CORRECT heuristic will also have its first concept be the one that inherits the property in question, while one which is inconsistent will have the most recent concept be the inheritor of the property. Therefore, a model that shows a preference for a given heuristic will succeed only on one test set and succumb on the other, while a model that is robust to the heuristics will succeed on both.

Reformulation into QA The original COMPS stimuli test for property inheritance using declarative statements, where models are tested for the log-probability they asign to the property (*has a flat tail*) given either of the two concepts (*wug* vs. *dax*). Here we additionally consider an alternate formulation of COMPS as a question answering task (COMPS-QA), where we make the property explicit in the prompt to the model and instead ask which of the two concepts possesses it:

(3) A **wug** is a beaver. A **dax** is a gorilla. **Question:** Which one of them has a flat tail? **Answer:** **wug/dax**

Since the shallow heuristics we consider are controlled by the relative ordering of the novel concepts, this formulation of the task directly allows us to link the models' output space (the novel concepts) to the heuristics (positions).

Testing setup For the original COMPS setting we follow Misra et al. (2023) and compare the log-probability of the property phrase given the correct vs. the incorrect prefix. For COMPS-QA however, since we have a constant prefix (same premises and question), we evaluate the relative log-probability of the two novel concepts, only one of which is the correct answer. Accuracy in both cases is the proportion of cases the correct surface form was assigned relatively higher log-probability. Since we use pairwise comparisons throughout, chance performance is 50%.

Instructions We consider four different kinds of instruction templates, with varying levels of detail (see appendix B) per formulation (COMPS and COMPS-QA). In our experiments we report results on the instruction that gives the best average performance for a given model.

LMs tested We evaluated 5 different open-source LMs, all of which are decoder-only, and were accessed using the huggingface hub (Wolf et al., 2020): **GPT-2 XL** (Radford et al., 2019); **OPT-6.7b** (Zhang et al., 2022); **Llama-2** (we used the **7b** and the **13b** versions; Touvron et al., 2023); and **Mistral-7b** (Jiang et al., 2023). Details about the models can be found in the appendix.

3 Analyses and Results

We evaluate on COMPS and COMPS-QA, with and without instructions. In each case, we progressively supply 0 through 6 in-context examples, allowing us to track the dynamics of the models' performance with an increasing amount of demonstrations. Together with our separate types of test sets and heuristics encoded in the in-context examples, along with five different instruction settings (four with and one without) we run 2420 experiments per LM. We hypothesize that LMs would be more sensitive to the positional heuristics in COMPS-QA because of the clear link between their output space and the relative position of the novel concepts—the feature that controls our target heuristics.

Figure 2 shows accuracies of the tested LMs on our four different COMPS settings as a function of the number of in-context examples provided to them, for both: cases where the **heuristics are consistent with success on the test set**, and cases where **they are not**. We also show an additional curve denoting the average performance across the two types of test sets to paint an overall picture of the models' performance. In this figure, the extent to which a model relies on heuristic is indicated by the gap between the dotted (●) and the dashed (▲) lines. A model that is robust to the heuristics will have curves of both colors rise above chance, with no gap between the two, while one that is prone to using heuristics will have its dotted (●) curve be substantially greater than its dashed (▲) curve.

Experimental context can improve attribution of properties to concepts... On COMPS, models unsurprisingly start off at chance performance on average, corroborating the previous findings of Misra et al. (2023). However, in the presence of in-context examples and instructions, they are able to improve monotonically as the number of in-context examples increases. It is worth noting **Llama-2-13b** does occasionally show a slight preference for heuristics in the absence of instructions (e.g., 84% vs. 62% when prompted with 2 examples). An intermediate conclusion that we draw here is that LMs can indeed demonstrate non-trivial property inheritance on observing a few examples that reflect that behavior.

...but not the attribution of concepts to properties While experimental context seems to aid models in attributing properties to the right concept in context, the same does not hold on COMPS-QA.

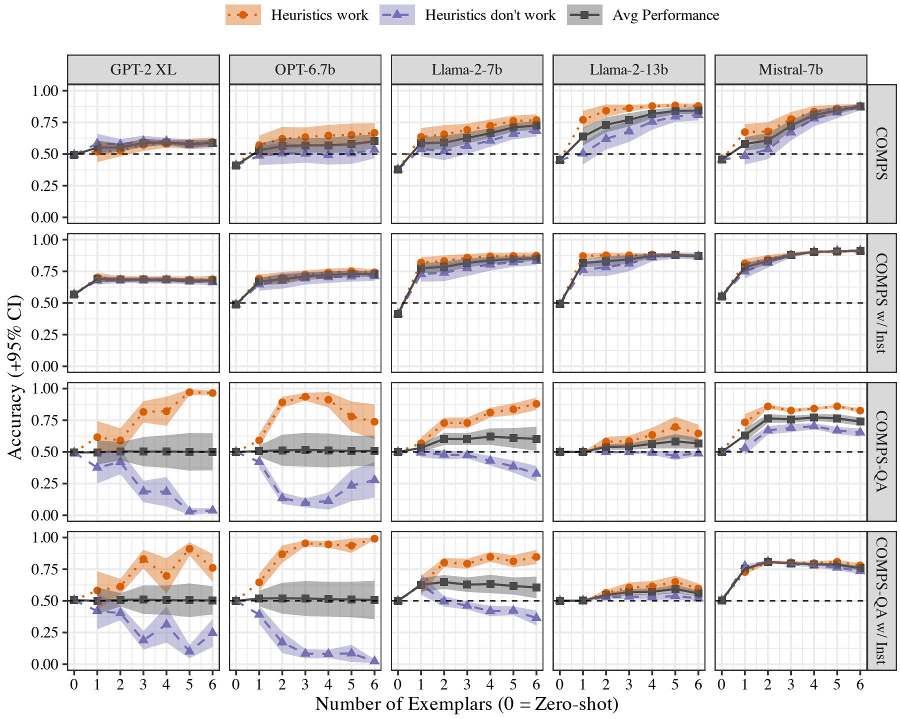


Figure 2: Overall results from our experiments testing LMs on COMPS and COMPS-QA using in-context examples, with and without instructions. Results are aggregated across both heuristics: FIRST-CORRECT and RECENT-CORRECT. Error bars are over different sets of in-context examples. All models start off near chance in the 0-shot case, but many improve as more examples are given. Some (e.g., GPT-2 XL on COMPS-QA) show strongly heuristically driven behavior, as evidenced by the diverging performance on items where heuristics work and those where they do not. Figures 3 and 4 show fine-grained results.

Similar to COMPS, models start off at chance performance on average with a zero-shot set up, however, unlike in the case of COMPS, LMs seem to consistently prefer the heuristics available in the prompt, showing worse than chance performance on cases where the test set does not follow the heuristic. This is most apparent for **GPT-2 XL**, **OPT6.7b**, and **Llama-2-7b**—here the gap between the accuracy for cases where heuristics support performance on the test set and the accuracy for cases where they do not almost always worsens with an increase in the number of exemplars, on average. This is especially notable for **OPT6.7b**, which attains perfect performance on cases where the heuristics match up with the test set and at the same time it is close to 0% on cases where they do not. A notable exception to this trend is **Mistral-7b**, which seems to be resilient to the spurious heuristics, showing a net-positive improvement from the zero-shot case, especially in the presence of instructions. Nevertheless in the absence of instructions, it too shows a slight preference for position-based heuristics—for instance, its accuracy with 6 in-context exemplars when the heuristics support success on the test set is 82% and on cases where the heuristics oppose the test set is 65%.

Our results suggest that LMs are more likely to show behavior that is compatible with the *use* of positional heuristics when their output space (choice between the two novel concepts) has a clear connection with positional artifacts in their input (relative ordering of the novel concepts). This is consistent with our hypothesis in about 8 out of 10 cases. When this link is not clear and models must instead predict likely properties given a novel concept (i.e., in COMPS), instructions and in-context examples do seem to lead to robust performance. It is important to note that instructions alone do not always account for the observed improvement—LMs' performance on zero-shot settings are consistently still at chance in all cases, suggesting that it is the in-context examples that critically alter models' output distribution to support desirable property inference behavior.

4 Conclusion

In this work, we investigated the extent to which in-context examples and instructions—key components that drive impressive performance in contemporary LMs—can overcome important limitations of LMs at tests that have poked holes in their ability to extract conceptual meaning from text. As a case study, we analyzed how well such experimental contexts can improve LM abilities to perform property inheritance (Murphy, 2002; Misra et al., 2023) in context—binding of novel concepts to existing concepts, and endowing them with valid property inferences as a result. Our findings suggest that mastery of this ability has yet to be robustly achieved, and that LMs in general are still prone to using shallower patterns in their context (when available) rather than systematically extracting conceptual meaning. At the same time, exploring precisely what makes Mistral less susceptible to heuristics will be useful to design more robust LMs, which we leave for future work.

5 Limitations

Single dataset A clear limitation of this work is that it exclusively focuses on a single dataset: COMPS (Misra et al., 2023). So, a question that arises here is to what extent are our findings localized to the chosen dataset vs. meaning-sensitive evaluations in general. This would require a further non-trivial, non-straightforward amount of work, since: (1) different meaning sensitive evaluations focus on different (though equally useful) operationalizations of meaning; and more importantly (2) not all prior work in this area focuses on a standardized and well-defined usage of heuristics that is directly transferable to the experimental setup we have used in this work (following McCoy et al., 2019, 2020; Warstadt et al., 2020b; Mueller et al., 2022; Si et al., 2023).

We do hope that our work contributes to the larger-scale vision of carefully benchmarking different types of meaning extraction abilities in LMs in a controlled manner.

Lack of mechanistic insight Our work continues the long-standing precedent of using carefully constructed behavioral experiments to conclude about the competence of LMs (Linzen et al., 2016; Gulordava et al., 2018; Futrell et al., 2019; Ettinger, 2020; Warstadt et al., 2020a) However, recent works have made impressive strides in localizing the kinds of computations that give rise to the observed behavior in LMs (Hanna et al., 2023; Wang et al., 2023, *i.a.*) Therefore, it is entirely possible that our conclusions about the precise nature of computations carried out by LMs can be greatly strengthened when supplemented by the methods developed in these aforementioned works.

Single Language Finally, this work only focuses on property inheritance problems stated in the English language. This does little to contribute towards diversity in NLP research.

6 Acknowledgments

We thank Tom McCoy and Andrew Lampinen for providing comments on a previous version of the draft. (KM)^2 acknowledge funding from NSF Grant 2104995 awarded to Kyle Mahowald.

References

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the Winograd schema. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10486–10500, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of* *the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Andrew Kyle Lampinen. 2022. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *arXiv preprint arXiv:2210.15303*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

R Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *arXiv preprint arXiv:2203.13112*.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.

Gregory L Murphy. 2002. *The Big Book of Concepts*. MIT press.

Lalchand Pandia and Allyson Ettinger. 2021. Sorting through the noise: Testing robustness of information processing in pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1596, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Sebastian Schuster and Tal Linzen. 2022. When a sentence does not introduce a discourse entity, transformer-based models still sometimes refer to it. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 969–982, Seattle, United States. Association for Computational Linguistics.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. Measuring inductive biases of in-context learning with underspecified demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11289–11310, Toronto, Canada. Association for Computational Linguistics.

Arabella Sinclair, Jaap Jumelet, Willem Zuidema, and Raquel Fernández. 2022. Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050.

Koustuv Sinha, Jon Gauthier, Aaron Mueller, Kanishka Misra, Keren Fuentes, Roger Levy, and Adina Williams. 2023. Language model acceptability judgements are not always robust to context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6043–6063, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Dataset and implementation details

Our experiments use the stimuli from COMPS, released with an MIT License by Misra et al. (2023), but with a modification that involves changing of the nonce words to obey the constraint that the in-context examples all have different nonce word pairs. To this end, we use the following nonce-words:

- **In-context examples:** *wug, dax, fep, zek, blick, toma, kiki, glorp, bova, zup, tufa, flib* (counter-balanced)
- **Test examples:** *gek, wif* (counter-balanced)

A.1 Methodological details

Following COMPS, as well as the precedent set by a number of previous minimal pair analyses (Linzen et al., 2016; Gulordava et al., 2018; Futrell et al., 2019; Wilcox et al., 2019; Warstadt et al., 2020a; Hu et al., 2020), we use a forced choice task to evaluate our LM subjects. Like in COMPS, we compare the log-probability of the property phrase (here, *has a flat tail*) given the choice of left contexts (which indicate whether the right vs. the wrong concept has the property). For example, we measure:

$$\log P_\theta(\textit{has a flat tail} \mid \textit{a } \textbf{\textit{gek}} \textit{ is a beaver. a } \textbf{\textit{wif}} \textit{ is a gorilla. therefore, a } \textbf{\textit{gek}}/\textbf{\textit{wif}}),$$

and for COMPS-QA, we compare the relative probabilities of the two novel concepts given a fixed left prefix which contains a question about the property. For example, we measure:

$$\log P_\theta(\textbf{\textit{gek}}/\textbf{\textit{wif}} \mid \textit{a } \textbf{\textit{gek}} \textit{ is a beaver. a } \textbf{\textit{wif}} \textit{ is a gorilla. } \textbf{\textit{Question:}} \textit{ Which one of them has a flat tail? } \textbf{\textit{Answer:}})$$

In both cases above, ***gek*** is the concept that should inherit the property. While these examples show the zero-shot case, cases with in-context examples and instructions simply add more context to the prefix, therefore the surface form of the output space remains the same regardless of the number of in-context examples or the presence of instructions.

Log-probabilities for all models were accessed using `minicons` (Misra, 2022),[2] a library that wraps around `transformers` (Wolf et al., 2020) by huggingface, and is written in `pytorch`. For our experiments with Llama-13B, we quantize the model to 4-bits in order to fit it onto a single GPU. All experiments were run on a cluster with 4 NVIDIA A40 GPUs, though each individual experiment on a model was computed on a single A40 GPU.

A.2 Model Metadata

Table 1 shows the LMs used in this work, along with their total parameters, tokens encountered during training, and vocabulary size.

B Instructions

Tables 2, 3, 4, 5 show our instruction templates.

C Fine-grained results

While Figure 2 shows results aggregated over both types of heuristics that we have used in this work, we additionally display finer-grained, heuristics-wise results in this section. Again, in each of these

[2] `https://github.com/kanishkamisra/minicons`

Model	Params	Pre-training Tokens	Vocab size
GPT-2 XL	1.5B	8B	50,257
OPT-6.7b	6.7B	180B	50,272
Llama-2-7b	7B	2T	32,000
Llama-2-13b	13B	2T	32,000
Mistral-7b	7B	?	32,000

Table 1: Overview of the LMs used in this work. '?' indicates that the given value was not made available in the LM's release.

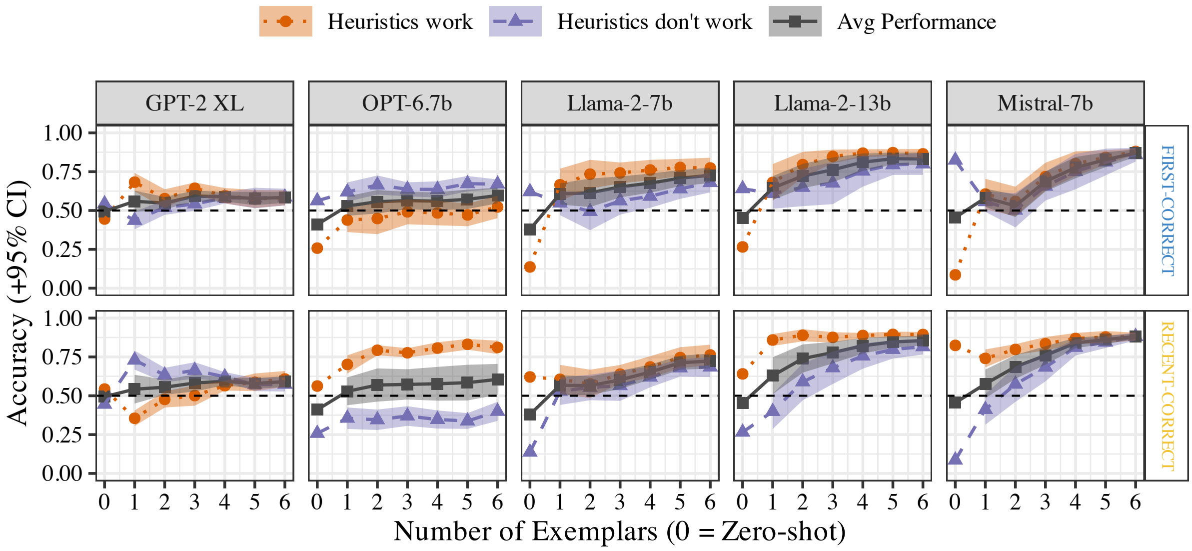
COMPS version	Instruction Template
COMPS	Given a pair of statements that introduce novel entities as types of real world animals, **write a true statement about the properties of the novel entities:** `{exemplars}` (omitted in zero-shot) `{test-stimulus}`
COMPS-QA	Given a pair of statements that introduce novel entities as types of real world animals, **answer the question that follows**: `{exemplars}` (omitted in zero-shot) `{test-stimulus}`

Table 2: Instructions for COMPS and COMPS-QA with instruction type: "`minimal`"

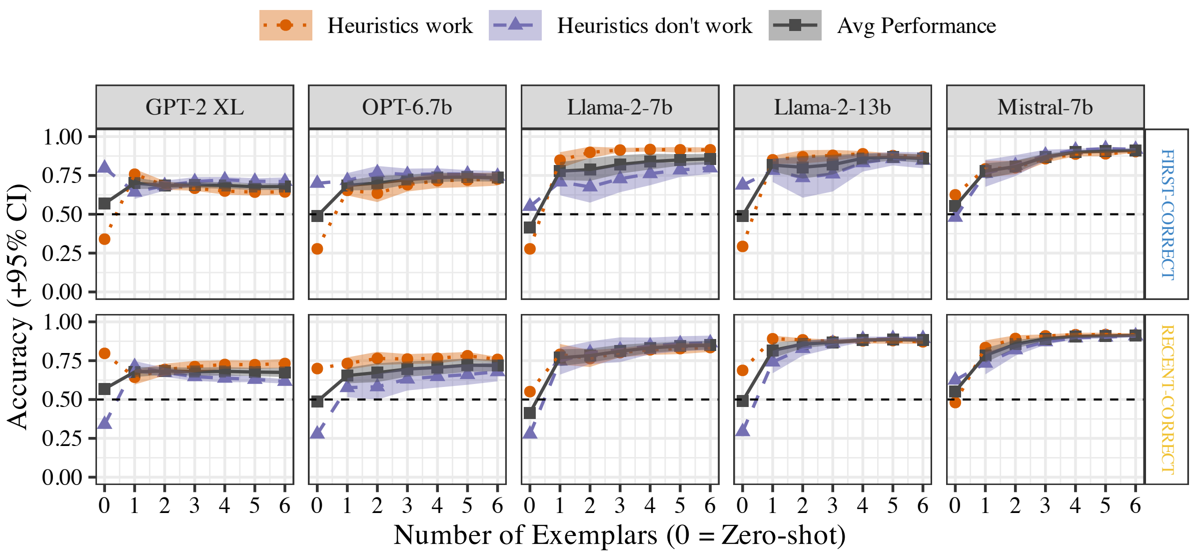
COMPS version	Instruction Template
COMPS	Some aliens have come to earth, and it turns out they have their own language for talking about our animals here on Earth. Your job is to help the aliens learn about our Earthling animals **by giving them some information about the animals.** Let's get started: `{exemplars}` (omitted in zero-shot) `{test-stimulus}`
COMPS-QA	Some aliens have come to earth, and it turns out they have their own language for talking about our animals here on Earth. Your job is to help the aliens learn about our Earthling animals **by answering some questions about them.** Let's get started: `{exemplars}` (omitted in zero-shot) `{test-stimulus}`

Table 3: Instructions for COMPS and COMPS-QA with instruction type: "`aliens`"

plots, the extent to which a model relies on heuristic is indicated by the gap between the dotted (●) and the dashed (▲) lines. This is now separately shown for each of our heuristics. Figure 3 shows results on COMPS with and without instructions for both the heuristics, and similarly Figure 4 shows results on COMPS-QA with and without instructions for both the heuristics.

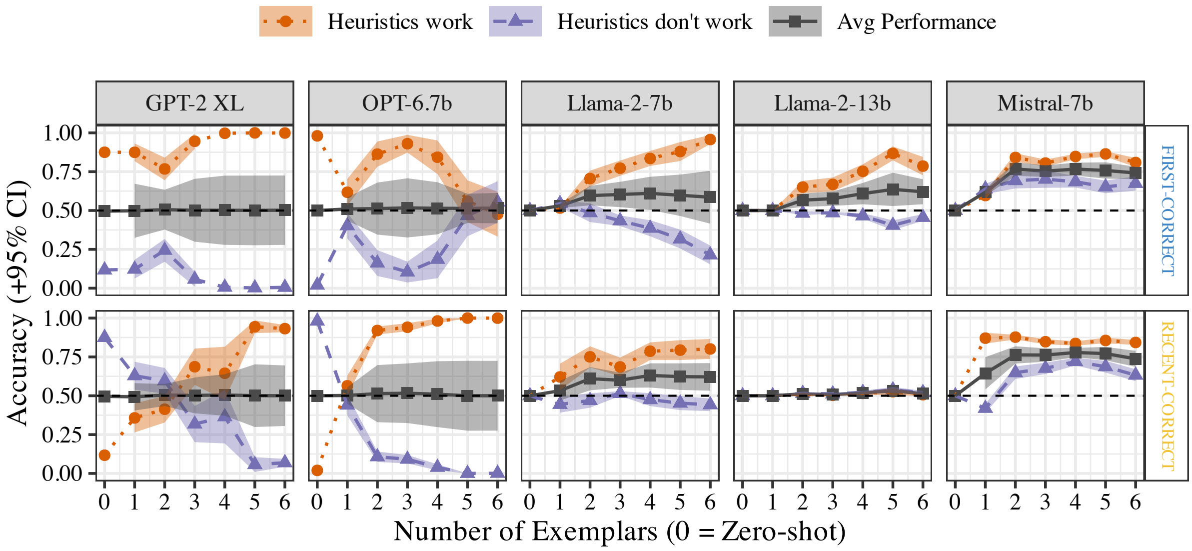


(a) COMPS

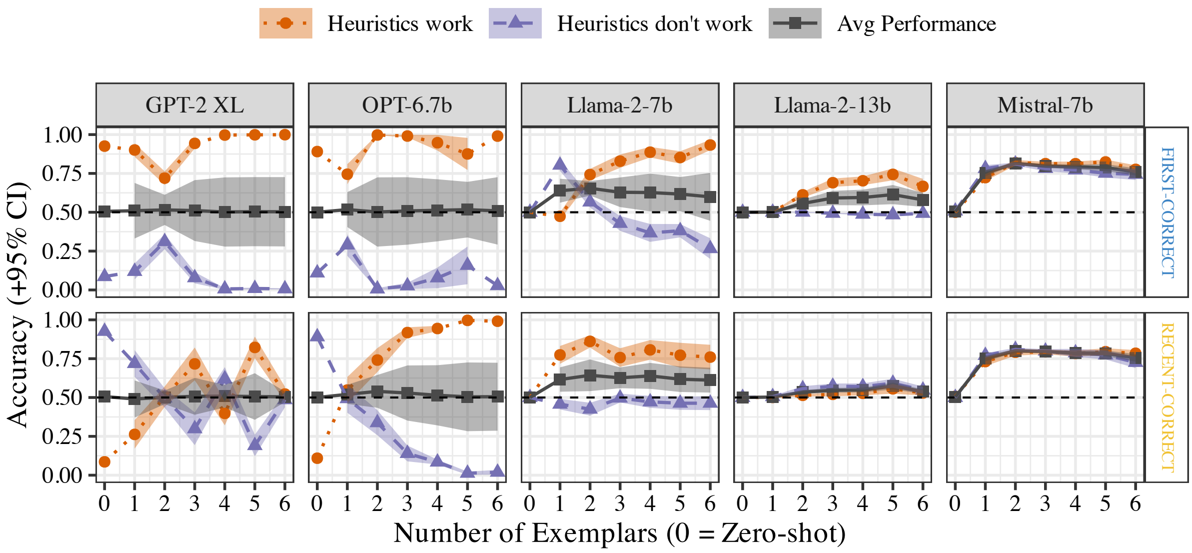


(b) COMPS with Instructions

Figure 3: Fine-grained results on COMPS as a function of the number of in-context examples (with and without instructions).



(a) COMPS-QA



(b) COMPS-QA with Instructions

Figure 4: Fine-grained results on COMPS-QA as a function of the number of in-context examples (with and without instructions).

COMPS version	No. of shots	Instruction Template
COMPS	Zero-shot	It is important to know and reason about the properties of entities in the world. The following is a pair of premise statements that introduce novel entities as new types of real world animals. Your task is to make a conclusion about the properties of one of the entities by reasoning over the premise statements. `{test_stimulus}`

	Few-shot	It is important to know and reason about the properties of entities in the world. The following example(s) show a pair of premise statements that introduce novel entities as new types of real world animals, followed by another statement that attributes a property to one of the entities introduced in the premise statements.
		
		Examples:
		`{examples}`
		
		Here is another pair of premise statements. Your task is to make a conclusion about the properties of one of the entities by reasoning over the premise statements.
		`{test_stimulus}`

COMPS-QA	Zero-shot	It is important to know and reason about the properties of entities in the world. The following is a pair of premise statements that introduce novel entities as new types of real world objects. The statements are followed by a question that asks which novel entity in the premise can a specific property can be attributed to. Answer the question by reasoning over the premise statements. `{test_stimulus}`

	Few-shot	It is important to know and reason about the properties of entities in the world. The following example(s) show a pair of premise statements that introduce novel entities as new types of real world animals. The statements are followed by a question that asks which novel entity in the premise can a specific property can be attributed to, and the answer to the question, obtained by reasoning over the premise statements.
		
		Examples:
		`{examples}`
		
		Here is another pair of premise statements. Answer the question that follows.
		`{test_stimulus}`

Table 4: Instructions for COMPS and COMPS-QA with instruction type: "`Detailed-1`"

COMPS version	No. of shots	Instruction Template
COMPS	Zero-shot	It is important to know and reason about the properties of entities in the world. The following is a pair of premise statements that introduce novel entities as new types of real world animals. **Your task is to write a true statement about the properties of the novel entities.** `{test_stimulus}`

	Few-shot	It is important to know and reason about the properties of entities in the world. The following example(s) show a pair of premise statements that introduce novel entities as new types of real world animals, followed by another statement that attributes a property to one of the entities introduced in the premise statements.
		
		Examples:
		`{examples}`
		
		Here is another pair of premise statements. **Your task is to write a true statement about the properties of the novel entities.**
		`{test_stimulus}`

COMPS-QA	Zero-shot	It is important to know and reason about the properties of entities in the world. The following is a pair of premise statements that introduce novel entities as new types of real world animals. The statements are followed by a question that asks **which of the introduced entities** a specific property can be attributed to. Answer the question by reasoning over the premise statements. `{test_stimulus}`

	Few-shot	It is important to know and reason about the properties of entities in the world. The following example(s) show a pair of premise statements that introduce novel entities as new types of real world animals. The statements are followed by a question that asks **which of the introduced entities** a specific property can be attributed to, and the answer to the question, obtained by reasoning over the premise statements.
		
		Examples:
		`{examples}`
		
		Here is another pair of premise statements. Answer the question that follows.
		`{test_stimulus}`

Table 5: Instructions for COMPS and COMPS-QA with instruction type: "`Detailed-2`"