TASK AMBIGUITY IN HUMANS AND LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models have recently achieved strong performance across a wide range of NLP benchmarks. However, unlike benchmarks, real world tasks are often poorly specified, and agents must deduce the intended behavior from a combination of context, instructions, and examples. We investigate how both humans and models behave in the face of such task ambiguity by proposing AmbiBench, a new benchmark of six ambiguously-specified classification tasks. We evaluate humans and models on AmbiBench by seeing how well they identify the intended task using 1) instructions with varying degrees of ambiguity, and 2) different numbers of labeled examples. We find that the combination of model scaling (to 175B parameters) and reinforcement learning from human feedback (RLHF) enables models to approach or exceed the accuracy of human participants across tasks, but that either one of these alone is not sufficient. In addition, we show how to dramatically improve the accuracy of language models trained without RLHF by finetuning on a small number of ambiguous in-context examples, providing a promising direction for teaching models to generalize well in the face of ambiguity.

1 INTRODUCTION

Language models have recently been applied to a wide range of NLP benchmarks, ranging from question answering, summarization, and logical reasoning, to solving riddles, dark humor detection, and ASCII word recognition (Brown et al., 2020; Srivastava et al., 2022). Performance across tasks has improved as models and datasets have grown in size, raising the prospect of a route towards generalist NLP models with broad utility.

However, one feature many of these benchmarks share is that they are carefully designed to make the desired task very clear to the language model, since this is a prerequisite for establishing performance on that task. Unfortunately, real-world uses of language models are not likely to feature such thought and clarity in their task specification. Rather than iterating over and perfecting a specification for their tasks, everyday users of language models may wish to effortlessly define tasks on

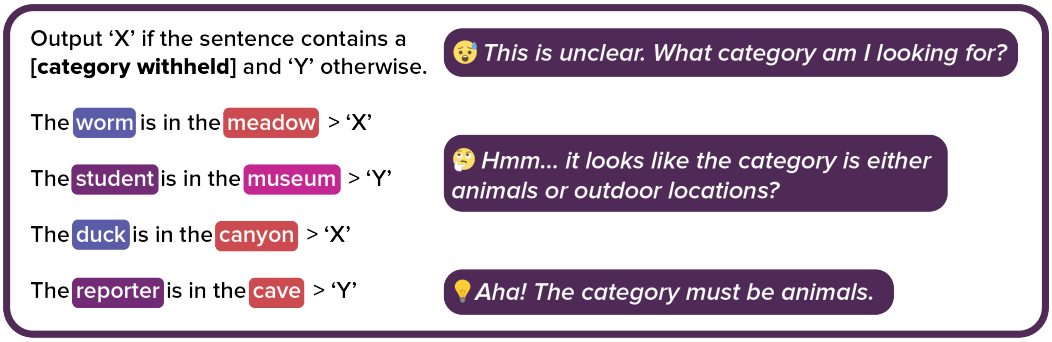


Figure 1: Complex tasks are often hard to specify precisely, leaving important pieces of information missing. Agents should be able to fill in the blanks by combining information from instructions and examples in order to identify the intended behavior.

an as-needed basis, without worrying that they will be misunderstood. More pressingly, in complex domains featuring high-dimensional inputs and outputs (e.g. programming, verification, generation) it is unlikely that even a thoughtful task specification will manage to perfectly capture all the features of an input and output which are salient or not salient to the task. This is especially important for safe and robust deployment of language models, as such undesirable dependencies can be hidden hazards that are only revealed when a model fails catastrophically in a new setting (Geirhos et al., 2020).

To operationalize this problem, we introduce AmbiBench, a new benchmark of six ambiguously-specified tasks. Each input in AmbiBench is a sentence (e.g. *The dog is in the meadow*) that has multiple associated classification tasks based on different linguistic features (e.g. *contains an animal*, *contains an outdoor location*). Task ambiguity arises when more than one task is consistent with the provided instructions or labeled examples.[1]

We establish how well different models and humans perform on a task given a wide range of task specifications, including clear vs unclear instructions, and zero vs many examples. We find that the largest models trained with reinforcement learning from human feedback (RLHF) approach or outperform humans across all specifications we try.

We also show how to improve standard language models' performance by finetuning on a small set of in context examples that demonstrate the desired generalization. This form of meta-learning dramatically improves a model's ability to learn new ambiguously-specified tasks, but crucially *only* when the finetuning tasks are ambiguous, not for control tasks of a similar format that are unambiguous. This suggests a possible mechanism for why RLHF models outperform standard language models (discussed in Section 4.4), as well a promising direction for improving how models learn tasks in ambiguous contexts.

To summarize our contributions, we:

1. Introduce and motivate the problem of studying task ambiguity in large language models
2. Evaluate humans and models on a new benchmark of ambiguously-specified tasks, demonstrating that while most models fail to disambiguate the intended task well, sufficiently-large models trained with RLHF are able to approach or even exceed the performance of our human participants to resolve the ambiguity between tasks
3. Show how finetuning on ambiguous in-context prompts and examples can enable traditional language models to surpass the performance of RLHF models when evaluated on unseen tasks, providing a promising route towards models that capably manage task ambiguity

2 Related Work

2.1 Ambiguity in natural language processing

Ambiguity is a well-studied topic in NLP, with work spanning topics as diverse as search queries (Cronen-Townsend & Croft, 2002; Wang & Agichtein, 2010), question answering (Min et al., 2020; Zhang & Choi, 2021), named entities (Bunescu & Pasca, 2006; Cucerzan, 2007; Dredze et al., 2010), coreference resolution (Webster et al., 2018), machine translation (Stanovsky et al., 2019), and information-seeking dialogues (Aliannejadi et al., 2019; Guo et al., 2021; Aliannejadi et al., 2021; Sun et al., 2022; Wu et al., 2022).

Our work differs from these prior streams of work by studying *task ambiguity* (Finn et al., 2018; Tamkin et al., 2022b), where the task the agent is being asked to perform is ambiguous, rather than an ambiguous input for a clearly specified task. This is of special relevance for self-supervised learning models that can be easily adapted to a broad range of downstream tasks (Bommasani et al., 2021). In these settings, models must infer the correct task from a user's specification, as opposed to a possibly unsafe or undesirable task that is also consistent with that specification.

[1] Importantly, task ambiguity is distinct from clearly-specified tasks with *ambiguous inputs*, e.g. determining the referent of the pronoun in sentences like *the nurse handed the doctor* ***her*** *phone*. Here, the task is clear (determine who **her** refers to), but there is not enough information in the input to answer it.

2.2 IN-CONTEXT LEARNING AND PROMPTING

Task ambiguity is especially relevant for language models, which can be adapted for many different tasks via in-context learning (Brown et al., 2020; Tamkin et al., 2021a; Bommasani et al., 2021; Liu et al., 2022b). Much work has attempted to improve the ability of such models to perform in-context learning by calibrating model predictions (Zhao et al., 2021), choosing good examples for the prompt (Liu et al., 2022a), finetuning models on natural language descriptions of tasks (Zhong et al., 2021; Wei et al., 2022; Sanh et al., 2022), or by training models with reinforcement learning from human feedback (Bai et al., 2022; Ouyang et al., 2022).

Prior work has suggested that language models may not effectively learn from the provided instructions (Webson & Pavlick, 2022) or few-shot examples (Min et al., 2022b; Kim et al., 2022); instead such models may be relying on cues such as the formatting of the examples or the label space. In this work, we present a way to measure how well models use instructions or few-shot examples irrespective of these cues, because each AmbiBench example is consistent with *multiple* possible tasks. Thus, models that perform well on AmbiBench must infer the desired task using additional information, such as the task instruction or other examples.

Past work has also explored finetuning on in-context learning examples (Chen et al., 2022; Min et al., 2022a). We extend this line of work to show how the content of these training examples can dramatically affect generalization: Finetuning on *ambiguously-specified* examples (but not a control set of unambiguous tasks) can enable the model to better disambiguate unseen task specifications—vastly improving the performance of traditional language models without the need for reinforcement learning from human feedback.

2.3 TASK AMBIGUITY

Systems capable of performing different tasks may experience *task ambiguity*, where the provided examples do not uniquely identify the user's intent (Finn et al., 2018; Tamkin et al., 2021a). One form of task ambiguity is shortcut learning (Geirhos et al., 2020), where the training examples can all be solved by identifying a simple feature (e.g. a watermark) as opposed to learning the intended task (e.g. object classification). Task ambiguity is particularly important in few-shot learning settings, where the small number of examples provided may leave the intended task ambiguous (Finn et al., 2018; Tamkin et al., 2021a). In this work, we study task ambiguity for in-context learning with language models and humans, considering not only the role of examples but also natural language instructions.

3 THE AMBIBENCH BENCHMARK

As a first step towards studying task amibiguity in language models, we construct the AmbiBench benchmark, a collection of six different sentence classification tasks. The goal of AmbiBench is to construct a testbed of *minimal complexity* where we can control and measure the degree of ambiguity in various task specifications. Despite the simplicity of this benchmark, we find large variability in performance across different language models.

3.1 SELECTION OF TASKS

AmbiBench contains six binary classification tasks, where a human or model must detect a simple linguistic feature in an input sentence—for example, whether an outdoor location or an animal was mentioned—and then *output* the appropriate classification letter (`X` or `Y`). Crucially, however, each sentence has two linguistic features (e.g. *The duck is in the canyon* has the features *animal* and *outdoor location*). The six features are grouped into three pairs, shown in Table 1, where a single sentence will have one feature in each pair.

Identifying the *salient feature* for the task thus requires either the presence of an informative instruction (e.g., *Output 'X' if the sentence contains an outdoor location and 'Y' otherwise*) or multiple labeled examples to identify which feature determines the label.

Tasks were chosen to represent a set of common semantic categories, excluding subtoken information such as periods and capitalization that might be much easier for humans to represent than

Salient feature	Example sentence
human subject	The **researcher/bear** is in the museum.
indoor location	The researcher is in the **museum/meadow**.
religious leader	He is in the museum with the **rabbi/judge**.
pronoun gender	**He/She** is in the museum with the judge.
proper noun	**Paul Atreides/The director** may not be in the film studio.
negation	Paul Atreides **may/may not** be in the film studio.

Instruction	Example
Uninformative	*Output 'X' if the sentence contains a **[category withheld]** and 'Y' otherwise.*
Informative	*Output 'X' if the sentence contains a **proper noun** and 'Y' otherwise.*

Table 1: **The AmbiBench benchmark**. *Left*: Each task involves detecting a salient feature in a sentence (bolded in the examples on the right). The same sentence could potentially receive a label according to two features, requiring a learner to use additional information (task instructions or other examples) to disambiguate the intended behavior. *Right:* Varying levels of instruction are inserted before the examples, providing different degrees of information about the format and salient feature of the task. See Figure 1 for an example of a complete prompt.

models. See Figure 1 for an example of this disambiguation process, and Table 1 for a full list of tasks and accompanying instructions.

3.2 Task construction

AmbiBench examples are programmatically constructed from a set of templates, allowing precise control over the amount of task ambiguity in each in-context example (see Table 1 and Appendix G for more details). Templated data has seen a recent resurgence in NLP for the purposes of evaluating large language models (Lake & Baroni, 2018; Srivastava et al., 2022), as they enable precise control and coverage over different variables of study. Furthermore, recent work has shown strong correlation between test performance on synthetic and naturalistic data Liu et al. (2021), suggesting that insights gained from such datasets may extend to a broader range of natural contexts. In our case, this dataset construction process enables us to formalize and characterize the degree of task ambiguity in different examples, allowing us to measure how well models can disambiguate between multiple potential classification tasks they may be asked to perform.

3.3 In-context learning formats

There are several ways an instruction and in-context examples can be assembled into a prompt for a language model. Given the demonstrated sensitivity of models to such parameters (Zhao et al., 2021; Liu et al., 2022b; Lu et al., 2022), we consider two different prompt formats, and report averaged performance across them:

Arrow:

```
Output 'X' if the sentence
contains an outdoor location
and 'Y' otherwise.
The worm is in the meadow
>X
The duck is in the canyon
>Y
...
```

Q/A:

```
Output 'X' if the sentence
contains an outdoor location
and 'Y' otherwise.
Q: The worm is in the meadow
A: X
Q: The duck is in the canyon
A: Y
...
```

4 EXPERIMENTS

We use AmbiBench to investigate how humans and language models respond to and resolve different manifestations of task ambiguity.

4.1 EXPERIMENTAL SETUP

First, we describe the different language models and human participants we study, and how we evaluate them.

Language models We examine a range of different models, including both OpenAI's normal language models and their "instruct" models trained with reinforcement learning from human feedback (RLHF) (Brown et al., 2020; Ouyang et al., 2022). In the rest of the paper, OpenAI's model names are reported as listed in their documentation [2] (e.g. `davinci`, `text-curie-001`). The instruct models have a numerical suffix (e.g. `002`) and the model size increases as one progresses through the alphabet (`ada`, `babbage`, `curie`, `davinci`). See Appendix E for more information. We also evaluate AI21 Studio's 178B-parameter Jurassic-1 Jumbo language model (`jurrasic-jumbo`) (Lieber et al.), as well as the 11B-parameter T0++ model (`t0pp`) (Sanh et al., 2022), which was finetuned on a large corpus of task instructions, but not trained with RLHF. This diversity of model providers, model sizes, and training strategies enables us to identify which ingredients are most crucial for resolving task ambiguity.

Human evaluation We compare model performance with the performance of human participants, evaluated by hiring contractors from Prolific (Palan & Schitter, 2017). We aimed to evaluate the human participants as similarly to language models as possible within the confines of an online survey methodology. We showed human participants exactly the same input that language models received, with minimal additional information presented to them before the study began. Participants typed the answer label (i.e. `X` or `Y`) into a textbox, as opposed to choosing from a set of preselected options, to mitigate priming effects and mirror the setting for language models. We also recruited a new participant for every single in-context instance, to avoid humans learning across examples in ways that language models do not. Human participants were paid $12-13/hr, in line with Prolific wage recommendations.[3]. See Appendix F for more details.

4.2 TASK DISAMBIGUATION USING NATURAL LANGUAGE INSTRUCTIONS

One way that people resolve task ambiguity is through the use of natural language instructions, which can explicitly indicate different aspects of the task. Past work has suggested that the best models do not fruitfully use natural-language instructions, as evidenced by experiments leveraging irrelevant or misleading directions (Webson & Pavlick, 2022). However, these experiments were performed for established natural language processing tasks that lack the explicit task ambiguity we study here, and did not investigate more recent models trained with reinforcement learning from human feedback (Bai et al., 2022; Ouyang et al., 2022).

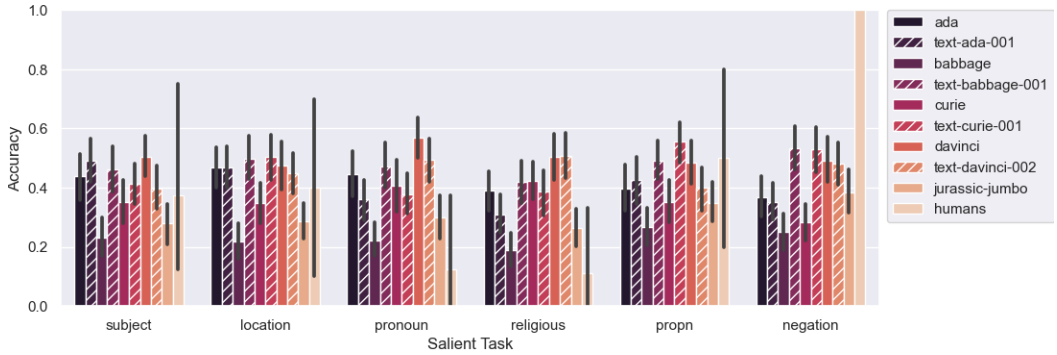
As a first set of experiments, we evaluate how humans and models are able to use differing levels of instruction to resolve task ambiguity. The humans and models receive two in-context examples, one from each class. Humans and models are then presented with a third query example in order to elicit the predicted output letter. Because there is only one example of each class, but two possible features, the salient feature can not be identified from these two examples alone, requiring the model to use the instruction to disambiguate the task. The order of the examples, the example format, as as well as the assignment of each class to an output letter (`X` or `Y`) are randomized. Each model is evaluated with 720 different in-context prompts for each level of instruction.

We consider two different levels of instruction:

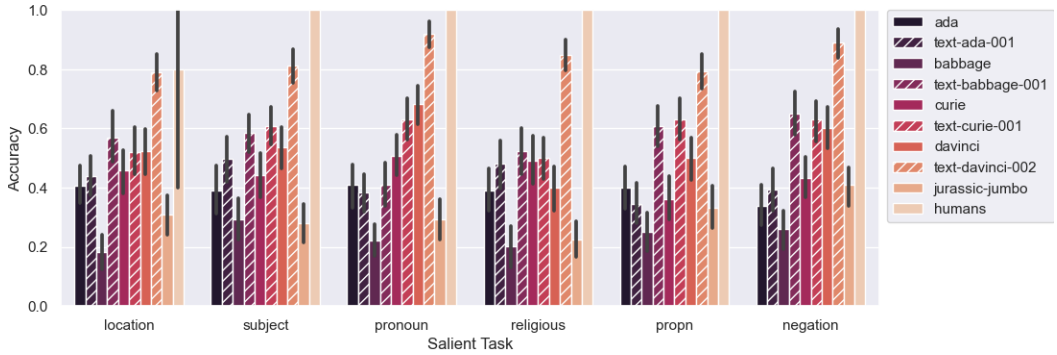
1. **Informative instruction**: The model receives a full specification of the salient feature and output format. Ex: *Output 'X' if the sentence contains an animal and 'Y' otherwise.*

[2] https://beta.openai.com/docs/

[3] https://www.prolific.co/pricing



(a) Uninformative Instructions



(b) Informative Instructions

Figure 2: **The largest RLHF model (`text-davinci-002`) approaches human accuracy for both uninformative and informative instructions.** Accuracy of humans and other models for tasks prompted with an instruction and two in-context examples. Error bars show 95% bootstrap CIs.

2. **Uninformative instruction**: The model receives the output format but the salient feature is redacted. Ex: *Output 'X' if the sentence contains a [category withheld] and 'Y' otherwise.*

Our setting is simple enough that crafting an informative instruction is not challenging, making it tractable for us to study. However, the insights from this simple case may generalize to more complex tasks language models may be used for, where operators may be more likely to inadvertently omit crucial information.

4.2.1 RESULTS

In the case of **uninformative instructions**, humans as well as many models are able to achieve approximately 50% accuracy by correctly understanding the output format and choosing `X` and `Y` at random. However, some non-instruct models, including `jurassic-jumbo`, `ada`, `babbage`, and `curie`, often output values other than `X` or `Y` (e.g. `Z`), leading to lower performance. Finally, in the case of negation, humans achieve 100% accuracy despite lacking an instruction identifying the salient feature. This may be due to an inductive bias present in people (but not models) that makes negation an especially salient feature.

In the case of **informative instructions**, humans perform the strongest at this task, with perfect performance in all but one task, showing that they are broadly able to identify the salient feature in the text inputs and output the correct letter. Humans are closely followed by the `text-davinci-002` RLHF model (see Figure 2). All other models perform relatively poorly, including the non-RLHF 175B+ parameter `davinci` and `j1-jumbo` models, as well as the smaller RLHF models `curie`, `babbage`, and `ada` (although we verify in Section that these models are still able to generate outputs in the correct format). This seems to suggest that most models are not reliably able to follow simple instructions to disambiguate a task, but that a *combination* of large-scale training and RLHF can approach human performance in some settings.

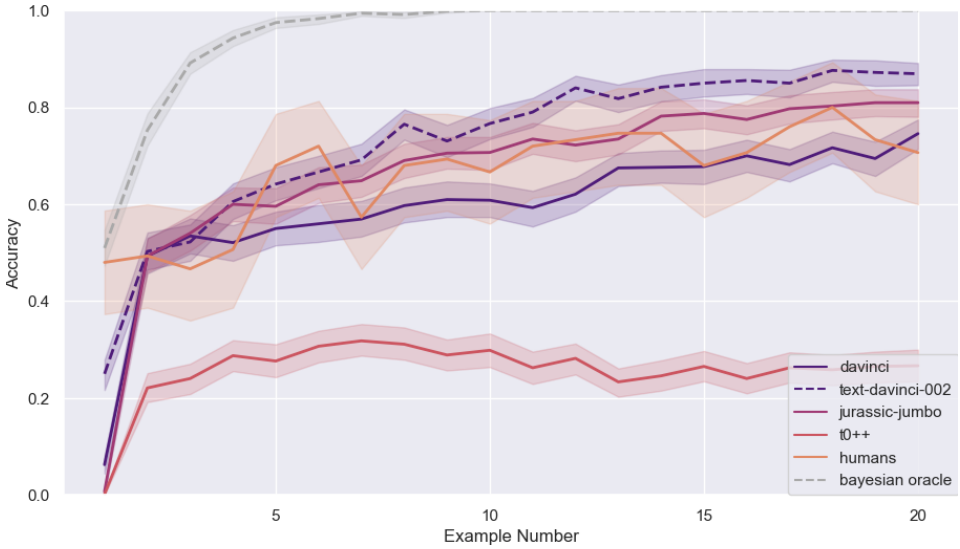


Figure 3: **The largest RLHF model (`text-davinci-002`) outperforms human participants at disambiguating the intended task.** Accuracy as the number of examples in the in-context window grows. Surprisingly, the smaller `curie` model reliably outperforms the larger `davinci` model across the examples. In addition, the RLHF training hurts at `curie` scale, but dramatically helps at `davinci` scale. Shaded regions are 95% bootstrap CIs.

4.3 Task disambiguation using multiple examples

While instructions are a simple way to specify a task, multiple examples can also disambiguate between different tasks a user might intend. For example, if there are multiple features that could explain the label of a single example, more examples will gradually identify the salient feature provided the features are sufficiently decorrelated.

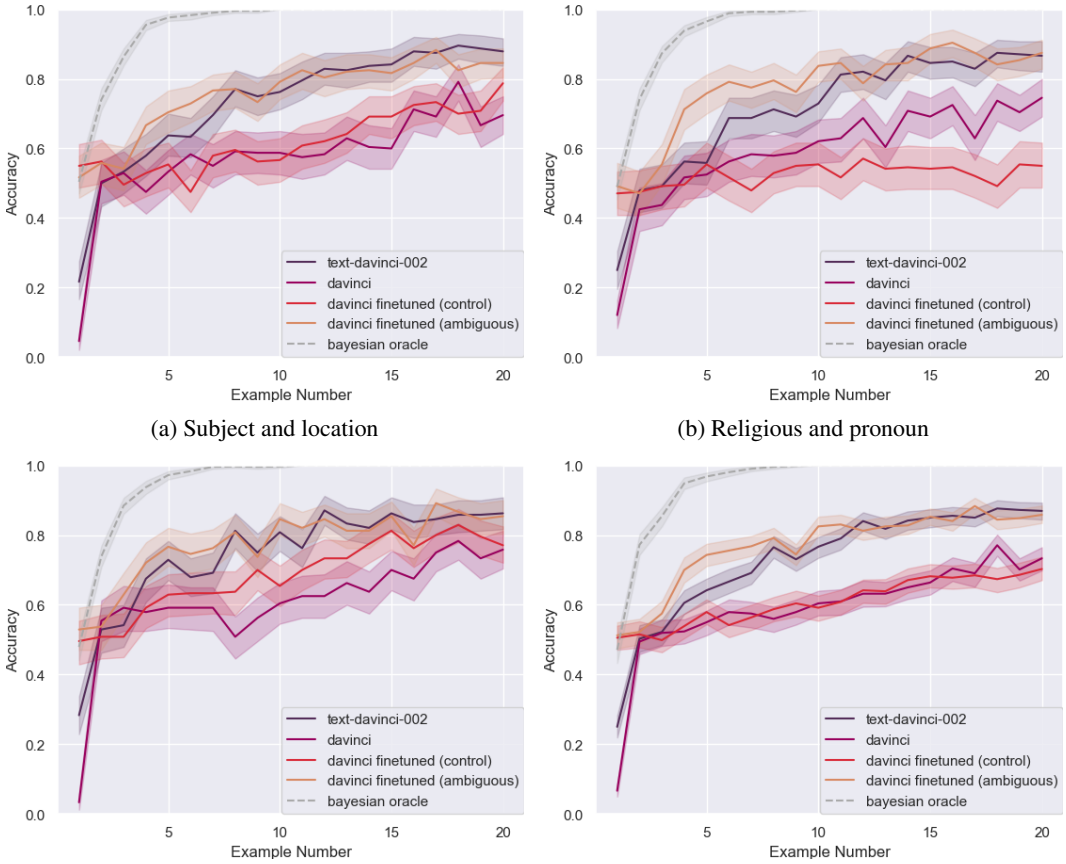
We investigate whether models and humans can identify the salient features in AmbiBench as the number of examples grows from zero (where the task is completely ambiguous) to twenty (where the task is almost certainly unambiguous). Both models and human participants predict the answer for each example, then are presented with the correct answer for that example and the next query.[4] All aspects of the examples are randomized, including the salient feature (chosen randomly, then held constant across the entire in-context example), the assignment of `X` or `Y` to the salient feature, the example order, and the specific instantiations of the salient and non-salient features for each example. Each model is evaluated with 720 different in-context prompts, each containing 20 examples.

We also compare human and models with a Bayesian oracle that represents how well an optimal learner could perform on the benchmark. This oracle performs perfectly as soon as it sees a set of examples which disambiguate the intended task, and performs at chance otherwise.

4.3.1 Results

To our surprise, the best language model (the RLHF-trained `text-davinci-002`) significantly outperformed the human participants (Figure 3). The human participants performed comparably to the `j1-jumbo` and `curie` models, which in turn performed better than the rest of OpenAI's models. The `t0pp` models exhibited large sensitivity to the prompt format—the model outputted invalid answers (typically nouns) for the **arrow** format. However, considering only the **Q/A** format, `t0pp` still only performed near chance. All models considerably underperform the Bayesian oracle, suggesting room for additional improvement. See Appendix A for initial experiments investigating whether models can describe the few-shot task in words.

[4]For causal language models, such as OpenAI's models and J1-Jumbo, which only attend backwards, this can be done efficiently by presenting the full 20 examples to the model and looking at the probability assigned to the correct answer for each example.



(a) Subject and location

(b) Religious and pronoun

(c) Proper noun and negation

(d) Average across (a-c)

Figure 4: **Finetuning on ambiguous in-context examples dramatically improves accuracy on unseen tasks that are ambiguously specified**. Accuracy after finetuning `davinci` on ambiguous and non-ambiguous (control) in-context examples. Models are finetuned on 272 examples from four tasks, then evaluated on the two held-out tasks (subfigure captions). Shaded regions are 95% bootstrap CIs.

We do not observe evidence that the imperfect human performance is due to low-quality participants or bot activity—human annotators mostly spent between 4 and 8 minutes on the task, did not appear to be guessing at random, and typically left thoughtful comments or feedback on the survey. That said, we caution against claims of "superhuman performance" given that annotators represent merely a sample from a single distribution of humans, and they may have experienced fatigue or distraction across the 20-example episode.

4.4 Finetuning a model to generalize well in the face of ambiguity

The strong performance of the RLHF models relative to the normal language models in Section 4.3 is somewhat surprising—these models are described as being trained to follow human instructions, not to resolve ambiguity in instructions by analyzing the training examples. While the training dataset of these models was not released, Ouyang et al. (2022) do report that some of the crowdsourced examples for the model contain instructions along with few-shot examples. If some of these instructions were ambiguous, the model may have learned from those examples to resolve that ambiguity more effectively.

Motivated by this hypothesis, we investigate whether finetuning on a small corpus of ambiguous in-context learning examples is sufficient to close the gap between the best-performing `text-davinci-002` RLHF model and the normal `davinci` language model. To do so, we partition the six AmbiBench tasks into three folds, each containing four finetuning tasks and two

evaluation tasks (following the feature pairs in Table 1). We finetune on 68 examples from each task (two for each number of examples, from 4 to 20), and evaluate on 240 examples randomly drawn from the other two tasks. While all tasks share some structural similarities, this partitioning ensures that the model is being tested on held-out features that never appeared in its finetuning dataset. Models are finetuned using the OpenAI API (see Appendix H for details).

As an added control to see whether ambiguity truly matters in the finetuning dataset, we also finetune on the same splits of data, but where only one feature varies within each in-context example. For example, if the two features were animal and indoor location, a given in-context example may contain examples with both animals and humans, but only indoor locations. See Appendix H for more details.

4.4.1 Results

Despite the small training dataset consisting of only 4 tasks (with 272 examples total), we find we are able to completely close the gap between the RLHF models and our finetuned models across all three splits of our data. Indeed, our finetuned models appear to even outperform `text-davinci-002` across the first eight examples, closing part of the gap to the Bayesian oracle.

Crucially, we do not observe any improvement for the control finetuned models, which were finetuned on the same kinds of examples but without task ambiguity between two potential salient features. This indicates that ambiguity is the crucial ingredient explaining the success of our finetuned models, and supports the hypothesis that the few-shot examples in `text-davinci-002`'s RLHF data may contribute to its strong performance.

More broadly, these results suggests that explicitly finetuning models to adapt to task ambiguity may result in a generalized capacity to do so across different kinds of ambiguous task specifications.

5 Discussion and Conclusion

We present the AmbiBench testbed for studying task ambiguity in language models and humans, showing how it can be used to investigate different factors influencing task ambiguity, as well as identify promising interventions that can improve how models resolve it.

5.1 Limitations

Our study has several limitations. First, we conduct a scientific and controlled study of task ambiguity in language models; this naturally elides many of the messy nuances of task ambiguity in the real world, and should be seen as complementary to in-the-wild case studies. We explore one such real-world use case in Appendix B, however more work is needed. Second, despite our efforts to match the experimental conditions between humans and language models, humans do require some additional instructions to orient them to the task interface, and may suffer from fatigue and uneven concentration across the length of a 20-example learning episode. Finally, our work studies of task ambiguity between two possible tasks—however, in general task ambiguity may occur between arbitrary many tasks, or even an infinitely large *family* of tasks.

5.2 Future Work

Task ambiguity is a pressing problem in machine learning with relevance for safety, fairness, and interpretability. Going forward, we are excited by the potential to study task ambiguity in self-supervised models trained on many different modalities (Reed et al., 2022; Tamkin et al., 2021b; 2022a; Alayrac et al., 2022), including multimodal settings, as self-supervised learning is applied increasingly broadly. The strong performance of our models on the AmbiBench testbed also suggests the tractability of studying task ambiguity in more complex real-world settings where language models are used, such as software engineering, law, and education, as well as assessing the efficacy of our proposed finetuning interventions.

Ethics statement Our research makes use of human subject experiments via the Prolific platform (Palan & Schitter, 2017). We pay workers a minimum of $12-13 / hour, consistent with Prolific wage recommendations.[5]. We also made efforts to solicit feedback from participants via pilot studies, which led to several changes to the research methodology to make the survey experience more pleasant (e.g. keyboard shortcuts to navigate the study more efficiently). Anecdotally, many participants expressed that they enjoyed the study:

1. *Zap! I'm suddenly back in high school with Ms. Langston's English class. Thank you for the smiles!*
2. *this was fun!*
3. *this was a really good study*

Reproducibility statement We include detailed experimental settings in Sections 1, 4.2, 4.3, 4, E, H, G. We will also release our codebase, including the benchmark data, to enable others to evaluate their own models and build on our work.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

Mohammad Aliannejadi, Hamed Zamani, Fabio A. Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeffrey Dalton, and Mikhail S. Burtsev. Building and evaluating open-domain dialogue corpora with clarifying questions. *ArXiv*, abs/2109.05794, 2021.

Yushi Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862, 2022.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William

[5]https://www.prolific.co/pricing

Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258, 2021.

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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. *ArXiv*, abs/2110.07814, 2022.

Steve Cronen-Townsend and W. Bruce Croft. Quantifying query ambiguity. 2002.

Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP*, 2007.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Timothy W. Finin. Entity disambiguation for knowledge base population. In *COLING*, 2010.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *NeurIPS*, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2:665–673, 2020.

M. Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. Abg-coqa: Clarifying ambiguity in conversational question answering. In *AKBC*, 2021.

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, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang goo Lee, Kang Min Yoo, and Taeuk Kim. Ground-truth labels matter: A deeper look into input-label demonstrations. *ArXiv*, abs/2205.12685, 2022.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018.

O Lieber, O Sharir, B Lentz, and Y Shoham. Jurassic-1: Technical details and evaluation, white paper, ai21 labs, 2021. *URL: https://uploads-ssl. webflow. com/60fd4503684b466578c0d307/61138924626a6981ee09caf6_jurassic_ tech_paper. pdf*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *DEELIO*, 2022a.

Nelson F. Liu, Tony Lee, Robin Jia, and Percy Liang. Can small and synthetic benchmarks drive modeling innovation? a retrospective study of question answering modeling approaches. *ArXiv*, abs/2102.01065, 2021.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys (CSUR)*, 2022b.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, 2022.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. In *EMNLP*, 2020.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *ArXiv*, abs/2110.15943, 2022a.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *ArXiv*, abs/2202.12837, 2022b.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Stefan Palan and Christian Schitter. Prolific.ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2017.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang A. Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Rose Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. *ArXiv*, abs/2110.08207, 2022.

Aarohi Srivastava, Abhinav Rastogi, Abhishek B Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Annasaheb Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Santilli, Andreas Stuhlmuller, Andrew M. Dai, Andrew D. La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakacs, Bridget R. Roberts, Bao Sheng Loe, Barret Zoph, Bartlomiej Bojanowski, Batuhan Ozyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Stephen Howald, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, C'esar Ferri Ram'irez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Tatiana Ramirez, Clara Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Daniel H Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Gonz'alez, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, D. Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth P. Donoway, Ellie Pavlick, Emanuele Rodolà, Emma FC Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fan Xia, Fatemeh Siar, Fernando Mart'inez-Plumed, Francesca Happ'e, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo,

Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-L'opez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Han Sol Kim, Hannah Rashkin, Hanna Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hubert Wong, Ian Aik-Soon Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, John Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, J. Brooker Simon, James Koppel, James Zheng, James Zou, Jan Koco'n, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Narain Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jenni Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Oluwadara Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Jane W Waweru, John Burden, John Miller, John U. Balis, Jonathan Berant, Jorg Frohberg, Jos Rozen, José Hernández-Orallo, Joseph Boudeman, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Ochieng' Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Luca Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Col'on, Luke Metz, Lutfi Kerem cSenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Madotto Andrea, Maheen Saleem Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, M Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew Leavitt, Matthias Hagen, M'aty'as Schubert, Medina Baitemirova, Melissa Arnaud, Melvin Andrew McElrath, Michael A. Yee, Michael Cohen, Mi Gu, Michael I. Ivanitskiy, Michael Starritt, Michael Strube, Michal Swkedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Monica Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, T MukundVarma, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas S. Roberts, Nicholas Doiron, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W. Chang, Peter Eckersley, Phu Mon Htut, Pi-Bei Hwang, P. Milkowski, Piyush S. Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, QING LYU, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ram'on Risco Delgado, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib J. Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Sam Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi S. Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo hwan Lee, Spencer Bradley Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Rose Biderman, Stephanie C. Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq A. Ali, Tatsuo Hashimoto, Te-Lin Wu, Theo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, T. N. Kornev, Timothy Telleen-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler O. Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, W Vossen, Xiang Ren, Xiaoyu F Tong, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yang Song, Yasaman Bahri, Ye Ji Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yu Hou, Yushi Bai, Zachary Seid, Zhao Xinran, Zhuoye Zhao, Zi Fu Wang, Zijie J. Wang, Zirui Wang, Ziyi Wu, Sahib Singh, and Uri Shaham. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615, 2022.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. *ArXiv*, abs/1906.00591, 2019.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. Conditionalqa: A complex reading comprehension dataset with conditional answers. In *ACL*, 2022.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021a.

Alex Tamkin, Vincent Liu, Rongfei Lu, Daniel Fein, Colin Schultz, and Noah Goodman. Dabs: A domain-agnostic benchmark for self-supervised learning. *arXiv preprint arXiv:2111.12062*, 2021b.

Alex Tamkin, Gaurab Banerjee, Mohamed Owda, Vincent Liu, Shashank Rammoorthy, and Noah Goodman. Dabs 2.0: Improved datasets and algorithms for universal self-supervision. 2022a.

Alex Tamkin, Dat Nguyen, Salil Deshpande, Jesse Mu, and Noah D. Goodman. Active learning helps pretrained models learn the intended task. *ArXiv*, abs/2204.08491, 2022b.

Yu Wang and Eugene Agichtein. Query ambiguity revisited: Clickthrough measures for distinguishing informational and ambiguous queries. In *NAACL*, 2010.

Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? *ArXiv*, abs/2109.01247, 2022.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617, 2018.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2022.

Zeqiu Wu, Ryu Parish, Hao Cheng, Sewon Min, Prithviraj Ammanabrolu, Mari Ostendorf, and Hannaneh Hajishirzi. Inscit: Information-seeking conversations with mixed-initiative interactions. *arXiv preprint arXiv:2207.00746*, 2022.

Michael J.Q. Zhang and Eunsol Choi. Situatedqa: Incorporating extra-linguistic contexts into qa. *ArXiv*, abs/2109.06157, 2021.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. *ArXiv*, abs/2102.09690, 2021.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *EMNLP*, 2021.

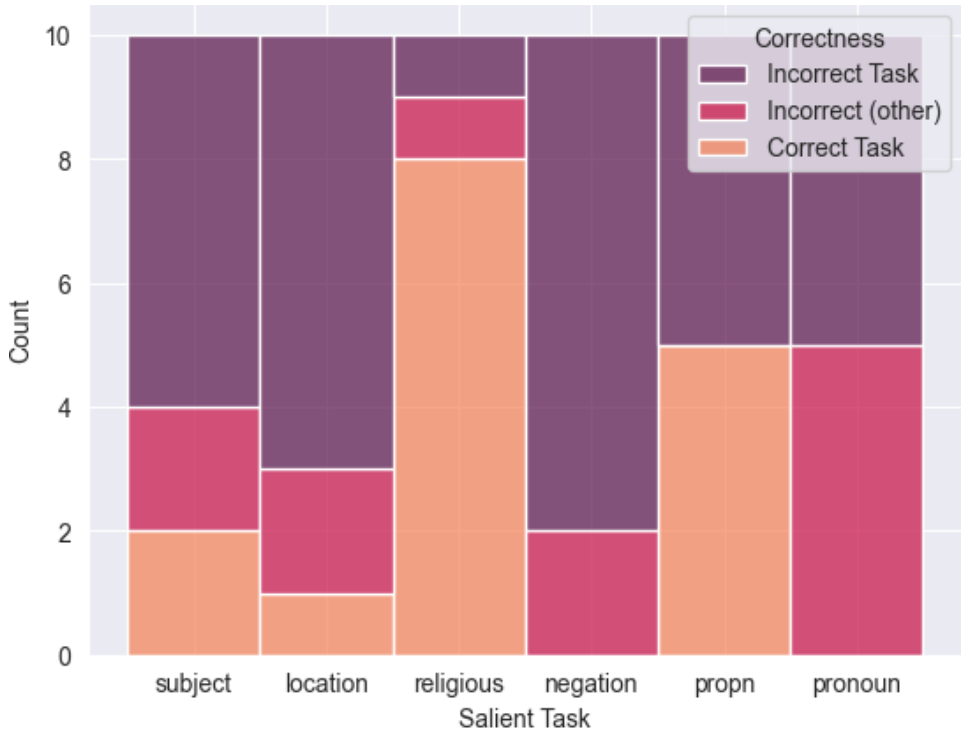


Figure 5: **Even models that perform very well on a task struggle to describe that task in words.** Accuracy of `text-davinci-002` for guessing the [category withheld] for each salient task

A CAN MODELS VERBALIZE THE TASK THEY ARE PERFORMING?

As models attempt to disambiguate between diferent tasks, a user may wish to know the model's best estimate of the task it is being asked to perform. In this section, we assess whether models can verbalize the task descriptions in Section 4.3 at the end of the in-context prompt. Specifically, after the 20 example prompt we append a newline with the string `'What is the [category withheld]?'`. We generate 10 outputs for each of the 6 tasks (Q/A format), and manually categorize them as one of three categories:

- **Correct task:** A correct verbalization of the task (e.g. "The [category withheld] is a religious leader."
- **Incorrect Task** The model guesses a task, but it is not the correct task. (e.g. "animals" when the salient task is "outdoor location")
- **Incorrect (Other)**: The verbalization does not mention a task (e.g. "The category is withheld.")

As shown in Figure 5, the model is sometimes able to verbalize the correct task, especially for religious figures and proper nouns, although for all other tasks it struggles. These initial experiments suggest that task verbalization is challenging even for the most capable models, even when they perform very well at the task, and suggests an exciting direction for future study. Note that our graph only shows results for the `text-02-davinci` model, Q/A format, as we did not observe in a preliminary investigation that other model-format combinations could could verbalize the task successfully.

B EXPLORING TASK AMBIGUITY FOR A NATURAL LANGUAGE COMMAND-LINE ASSISTANT

In this section we explore how the methodologies we introduce in AmbiBench can be extended to more real-world settings. Here, we consider the case of a natural language assistant for command-line tasks. In our hypothetical setting, an employee of a company prompts a language model to produce Amazon Web Services buckets for different users by providing it with two input-output examples of the desired behavior. By chance, the two training examples both use Japanese names and have the bucket location set to the `ap-northeast-1` region in Japan. The test examples ask for a bucket to be made for a person with a non-Japanese name (in our case, White American, Greek, or Indian names).

Concretely, examples look like the following:

```
Write the AWS CLI command to create an AWS Bucket.

Input: Create a bucket for Sato Tamotsu
Output: aws s3 mb s3://bucket-for-sato --region ap-northeast-1

Input: Create a bucket for Yuki Hashimoto
Output: aws s3 mb s3://bucket-for-yuki --region ap-northeast-1

Input: Create a bucket for Margaret Richards
Output:
```

The task ambiguity that arises is that sometimes the model assumes the bucket should be placed in the same region as the training examples, but in other cases it assumes the bucket should be placed in a different region based on the person's name. We omit other commandline arguments for readability, however we note that this ambiguity might be very difficult to notice in practice with long commands comprised of many arguments.

In Figure 6 we show that not only does this ambiguity manifest in the `text-davinci-002` language model outputs (which can output both regions), but it also varies based on the national origin of the person's name. White American names induce the model to output other regions (typically ones in the US and Europe) far more than Indian or Greek names do. However, after one additional name from that same national origin is added (indicating that the name should not determine the region) the model performs near perfectly at other names of this national origin.

This initial investigation illustrate how task ambiguity can manifest in a more real-world setting, and how we can measure it using the techniques we discuss in the rest of the paper. Future work could explore finetuning on similar in-context examples to meta-train the model such that it avoids assuming that an individual's name should impact the task behavior, for example.

B.1 LIST OF NAMES

Here we provide the full list of names we used to query the model. Names were constructed from a list of popular first and last names.[6]

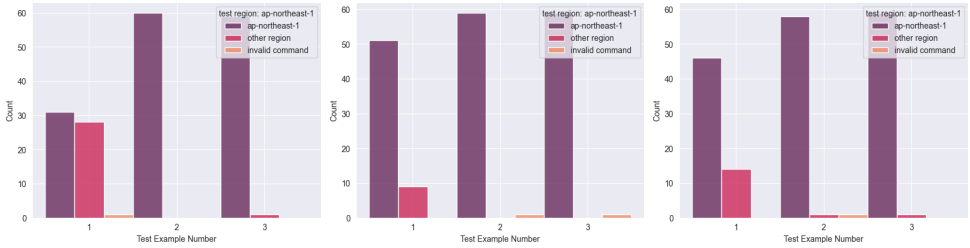
Japanese names: Sato Tamotsu, Yuki Hashimoto, Yamamoto Mitsuo, Kuroda Kumiko, Mita Naoki, Tanaka Sakura, Kuramoto Hideki, Sazama Miyuki, Hora Izanagi, Itoh Akane

White American names: Trevor Fitzpatrick, Jessica Price, Logan Evans, Leah Mills, Ava Smith, Cole Jones, Isabella Brown, Emily Davis, Mia Miller, Madison Wilson

Indian names: Rajesh Kumar, Abhishek Yadav, Rahul Singh, Archana Kumar, Anisha Patel, Priya Sharma, Riya Gupta, Ayush Jain, Prisha Rao, Sagar Goyal

Greek Names: Yiannis Papadopoulou, Vassilis Karagiannis, Athanasios Ioannou, Dimitra Papadakis, Vasiliki Georgiou, Eleni Oikonomou, Kostas Giannopoulos, Nikos Theodorou, Ioannis Vlachos, Katerina Makris

[6] forebears.io



(a) White American Test Names (b) Indian Test Names (c) Greek Test Names

Figure 6: **Features such as name origin can influence how models behave under task ambiguity.** Effect of task ambiguity on `text-davinci-002` when generating an AWS bucket command given an ambiguous natural language string. Graph shows distribution of generated regions: ap-northeast-01 (the region in the prompts), another region, or invalid command.

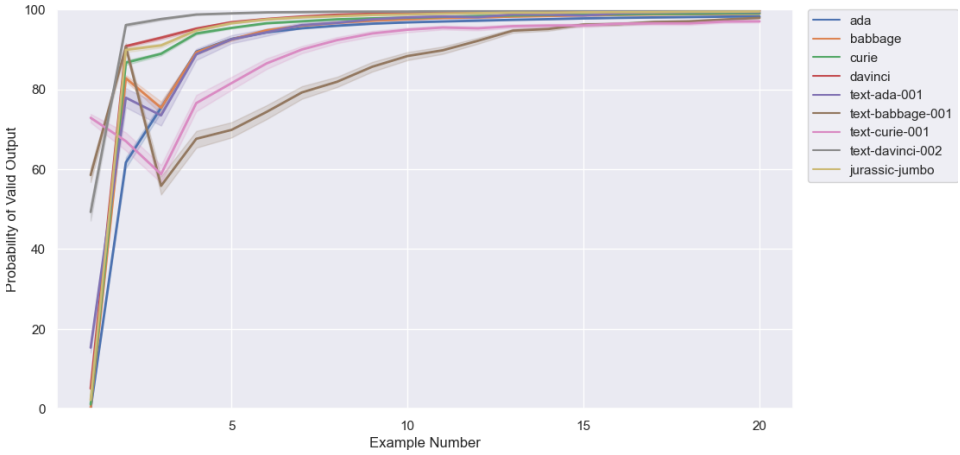


Figure 7: **Models of all sizes generate valid outputs by the end of the in-context learning episode.** Probability of a valid output (either `X` or `Y`) for all models across examples

C DO SMALLER MODELS UNDERSTAND THE TASK FORMAT?

In Section 4.3 we find that smaller models perform worse at disambiguating the intended task from multiple examples. However, is this due simply to models not understanding the desired output format? We validate the output formats of all models and find (Figure 7) that by the end of training, even the smallest models always generate valid outputs (i.e. either X or Y) across the experiments we considered, suggesting that their poor performance on task disambiguation is not simply due to their poor prompting abilities.

D HOW IMPORTANT IS THE FORMAT OF THE UNINFORMATIVE INSTRUCTIONS?

As previously noted, our experiments on uninformative examples in the main text use an instruction format containing `[CATEGORY WITHHELD]`. Here we test whether the behavior of models is sensitive to variations in this phrasing. We replace `[CATEGORY WITHHELD]` with the linguistic nonce word `wug`, so a full instruction might read `Output 'X' if the sentence contains a wug and 'Y' otherwise`. As shown in Figure 8 we find that performance is the same for both tasks, suggesting that the particular form of uninformative instruction is not very important.

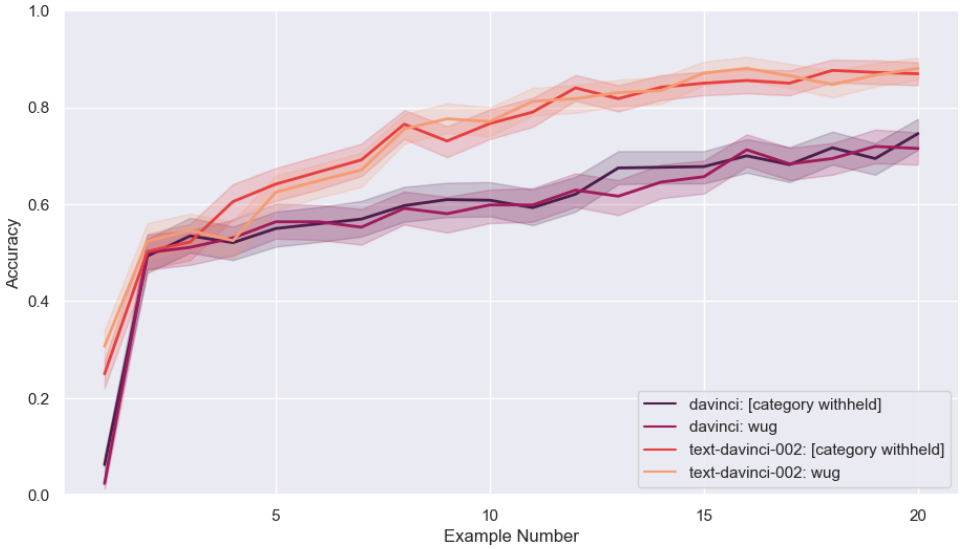


Figure 8: **The choice of format for the uninformative instructions has negligible effect.** `davinci` and `text-davinci-002` performance with different forms of uninformative instructions

Model	Number of Parameters
OpenAI Ada *(normal and RHLF models)*	350M
OpenAI Babbage *(normal and RHLF models)*	1.3B
OpenAI Curie *(normal and RHLF models)*	6.7B
OpenAI Davinci *(normal and RHLF models)*	175B
AI21 J1-Jumbo	178B
T0++	11B

Table 2: Estimated number of parameters for each model. Note that the OpenAI parameter counts have not been released, and it is also possible that the normal and RLHF models have different numbers of parameters.

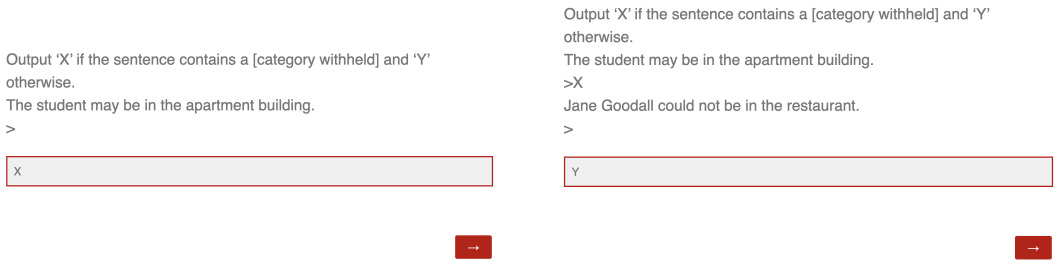
E Parameter counts for each model

An estimate of the number of parameters for each model is shown in Table 2 for Ada,[7] Babbage,[8] Curie,[9] Davinci,[10] J1-Jumbo,[11] and T0++.[12] Note that because the exact parameter counts of the OpenAI models have not been released, it is entirely possible that these estimates are incorrect, and that the RLHF versions of the models are not the same size as their traditional counterparts. Furthermore, note that parameter count is not necessarily the most informative proxy for a model's capabilities, given the importance of training data quality and number of tokens seen by the model (Hoffmann et al., 2022).

F Additional Information on Humans (Prolific) Experiments

Here we provide more information about the experiments with human participants via Prolific (Palan & Schitter, 2017).

[7]https://blog.eleuther.ai/gpt3-model-sizes/
[8]https://blog.eleuther.ai/gpt3-model-sizes/
[9]https://blog.eleuther.ai/gpt3-model-sizes/
[10]https://blog.eleuther.ai/gpt3-model-sizes/
[11]https://www.ai21.com/blog/announcing-ai21-studio-and-jurassic-1
[12]https://bigscience.huggingface.co/blog/t0



(a) First example shown to participants in 20-example test

(b) Second example shown to participants in 20-example test

Figure 9: Example questions shown to participants in Section 4.3

Prior to beginning the experiment, Prolific participants were shown a message confirming consent followed by a set of boiler-plate instructions on the upcoming task which stated: 1) "Continue the pattern in the following screens to the best of your ability, using the provided instructions and examples given" 2) "Each time you make a prediction, the next screen will show the correct answer for that example (not necessarily your previous answer)" and 3) "The pattern may not be obvious at first but may become more clear over time. Note: some information in the instructions may be [intentionally withheld] (denoted by brackets)" This prelude was deemed necessary after a series of pilot runs in which participants quit out of the survey due to their confusions and/or frustrations with the lack of guidance. To further streamline the process, participants were also given a set of useful keyboard shortcuts with which to progress through the survey (although participants were not allowed to edit previous answers).

For the tests in Section 4.3, participants were shown examples one-by-one with the correct answer appearing on the screen following their input. For the tests in Section 4.2 tests, participants were shown both labeled examples and the one unlabeled query example at the same time.

For the tests in Section 4.3, each participant was only given one set of 20 examples. For the tests in Section 4.2, each participant was only given a single prompt. Prompts given to participants were randomized and roughly evenly distributed across all tests. Participant IDs were filtered to ensure that the same participant did not participate multiple times across studies. All surveys were created using Qualtrics' survey builder.

F.1 Number of Participants for Human Experiments

Task disambiguation using natural language instruction For the experiment referenced in Section 4.2 we recruited 51 and 33 participants for the unambiguous instruction and informative instruction tests respectively.
Task disambiguation using multiple examples For the experiment detailed in Section 4.3, we recruited 96 total participants: 16 per salient feature.

G AmbiBench Benchmark: Full Details

AmbiBench consists of ambiguous sentences with two explicitly varying (salient) features in each sentence. Salient features are grouped into pairs: 'subject' and 'location, 'religious' and 'pronoun', and 'proper noun' and 'negation.' Each salient feature has two values (e.g. *animal subject* or *human subject*) with each choice having a 0.5 probability of being selected.

G.1 Salient Feature Pairings

The pairings of the salient features are detailed below. The pairings remained constant throughout all experiments.

G.1.1 SUBJECT-LOCATION SENTENCES

Sentences follow the template: "The {*human/animal subject*} is in the {*indoor/outdoor location*}." The subject is randomly assigned to either a human or an animal. If chosen to be a human, the subject is randomly chosen from a list of professions: *[student, reporter, hiker, researcher, firefighter, fugitive, critic, photographer, director, surveyor]*. If chosen to be an animal, the subject is randomly chosen from a list of common animals: *[boar, worm, hawk, hound, butterfly, snake, duck, bear, mountain lion, horse]*. The location is randomly assigned to either an outdoor location or an indoor location. If chosen to be an outdoor location, the location is chosen from a list of common outdoor locations: *[river, pond, woodlands, cave, canyon, prairie, jungle, marsh, lagoon, meadow]* whereas if chosen to be an indoor location, the location is chosen from a list of common indoor locations: *[laboratory, theatre, museum, courtroom, apartment building, restaurant, house, film studio, hotel lobby, grocery store]*.

G.1.2 RELIGIOUS-PRONOUN SENTENCES

Sentences follow the template: "{*She/He*} is in the {indoor location} with the {*religious/secular leader*}." The pronoun is randomly assigned to either a *He* or *She*. The leader is randomly assigned to either a religious leader or a secular leader. If chosen to be an religious leader, the leader is chosen from a list of common religious leaders: *[pope, reverend, bishop, Dalai Lama, rabbi, cardinal, pastor, deacon, imam, ayatollah]* whereas if chosen to be a secular leader, the leader is chosen from a list of common secular leader: *[president, CEO, principal, sheriff, judge, ambassador, officer, prime minister, colonel, professor]*. The indoor location in the sentence is randomly chosen from the aforementioned list of indoor locations.

G.1.3 PROPER NOUN-NEGATION SENTENCES

Sentences follow the template: "{*Proper/Common noun*} {*negation/affirmation*} in the {indoor location}." The noun is randomly assigned to either a proper noun or a common noun. If chosen to be a proper noun, the noun is randomly chosen from a list of famous individuals across a variety of disciplines: [*Lebron James, Bernie Sanders, Christopher Nolan, Paul Atreides, Noam Chomsky, Serena Williams, Margot Robbie, Alexandria Ocasio-Cortez, Hermione Granger, Jane Goodall*]. If chosen to be a common noun, the noun is randomly chosen from the prior list of human subjects: [*student, reporter, hiker, researcher, firefighter, fugitive, critic, photographer, director, surveyor*] and appended to 'The' to maintain grammaticality of the sentence. The verb is randomly assigned to either a negation or an affirmation. If chosen to be a negation, the verb is chosen from a list of common negation verb phrases: *[is not, was not, has not been, may not be, could not be]*. If chosen to be an affirmation, the verb is chosen from the list of affirmations directly contrasting the list of negations: *[is, was, has been, may be, could be]*.

G.2 INSTRUCTIONS

Instructions given were either informative instructions, explicitly stating the salient feature which determined the label for each sentence, or uninformative instructions which did not divulge the salient feature. Informative Instructions were of the template: "Output 'X' if the sentence {*salient feature instruction*} and 'Y' otherwise", where *salient feature instruction* could be filled in with one of {contains a reference to an indoor/outdoor location, contains a reference to a human/an animal, contains a reference to a religious leader, does not contain a reference to a religious leader, contains a reference to a religious leader, contains a male pronoun, contains a female pronoun, contains a proper noun, does not contain a proper noun, contains a negation, does not contain a negation}

Uninformative instructions were always given as "Output 'X' if the sentence contains a [category withheld] and 'Y' otherwise."

G.3 TASK AMBIGUATION USING NATURAL LANGUAGE INSTRUCTIONS

When constructing tests for Section 4.2, we ensured task ambiguity by grouping one possibility for a given salient feature with one possibility for its paired salient feature.

For example, for Subject-Location sentences, humans and indoor locations were grouped and animals and outdoor locations were grouped.

In this gtest, humans and models were given one example with the first of these groupings and a second example with the second of the groupings. The final, 'disambiguating,' example broke the groupings, therefore displaying which of the features was responsible for the example's label (the salient feature).

Here is a demonstration of a single prompt for an example with the Subject-Location sentence type:

Q: The hawk is in the canyon.
A: X
Q: The director is in the museum.
A: Y
Q: The hiker is in the meadow.
A: X

For the first two examples, it is unclear whether the subject or the location is controlling the label but upon seeing the third example, it becomes clear that the location and not the subject is controlling the label (as the sentence is labeled X whenever an outdoor location is referenced).

Tests contained either an uninformative instruction or an informative instruction followed by three examples. Humans and models were shown all three examples at once. Humans were shown the first two examples with labels and the last example without the label. Models were shown all three examples with labels within the same API query (as the log probabilities for the final label can be acquired without requiring a completion).

The order of groupings, order of labels, labels' correspondence with a salient feature, and the salient feature were randomized across all tests with equal probabilities being given to each possibility.

Some example prompts from the test set:

Prompt 1. Salient feature is 'subject' (Q/A format with informative instructions):

```
Output 'X' if the sentence contains a reference to a human and 'Y'
otherwise.
The horse is in the woodlands.
A: Y
Q: The student is in the laboratory.
A: X
Q: The mountain lion is in the film studio.
A: Y
```

Prompt 2. Salient feature is 'pronoun' (arrow format with uninformative instructions):

```
Output 'X' if the sentence contains a [category withheld] and 'Y'
otherwise.
He is in the film studio with the imam.
>Y
She is in the restaurant with the judge.
>X
She is in the apartment building with the bishop.
>X
```

Prompt 3. Salient feature is 'proper noun' (Q/A format with uninformative instructions):

```
Output 'X' if the sentence contains a [category withheld] and 'Y'
otherwise.
Q: The student was not in the hotel lobby.
A: X
Q: Lebron James is in the theatre.
A: Y
Q: Christopher Nolan could not be in the house.
A: Y
```

G.4 TASK DISAMBIGUATION USING MULTIPLE EXAMPLES

When constructing tests for Section 4.3, humans and models were shown a set of 20 questions with the same salient feature. All sentences within the set contained the same sentence pairing Subject-Location, Religious-Pronoun, Proper Noun-Negation.

For each example, the possibilities for each of the two features were randomized and given equal probability (as described in the process of sentence construction). The groupings mentioned for the tests in Section 4.2 were not used.

The salient feature was randomly selected (with each possibility being given an equal probability).

Tests always contained an uninformative instruction prior to the set of examples.

Humans were shown questions one at a time, progressively building up from 1 to 20 questions. After each question, the correct answer was displayed on the screen (in place of their solution if incorrect). Models were shown all 20 questions within the same API query and the log probabilities for each of the twenty labels were tracked.

As with the tests in Section 4.2, the labels' correspondence with a salient feature and the order of labels were randomized across all tests with equal probabilities being given to each possibility.

Some example prompts from the test set:

Prompt 1. Salient feature is 'location' (arrow format):

```
Output 'X' if the sentence contains a [category withheld] and 'Y'
otherwise.
The photographer is in the restaurant.
>Y
The mountain lion is in the river.
>X
The hiker is in the cave.
>X
The butterfly is in the grocery store.
>Y
The surveyor is in the river.
>X
The boar is in the prairie.
>X [continues until 20 examples]
```

Prompt 2. Salient feature is 'religious' (Q/A format):

```
Output 'X' if the sentence contains a [category withheld] and 'Y'
otherwise.
Q: He is in the film studio with the ayatollah.
A: X
Q: She is in the house with the CEO.
A: Y
Q: He is in the apartment building with the ambassador.
A: Y
Q: She is in the museum with the rabbi.
A: X
Q: She is in the museum with the Dalai Lama.
A: X
Q: He is in the laboratory with the rabbi.
A: X [continues until 20 examples]
```

Prompt 3. Salient feature is 'negation' (arrow format):

```
Output 'X' if the sentence contains a [category withheld] and 'Y'
otherwise.
The student was not in the restaurant.
>X
Christopher Nolan was in the apartment building.
>Y
```

```
The photographer has not been in the theatre.
>X
Alexandria Ocasio-Cortez could be in the film studio.
>Y
Christopher Nolan was not in the museum.
>X
The photographer was in the film studio.
>Y [continues until 20 examples]
```

G.5 Finetuning a Model to Generalize in the Face of Task Ambiguity

When constructing the finetuning dataset for Section 4.4, two out of the three possible sentence pairings were used. We then tested on the held-out sentence pairing. For example, if the finetuning dataset contained the salient features: 'subject,' location,' 'religious,' and 'pronoun,' the held-out features would be 'proper noun' and 'negation.'

For the control experiments, the dataset consisted of examples for which only one feature varied. For example, if the sentence was of the Subject-Location sentence type and the salient feature was location, a given example may contain examples with both outdoor and indoor locations but only humans.

For the ambiguous experiments, the dataset consisted of examples with ambiguity: both features varied between the two possibilities for each feature.

The number of examples in each prompt varied: building from 4 examples to 20 examples for each salient feature. But the examples across these different prompts did not remain constant (rather just the salient feature did), unlike in Section 4.3 where a prompt with a length of n examples simply added one example to the preceding prompt of length n-1.

As with the tests in Section 4.2 and Section 4.3, the labels' correspondence with a salient feature and the order of labels were randomized across all tests with equal probabilities being given to each possibility.

H Additional Information about Finetuning Experiments

Hyperparameters We use OpenAI's finetuning API to finetune their `davinci` model.[13] When conducting the finetuning, we used a batch size of 1, a learning rate multiplier of 0.1, and a prompt loss weight of 0.1.

Control experiment settings Our control experiment for the finetuning only varied one of the two features in each set of 20 examples (e.g. *boar* could change to *worm* but not *firefighter*). The choice of which feature would be constant (e.g. *human/animal* vs *indoor/outdoor*), as well as the value of that feature (e.g. *human* vs *animal*) were decided randomly for each set of 20 examples. By following this procedure, we did not introduce task ambiguity into the finetuning data, but still ensured that the control model saw the same range of sentence constructions as the model trained on ambiguous examples.

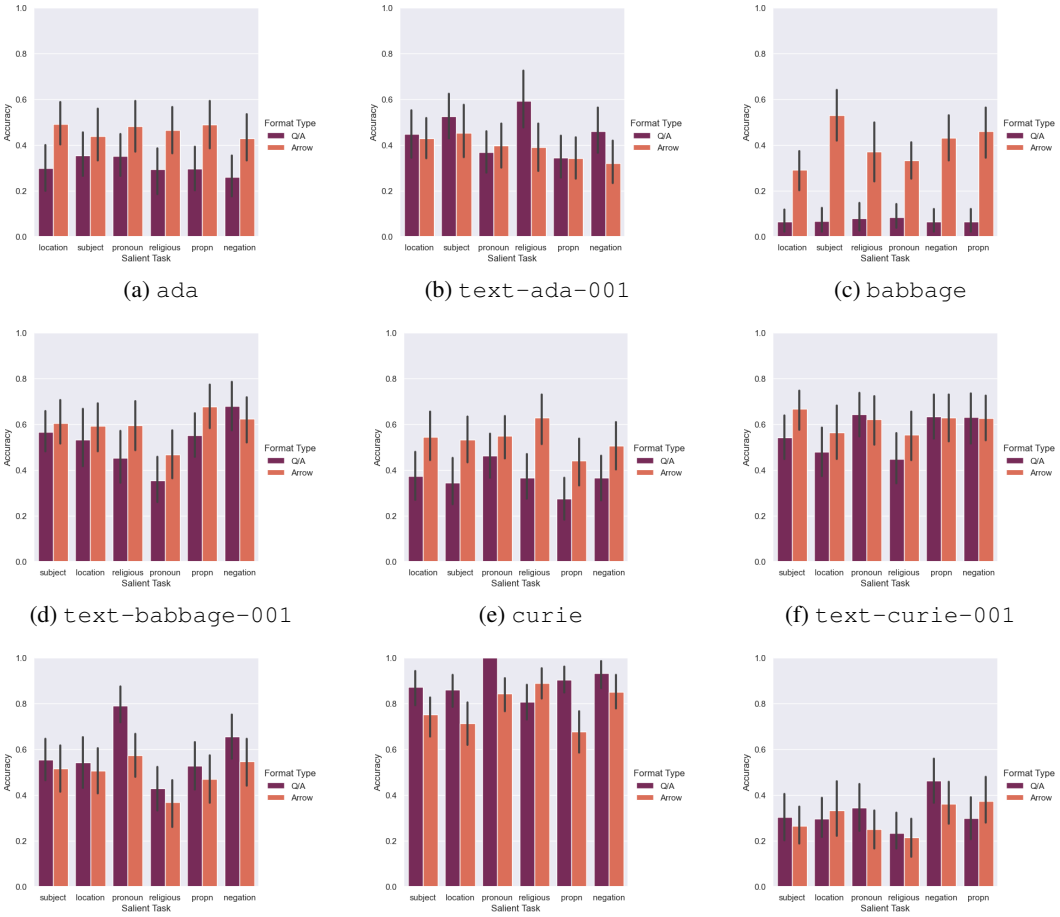
I Additional Graphs

I.1 Task Ambiguation Using Natural Language Instructions

I.1.1 Informative Instructions

Figure 10 details the performance across salient features for each model individually across both format types in instances of informative instructions. Figure 11 averages the performance of the models across salient features, demonstrating the difference in performance by format type for each salient feature.

[13] https://beta.openai.com/docs/guides/fine-tuning



(a) `ada` (b) `text-ada-001` (c) `babbage`

(d) `text-babbage-001` (e) `curie` (f) `text-curie-001`

(g) `davinci` (h) `text-davinci-002` (i) `j1-jumbo`

Figure 10: Performance with informative instructions across salient features on Task Ambiguation Using Natural Language Instructions (*see Section 4.2*)

I.1.2 Uninformative Instructions

Figure 12 and Figure 13 mirror Figure 10 and Figure 11 respectively but instead demonstrate the performance in prompts with uninformative instructions.

I.2 Task Disambiguation Using Multiple Examples

Figure 14, Figure 15, Figure 16, Figure 17, and Figure 18 individually detail the performance of each model across each salient feature from of 1-20 examples.

Figure 19b demonstrates difference in performance across the two format types, averaged for all models. Because of `t0pp`'s poor performance on prompts with the arrow format, we do not include `t0pp` in the collective performance graph but instead display its performance across formats individually from the other models.

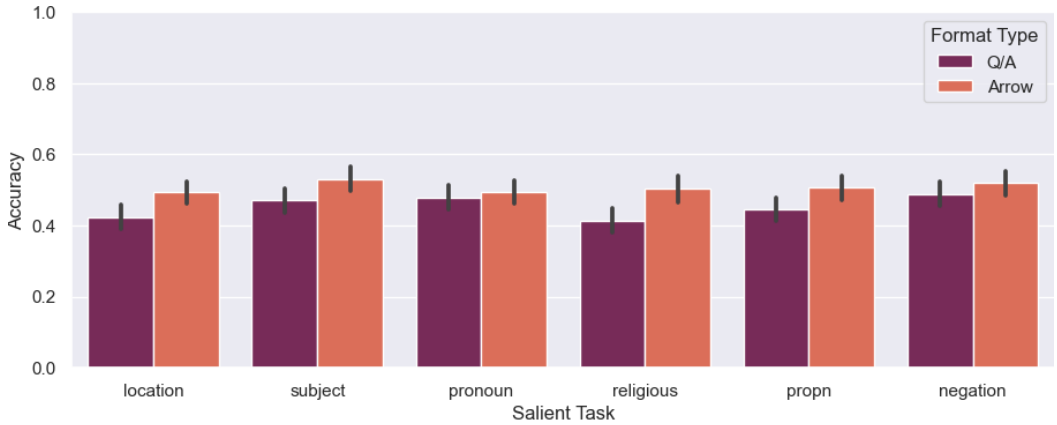
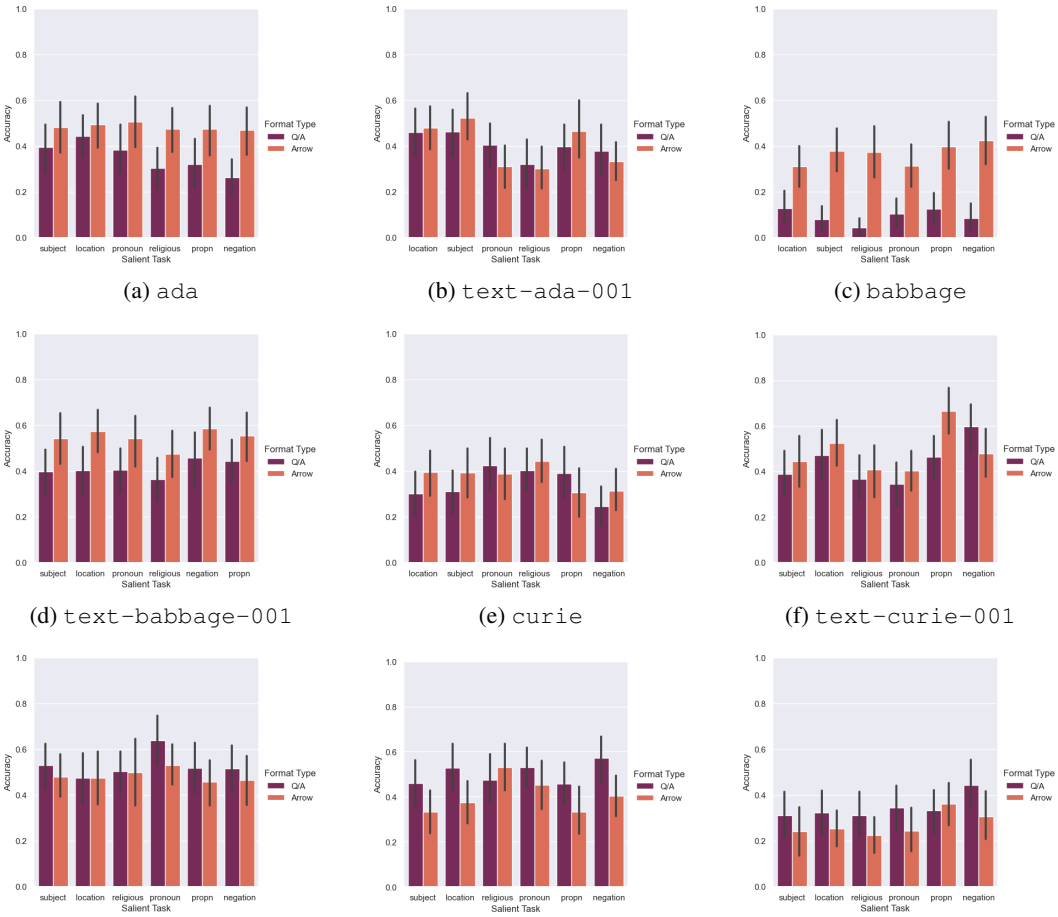


Figure 11: Performance with informative instructions for each format averaged over all models on Task Ambiguation Using Natural Language Instructions (*see Section 4.2*)



(a) `ada` (b) `text-ada-001` (c) `babbage`

(d) `text-babbage-001` (e) `curie` (f) `text-curie-001`

(g) `davinci` (h) `text-davinci-002` (i) `j1-jumbo`

Figure 12: Performance with uninformative instructions across salient features on Task Ambiguation Using Natural Language Instructions (*see Section 4.2*)

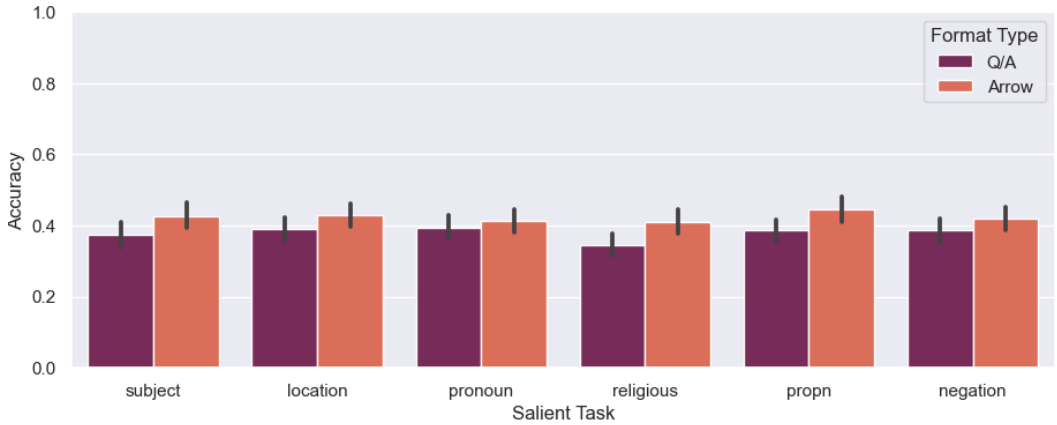
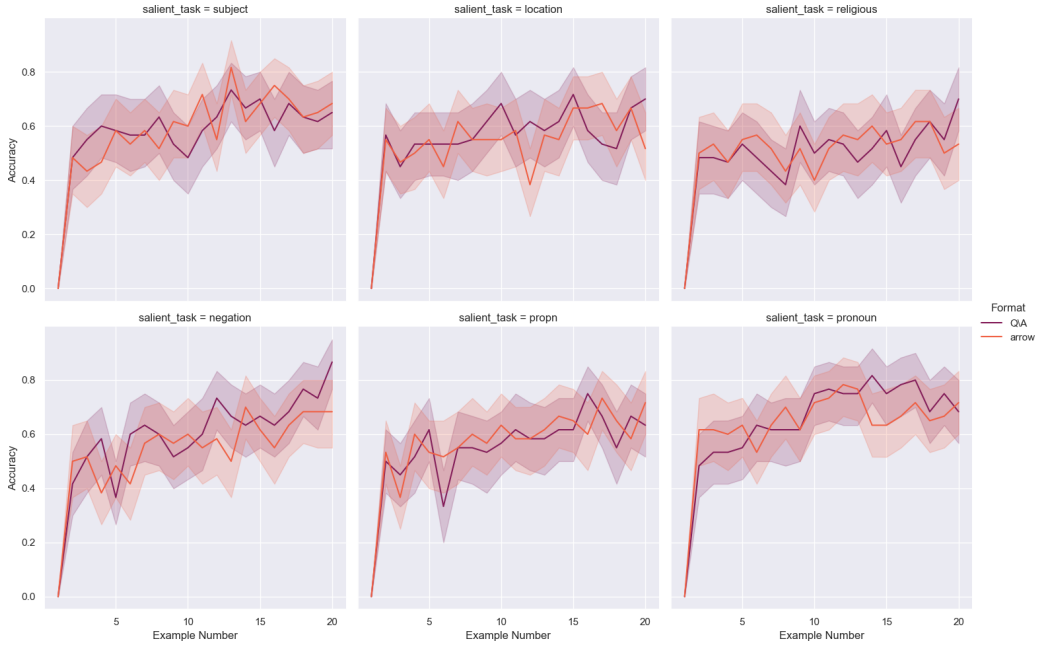
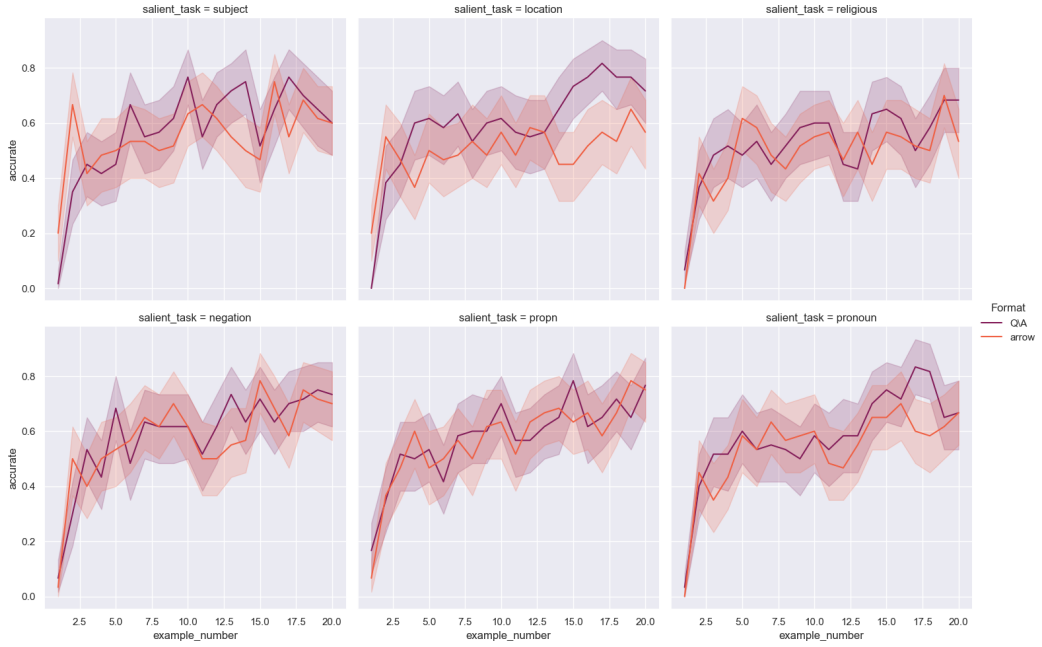


Figure 13: Performance with uninformative instructions for each format averaged over all models on Task Ambiguation Using Natural Language Instructions (*see Section 4.2*)

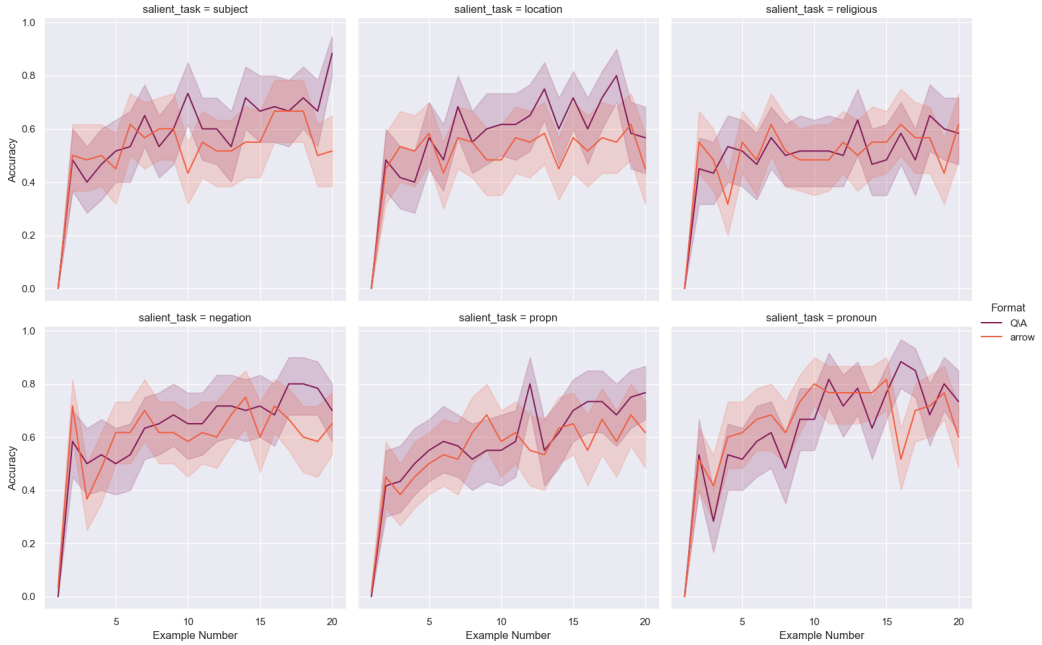


(a) `ada`

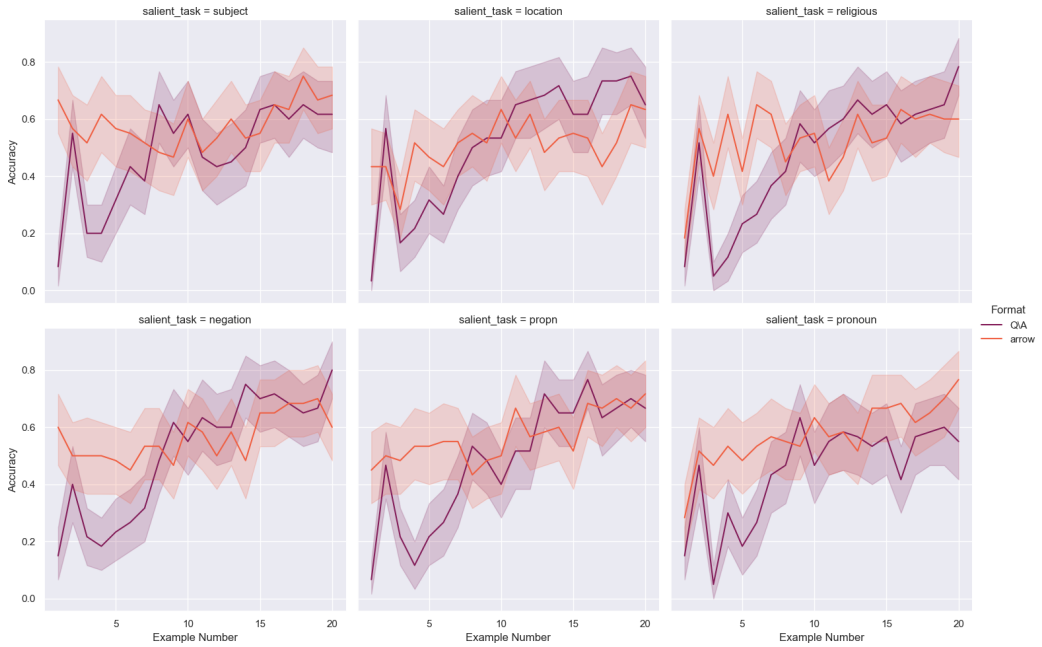


(b) `text-ada-001`

Figure 14: Performance across salient features on Task Disambiguation Using Multiple Examples (*see Section 4.3*)

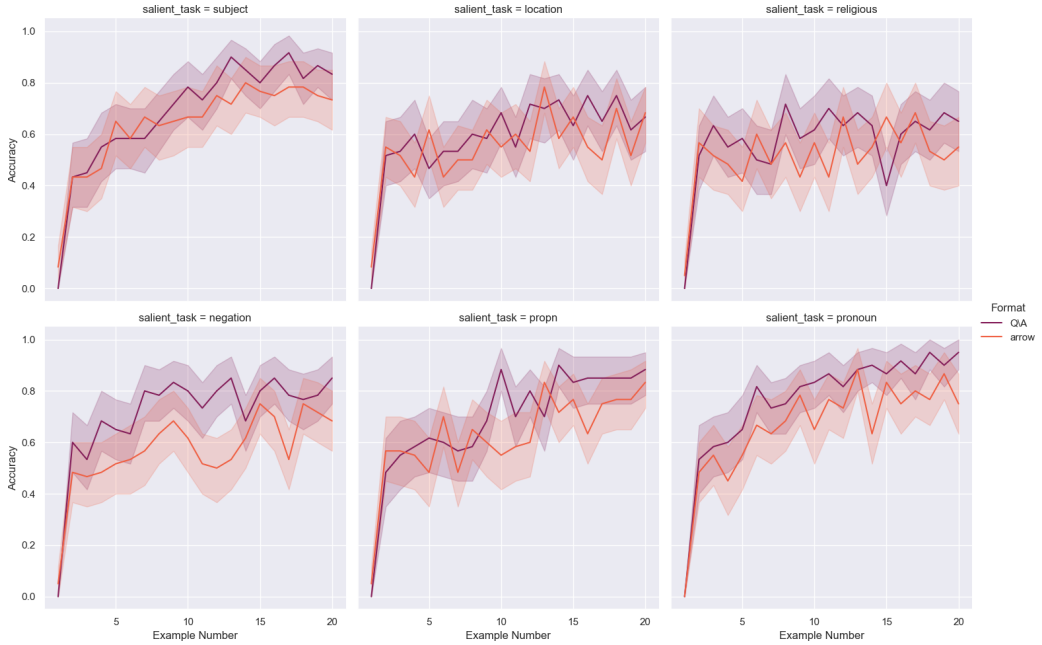


(a) `babbage`

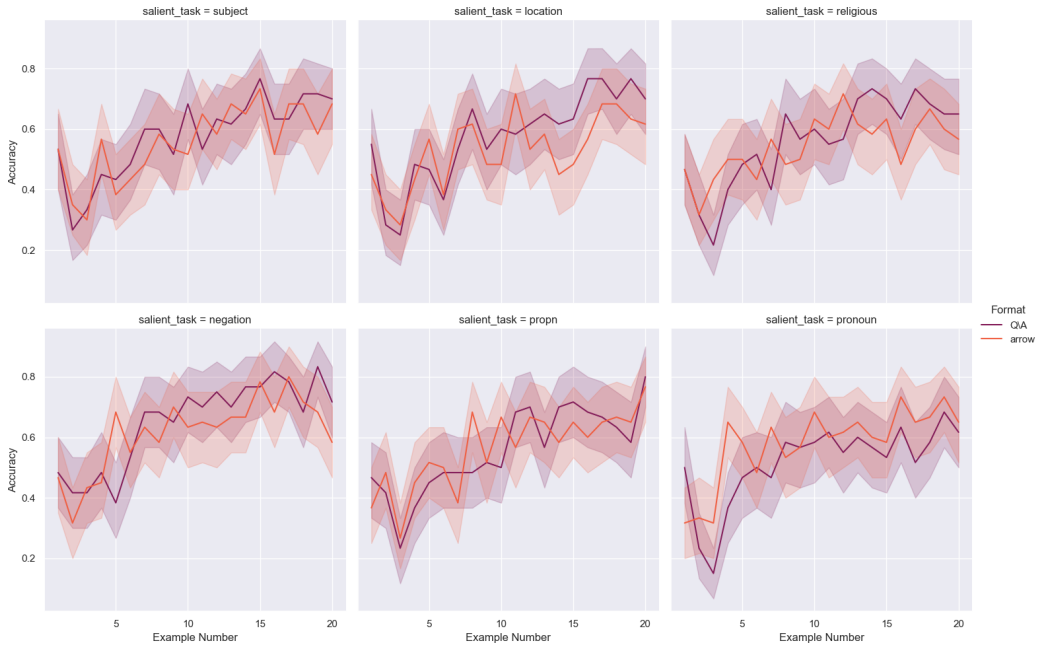


(b) `text-babbage-001`

Figure 15: Performance across salient features on Task Disambiguation Using Multiple Examples (*see Section 4.3*)

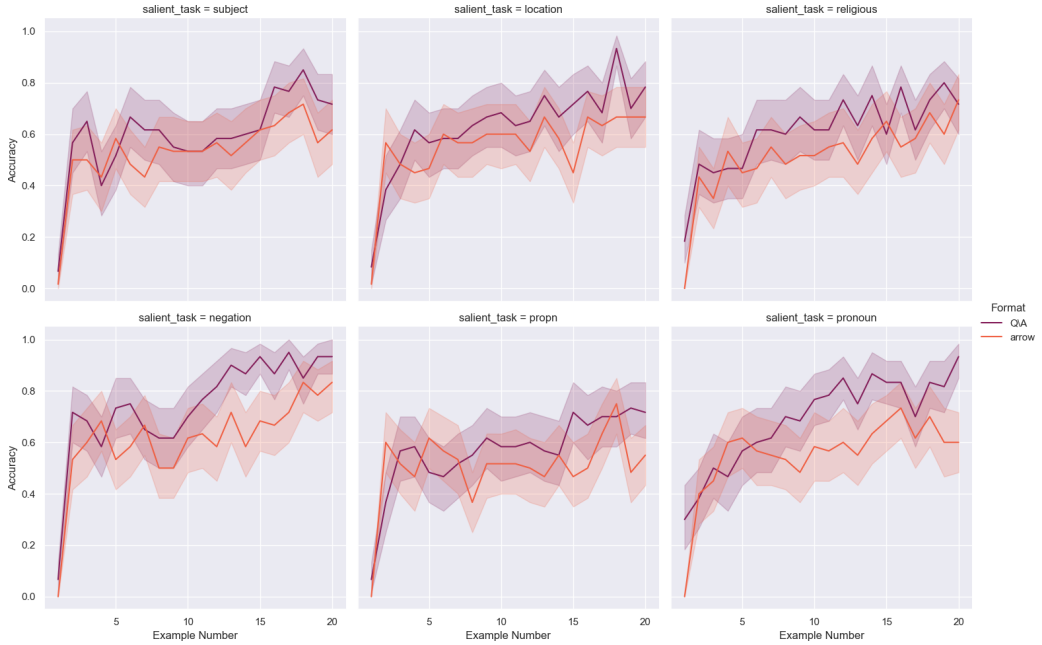


(a) `curie`

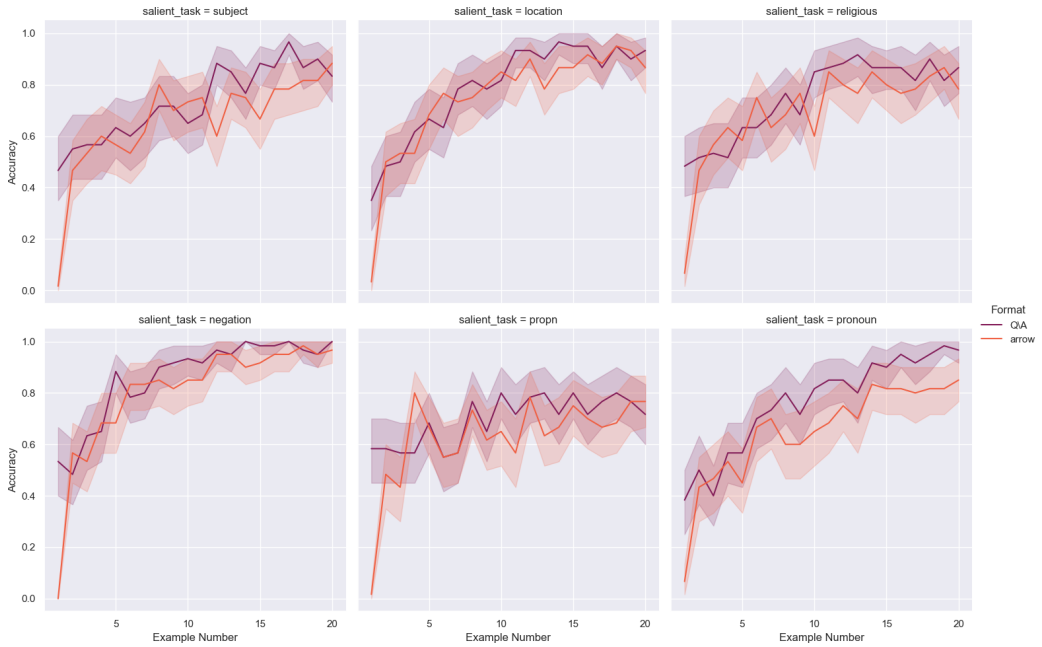


(b) `text-curie-001`

Figure 16: Performance across salient features on Task Disambiguation Using Multiple Examples (*see Section 4.3*)

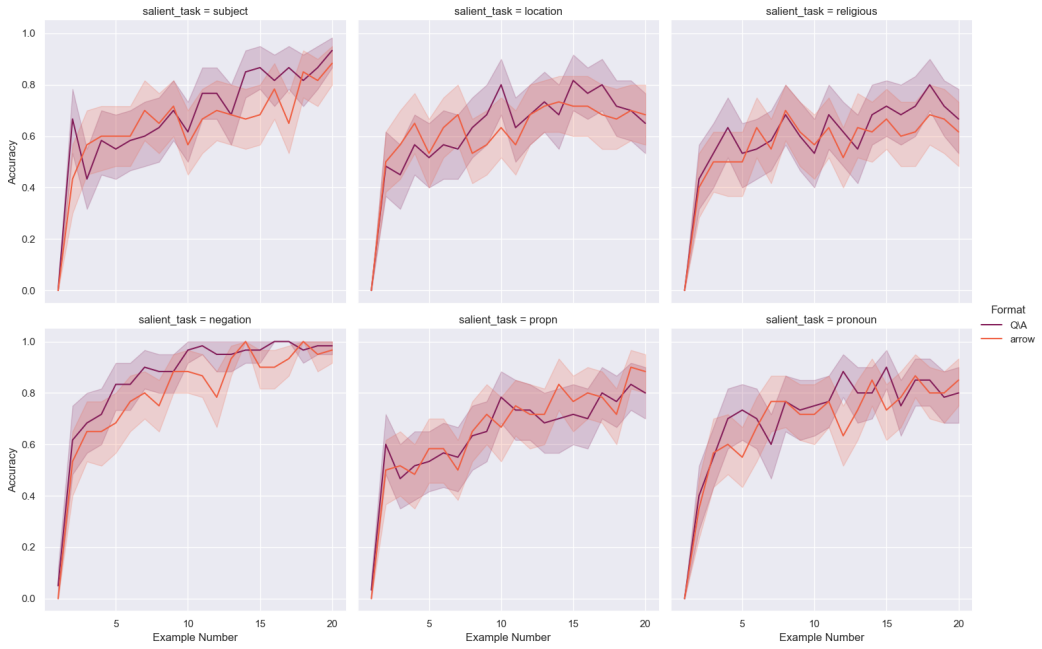


(a) `davinci`



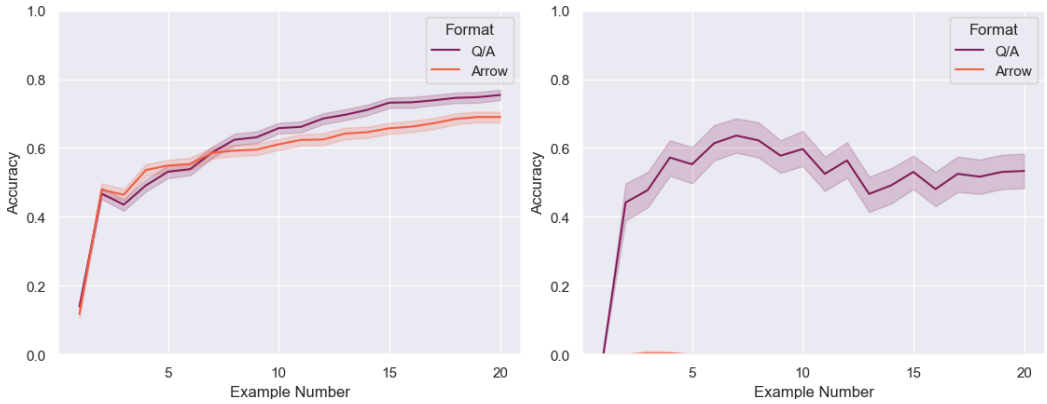
(b) `text-davinci-002`

Figure 17: Performance across salient features on Task Disambiguation Using Multiple Examples (*see Section 4.3*)



(a) `j1-jumbo`

Figure 18: Performance across salient features on Task Disambiguation Using Multiple Examples (*see Section 4.3*)



(a) Averaged over all models except `t0pp`

(b) `t0pp`
note: 'Arrow' format is at 0.0 throughout

Figure 19: Performance of each format on Task Disambiguation Using Multiple Examples (*see Section 4.3*)